HỒ HOÀNG TRIẾT

Coding Techniques

CÁC CHƯƠNG TRÌNH MẪU

.NET

Visual Basic.NET
ASP.NET
.NET Framework
Data Access
Security
Networking
...

http://www.ntbooks.vn

NHÀ XUẤT BẢN THỐNG KÊ

HỒ HOÀNG TRIẾT

Coding Techniques

CÁC CHƯƠNG TRÌNH MẪU

.NET

Visual Basic.NET
ASP.NET
.NET Framework
Data Access
Security
Networking

NHÀ XUẤT BẢN THỐNG KÊ

CÁC CHƯƠNG TRÌNH MẪU .NET

CHỊU TRÁCH NHIỆM XUẤT BẢN : CÁT VĂN THÀNH
BIÊN TẬP : NGUYÊNHẠNH
SỬA BẢN IN : TÁC GIẢ
BÌA : HHH

In 1.000 cuốn khổ (16x24)cm tại Xưởng In Trung Tâm Hội Chợ Triển Lãm Việt Nam. GPXB số 198-205/XB-QLXB do Cục Xuất Bản cấp ngày 03/03/2003. In xong & nộp lưu chiểu quý 3 năm 2003.

Mục lục

MỤC LỤC **5**

PHẦN I - WINDOWS FORMS **9**

1. Kết hợp Help vào ứng dụng 9
2. Tạo ứng dụng theo kiểu Explorer 13
3. Crystal Report Print Preview 27
4. Tạo control động 40
5. Điều khiển Office 56
6. Các hộp thoại thông dụng 65
7. Tạo một chương trình trên System Tray 74
8. Các Exception tùy biến 78
9. Truy xuất và duyệt dữ liệu 87
10. Định dạng DataGrid 94
11. Sắp xếp và lọc dữ liệu trên DataGrid 103
12. In ấn đơn giản 107
13. ListBox và ComboBox 111
14. Sử dụng các menu 119
15. Quản lý nhiều form cấp cao nhất (Top-Level) 131
16. Sử dụng Kéo và Thả (Drag and Drop) 140
17. Thừa kế Windows Forms 145
18. Kiểm tra tính hợp lệ của các TextBox (Validating TextBox) 152
19. Tự vẽ Menu 158
20. Sử dụng các mã định dạng để định dạng dữ liệu trong các chuỗi. 165
21. Sử dụng Regular Expression 174
22. Sử dụng Clipboard 192

PHẦN II - SECURITY **197**

23. Tạo hộp thoại Login 197
24. Mã hóa và giải mã dữ liệu 203
25. Windows Role-Based Security 216
26. Sử dụng thuật toán mã hóa Hash (băm) 222

PHẦN III - KẾT HỢP NHIỀU PHẦN **228**

27. Điều khiển Internet Explorer 228

PHẦN IV - FILE 233

28. Truy xuất File System 233
29. Theo dõi File System 238

PHẦN V - WEB DEVELOPMENT 247

30. Tạo một ứng dụng Web nhập dữ liệu 247
31. Xây dựng các dịch vụ Web (Web Service) 259
32. Tạo ứng dụng Web Master-Details 269
33. Phân trang DataGrid trên Web Forms 277

PHẦN VI - WEB SERVICES 286

34. Sử dụng các Web Service 286
35. Truyền dữ liệu nhị phân thông qua HTTP 297

PHẦN VII - LANGUAGE 307

36. Sử dụng mảng - Array 307
37. Callback 317
38. Sử dụng các kiểu dữ liệu DateTime và TimeSpan 322
39. Các tính năng hướng đối tượng trong Visual Basic .NET 331
40. Chuỗi – String 341
41. Try – Catch – Finally 359

PHẦN VIII - CÁC CHỦ ĐỀ CAO CẤP 364

42. Đa tiểu trình – Multithreading 364
43. TCP Remoting 367
44. Serializing và Deserializing các đối tượng 388

PHẦN IX - XỬ LÝ DỮ LIỆU 397

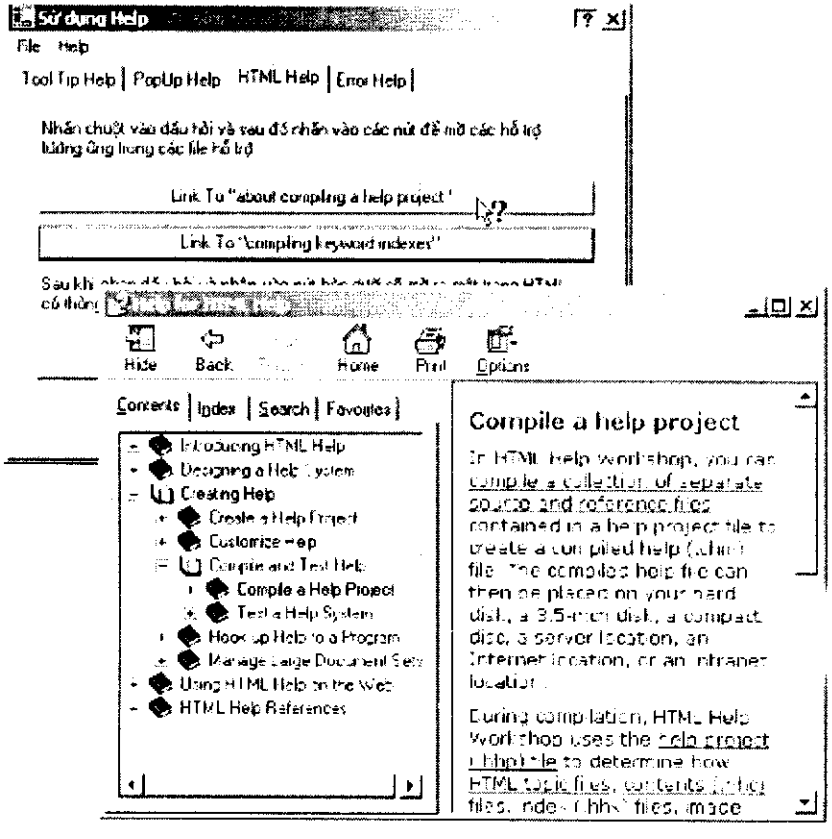
45. Kết dữ liệu vào một ComboBox 397
46. Tạo ứng dụng Master-Details Windows Forms 403
47. Tạo một ứng dụng Offline 417
48. Tạo một Form nhập liệu 431
49. Tạo và sử dụng một Database 446
50. Tạo ứng dụng theo mô hình 3 lớp (3-Tier) 457
51. Đọc và ghi các hình ảnh vào Database 468
52. Truy xuất và xử lý dữ liệu sử dụng DataReader 476
53. Sắp xếp và lọc dữ liệu trên DataView 490
54. DataSet có cấu trúc trước (Typed DataSet) 497
55. Sử dụng ADO 2.6 trong .NET 509
56. Định dạng dữ liệu trước khi hiển thị và lưu vào database 517
57. Sử dụng Stored Procedure 530

PHẦN X - MICROSOFT .NET FRAMEWORK 547

58. Làm hoạt hình với GDI+ 547
59. Tạo một Windows Service 553
60. Tạo một Screen Saver với GDI+ 564
61. Tạo và sử dụng các Trace Listener 573
62. Tạo Enterprise Service Component 578
63. Tạo và sử dụng File tạm thời 592
64. Cấu hình cho ứng dụng 596
65. Lấy thông tin hệ thống và môi trường. 611
66. MSMQ (Microsoft Message Queuing) 620
67. Process Viewer 632
68. Reflection 646
69. Gửi và nhận dữ liệu thông qua HTTP 651
70. Gửi Email 660
71. Service Manager 667
72. Stack Frame 673
73. System Events 676
74. Sử dụng class XMLDocument 680
75. Sử dụng [GDI+] Pen 699
76. Tương tác với Windows Service 706
77. Xử lý hình ảnh với GDI+ 714
78. Phân chia một ứng dụng 724
79. Class Process 729
80. Đọc và ghi file Text 735
81. Tương tác với các Performance Counter 741
82. Scope, Override và Overload 748
83. Sử dụng Thread Pooling 763
84. Tìm hiểu Garbage Collector 776
85. Lập trình Socket 784
86. So sánh các control dùng trên Web Forms và Windows Forms ... 797
87. Sử dụng cổng COM 812
88. Sử dụng Event Log 838
89. Sử dụng Win32 API 846
90. Một số lợi điểm của Visual Basic .NET 859
91. Windows Management Instrumentation 874
92. Làm việc với các ứng dụng Console 879
93. Sử dụng các GDI+ Brush 883
94. GDI+ Text 891
95. Sử dụng các file Resource 898

Phần I - Windows Forms

1. Kết hợp Help vào ứng dụng



Chương trình này cho thấy cách thêm chức năng hỗ trợ (help) vào một ứng dụng Windows viết bằng Visual Basic .NET bằng cách sử dụng các control ToolTip, HelpProvider, và ErrorProvider.

Chú ý:

- Source code hỗ trợ cho các chương trình được trình bày trong sách này có thể download tại trang web **http://www.ntbooksvn.com/.**
- Vì khổ sách có giới hạn nên Source code trình bày trong sách này có một dấu chấm tại đầu mỗi dòng để biểu diễn một dòng code trong Visual Basic .NET.

Danh sách các control và thuộc tính chính:

Control	Thuộc tính	Giá trị
Form	(Name)	frmMain
	FormBoderStyle	FixedSingle
	HelpButton	True
	MaximizeBox	False
	MinimizeBox	False
MainMenu	(Name)	mnuMain
HelpProvider	(Name)	hpAdvanceCHM
	HelpNameSapce	..\htmlhelp.chm
HelpProvider	(Name)	hpPlainHTML
	HelpNameSpace	..\help.htm
ToolTip	(Name)	ToolTip1
ErrorProvider	(Name)	ErrorProvider1
TextBox	(Name)	txtProductName
	ToolTip on ToolTip1	Enter a product name.
RichTextBox	(Name)	rtbTextEntry
	HelpString on hpAdvanceCHM	This is the text entry area. Use this area to enter text which can be saved to an rtf file.
	ShowHelp on hpAdvanceCHM	True
Button	(Name)	btnLink1
	HelpKeyword on hpAdvanceCHM	about compiling a help project
	HelpNavigator on hpAdvancedCHM	KeywordIndex
	ShowHelp on hpAdvanceCHM	True
Button	(Name)	btnLink3
	HelpKeyword on hpPlainHTML	help.htm
	ShowHelp on hpPlainHTML	True
TextBox	(Name)	txtNumberValue

Chi tiết mnuMain:

Loại	Thuộc tính	Giá trị
Cha	(Name)	mnuFile
	Text	&File
Con của File	(Name)	mnuExit
	Text	E&xit
Cha	(Name)	mnuHelp
	Text	&Help
Con của Help	(Name)	mnuContentsHelp
	Text	Contents...
Con của Help	(Name)	mnuIndexHelp
	Text	Index...
Con của Help	(Name)	mnuSearchHelp
	Text	Search...

Chức năng và cách sử dụng các control:

- Control **ToolTip** được sử dụng để hiển thị thông điệp hỗ trợ ngắn cho các control khi đặt con trỏ chuột phía trên control. Ta chỉ cần đặt vào project một control ToolTip và sau đó trên các control muốn hiển thị tooltip ta chỉ cần nhập vào chuỗi cần hiển thị vào thuộc tính ToolTip on <tên ToolTip> trong cửa sổ **Properties** của control đó (ví dụ: TextBox txtProductName).
- Control **HelpProvider** được sử dụng để hỗ trợ chức năng "What's this" trên các control, hiển thị thông tin hỗ trợ trên một cửa sổ nhỏ. Đặt control HelpProvider vào form, và nhập đường dẫn của file help (.chm, .htm) vào thuộc tính HelpNamespace. Đối với control cần hỗ trợ, nhập chuỗi muốn hiển thị vào thuộc tính HelpString on <tên HelpProvider> và đặt thuộc tính ShowHelp on <tên HelpProvider> bằng True. Người sử dụng chọn dấu hỏi ở trên thanh tiêu đề của form và nhấn vào control, một cửa sổ nhỏ xuất hiện chứa nội dung là chuỗi đã gán (ví dụ: RichTextBox rtbTextEntry).
- Control **HelpProvider** còn có thể được sử dụng để liên kết với một từ khóa cụ thể trong một file .chm. Nhập chuỗi từ khóa cần tìm trong file .chm vào thuộc tính HelpKeyword on <tên HelpProvider> và gán thuộc tính HelpNavigator on <tên HelpProvider> bằng giá trị KeywordIndex, cuối cùng đặt giá trị của thuộc tính ShowHelp on <Tên HelpProvider> là True. Khi người sử dụng dùng chuột với dấu hỏi nhấn vào control, cửa sổ HTML help sẽ mở ra và tìm nội dung với từ khóa trên (ví dụ: Button btnLink1).
- **HelpProvider** được sử dụng để hiển thị một trang HTML cơ bản để cung cấp chức năng hỗ trợ cho control. Gán đường dẫn đến một file HTML vào thuộc tính HelpNamespace của HelpProvider. Trên control muốn hỗ trợ, gán giá trị thuộc tính ShowHelp on <Tên HelpProvider> là True. Khi người sử dụng có yêu cầu hỗ trợ trên control này, một trình duyệt với nội dung là file HTML trên được mở ra, (ví dụ: btnLink3 và HelpProvider hpPlainHTML).
- Control **ErrorProvider** được sử dụng để hiển thị thông báo trực quan trên một control mà dữ liệu nhập vào không hợp lệ. Trên control muốn thông báo khi có lỗi, đặt thuộc tính CauseValidation bằng True (ví dụ txtNumberValue), và trong thủ tục Validation ta sử dụng phương thức SetError của ErrorProvider để thông báo lỗi.

Project: **Ket hop Help vao ung dung**

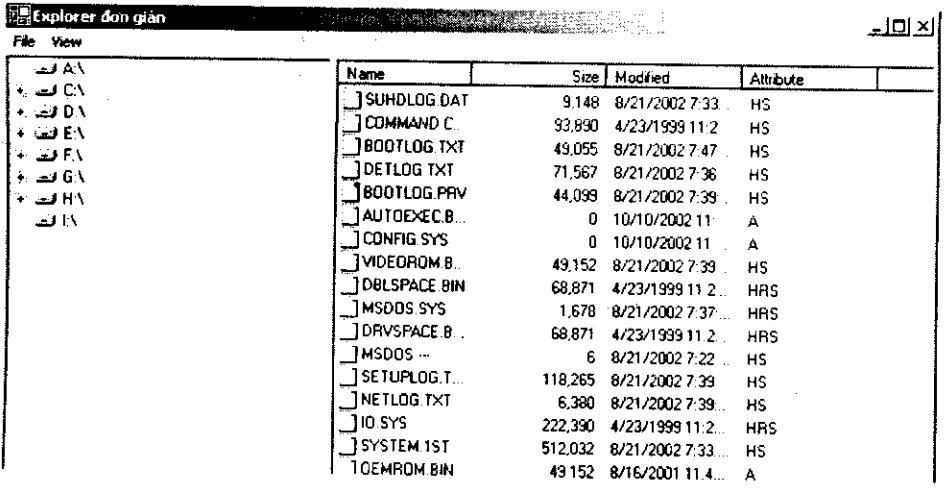
```
Private Sub txtNumberValue_Validating(ByVal sender As Object,
  ByVal e As System.ComponentModel.CancelEventArgs) Handles
  txtNumberValue.Validating
  If Not IsNumeric(txtNumberValue.Text) Then
    ' Kích hoạt errorprovider để thông báo lỗi cho người sử dụng.
    ErrorProvider1.SetError(txtNumberValue, "Not a numeric value.")
  Else
    ' Xóa thông báo lỗi
    ErrorProvider1.SetError(txtNumberValue, "")
  End If
End Sub
```

➢ Control **HelpProvider** còn được sử dụng để hiển thị các tab chức năng Contents, Index, và Search trong một file .chm. Để hiển thị tab Contents ta sử dụng phương thức ShowHelp, hiển thị tab Index sử dụng phương thức ShowHelpIndex, và hiển thị tab Search ta sử dụng phương thức ShowHelp kèm với tham số HelpNavigator.Find. Ví dụ:

```
Private Sub mnuContentsHelp_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuContentsHelp.Click
    ' Hiển thị nội dung (contents) của file help .chm.
    Help.ShowHelp(Me, hpAdvancedCHM.HelpNamespace)
End Sub

  Private Sub mnuIndexHelp_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuIndexHelp.Click
    ' Hiển thị chỉ mục (Index) của file help.
    Help.ShowHelpIndex(Me, hpAdvancedCHM.HelpNamespace)
  End Sub

  Private Sub mnuSearchHelp_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuSearchHelp.Click

    ' Hiển thị tab tìm kiếm của file help .chm.
    Help.ShowHelp(Me, hpAdvancedCHM.HelpNamespace, HelpNavigator.Find, "")
  End Sub
```

2. Tạo ứng dụng theo kiểu Explorer

Phần này gồm hai ví dụ biểu diễn các ứng dụng kiểu Explorer: DirectoryScanner và ExplorerStyleViewer.



Các chức năng và các control:

- **DirectoryScanner** là ứng dụng đơn giản quét qua tất cả thư mục và các thư mục con trong tất cả các ổ đĩa có trên máy tính hoặc người sử dụng chọn một thư mục khởi đầu để quét. Một control TreeView được sử dụng để hiển thị cấu trúc thư mục được quét. Các thư mục được hiển thị theo các màu xanh, vàng, và đỏ tùy vào tổng kích thước của thư mục. Control ListView sử dụng để hiển thị các thư mục con và kích thước của chúng.

Project: **Tao ung dung kieu Explorer**

DirectoryScanner.vb

```
Option Strict On
Imports System.IO

Public Class DirectoryScanner
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
    Const MB As Long = 1024 * 1024
```

```
    ' Thủ tục xử lý cho menu "Scan | All Directories".
    Private Sub mnuScanAll_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuScanAll.Click

        ' Lấy danh sách các ổ đĩa logic cho vào một mảng
        Dim Drives As String() = Directory.GetLogicalDrives()
        Dim drive As String

        tvwDirectories.Nodes.Clear()
        lvwDirectories.Items.Clear()

        For Each drive In Drives
            Dim dnDrive As DirectoryNode

            Try
            ' Tạo một DirectoryNode tương ứng cho một đĩa logic và cho nó vào TreeView.
                dnDrive = New DirectoryNode()
                dnDrive.Text = drive.Remove(Len(drive) - 1, 1)
                tvwDirectories.Nodes.Add(dnDrive)

                ' Tính toán kích thước của ổ đĩa bằng cách cộng kích thước
                ' của tất cả các thư mục con của nó lại.
                dnDrive.Size += GetDirectorySize(drive, dnDrive)
            Catch exc As Exception
                ' Không làm gì cả. Chỉ đơn giản là nhảy qua các thực mục
                ' không thể đọc. Và chương trình tiếp tục tại dòng code đầu
                ' tiên đứng sau  End Try.
            End Try
        Next
    End Sub

    ' Thủ tục xử lý cho menu "Scan | From One Directory"
    Private Sub mnuScanFromOn_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuScanFromOne.Click

        ' Hiển thị hộp thoại FolderBrowser và chọn thư mục bắt đầu bằng
        ' thư mục do người sử dụng chọn.
        Dim strSelectedDirectory As String = FolderBrowser.ShowDialog()
        Dim dnSelectedDirectory As DirectoryNode
```

```
    tvwDirectories.Nodes.Clear()
    lvwDirectories.Items.Clear()

    Try
        ' Thêm DirectoryNode tương ứng cho thư mục được chọn vào TreeView
        dnSelectedDirectory = New DirectoryNode()
        dnSelectedDirectory.Text = strSelectedDirectory
        tvwDirectories.Nodes.Add(dnSelectedDirectory)
        ' Tính kích thước của thư mục được chọn.
        dnSelectedDirectory.Size += GetDirectorySize(strSelectedDirectory,
dnSelectedDirectory)
    Catch exc As Exception
        ' Không làm gì cả
    End Try
End Sub

' Thủ tục xử lý sự kiện Load của form DirectoryScaner.
Private Sub DirectoryScanner_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ' Thiết lập các cột của ListView
    lvwDirectories.Columns.Add("Size", 90, HorizontalAlignment.Left)
    lvwDirectories.Columns.Add("Folder Name", 400, HorizontalAlignment.Left)
End Sub

Private Sub TreeView_AfterExpand(ByVal sender As Object, ByVal e As
TreeViewEventArgs) Handles tvwDirectories.AfterExpand e.Node.Expand()
    ShowSubDirectories(CType(e.Node, DirectoryNode))
End Sub

Private Sub TreeView_AfterSelect(ByVal sender As Object, ByVal e As
TreeViewEventArgs) Handles tvwDirectories.AfterSelect

    Dim strSubDirectory As DirectoryNode = CType(e.Node, DirectoryNode)
    lvwDirectories.Items.Clear()
    AddToListView(Format(strSubDirectory.Size / (1024 * 1024), "F") + "MB", _
                  strSubDirectory.Text)

End Sub
```

```
    ' Thủ tục này thêm strSubDirectory do người sử dụng chọn trong
    ' TreeView vàoListView, và quy định các giá trị text, size, và color
Private Sub AddToListView(ByVal strSize As String, ByVal strFolderName As String)
    Dim lvi As New ListViewItem()
    Dim lvsi As ListViewItem.ListViewSubItem

    lvi.Text = strSize
    lvi.ForeColor = GetSizeColor(strSize)

    lvsi = New ListViewItem.ListViewSubItem()
    lvsi.Text = strFolderName
    lvi.SubItems.Add(lvsi)
    lvwDirectories.Items.Add(lvi)
End Sub

    ' Hàm này trả về một màu dựa trên kích thước tổng của thư mục và
    ' tất cả thư mục con của nó. Đây là hàm thứ nhất trong hai hàm overload.
Private Function GetSizeColor(ByVal strSize As String) As System.Drawing.Color
    Return GetSizeColor(CLng(CDbl(strSize.Substring(0, _
        strSize.LastIndexOf("M") - 1)) * MB))
End Function

    ' Hàm này trả về một màu dựa trên kích thước tổng của thư mục và
    ' tất cả thư mục con của nó. Đây là hàm thứ hai trong hai hàm overload.
Private Function GetSizeColor(ByVal intSize As Long) As System.Drawing.Color
    Select Case intSize
        Case 200 * MB To 500 * MB
            Return System.Drawing.Color.Gold
        Case Is > 500 * MB
            Return System.Drawing.Color.Red
        Case Else
            Return System.Drawing.Color.Green
    End Select
End Function

    ' Hàm này trả về kích thước của một thư mục, và tất cả thư mục con của nó.
Public Function GetDirectorySize(ByVal strDirPath As String, _
    ByVal dnDriveOrDirectory As DirectoryNode) As Long
    ' Hiển thị thư mục đang được quét trên thanh trạng thái
```

```
    sbrScan.Text = "Scanning : " + strDirPath
    Try
      Dim astrSubDirectories As String() = Directory.GetDirectories(strDirPath)
      Dim strSubDirectory As String

        ' Kích thước của thư mục hiện hành phụ thuộc vào kích thước
        ' của các thư mục con trong mảng astrSubDirectories. Do đó lặp
        ' lại trong mảng và sử dụng đệ quy để tính tổng kích thước của
        ' thư mục  hiện hành và tất cả thư mục con của nó.
      For Each strSubDirectory In astrSubDirectories
        Dim dnSubDirectoryNode As DirectoryNode
        dnSubDirectoryNode = New DirectoryNode()

          ' Gán giá trị text của node bằng phần cuối cùng của đường dẫn đầy đủ
        dnSubDirectoryNode.Text =   strSubDirectory.Remove(0, _
                                    strSubDirectory.LastIndexOf("\") + 1)
          ' Dòng bên dưới là đệ quy.
        dnDriveOrDirectory.Size += _
          GetDirectorySize(strSubDirectory, dnSubDirectoryNode)
        dnDriveOrDirectory.Nodes.Add(dnSubDirectoryNode)
      Next

      ' Cộng với kích thước của tất cả các file trong thư mục  hiện hành
      Dim astrFiles As String() = Directory.GetFiles(strDirPath)
      Dim strFileName As String
      Dim Size As Long = 0

      For Each strFileName In astrFiles
        dnDriveOrDirectory.Size += New FileInfo(strFileName).Length
      Next
        ' Gán màu cho TreeNode dựa trên tổng kích thước tính được
      dnDriveOrDirectory.ForeColor = GetSizeColor(dnDriveOrDirectory.Size)
    Catch exc As Exception
        ' không làm gì cả đối với các thư mục không đọc được
    End Try
     ' Xóa thanh trạng thái và trả về tổng kích thước của thư mục này
    sbrScan.Text = ""
    Return dnDriveOrDirectory.Size
  End Function
```

```
    ' Khi một thư mục được mở ra, thêm các thư mục con của nó vào Listview.
    Public Sub ShowSubDirectories(ByVal dnDrive As DirectoryNode)
        Dim strSubDirectory As DirectoryNode
        lvwDirectories.Items.Clear()
        For Each strSubDirectory In dnDrive.Nodes
          AddToListView(Format(strSubDirectory.Size / MB, "F") + "MB", _
                                                    strSubDirectory.Text)
        Next
    End Sub
End Class
```

DirectoryNode.vb

```
Option Strict On
' Class này chỉ đơn giản mở rộng class TreeNode, thêm thuộc tính Size
' để lưu kích thước của thư mục nhằm hỗ trợ cho mục đích chọn màu.
Public Class DirectoryNode
    Inherits TreeNode
    Public Size As Long
End Class
```

FolderList.vb

```
Option Strict On
Imports System.Windows.Forms

' Cần có class này bởi vì
' System.Windows.Forms.Design.FolderNameEditor.FolderBrowser
' được bảo vệ (Protected) và không thể truy xuất trong ngữ cảnh này.
' Phát sinh một class Public từ nó sẽ cho phép sử dụng hộp thoại trong code.
Public Class FolderBrowser
    Inherits System.Windows.Forms.Design.FolderNameEditor

    Public Shared Function ShowDialog() As String
        Dim fb As New FolderBrowser()
        fb.Description = "Select a Directory to Scan"
        fb.Style = Design.FolderNameEditor.FolderBrowserStyles.RestrictToFilesystem
        fb.ShowDialog()
        Return fb.DirectoryPath
    End Function
End Class
```

- **ExplorerStyleViewer** là một phiên bản đơn giản của Windows Exolorer. Chương trình này cung cấp nhiều thông tin hơn **DirectoryScanner**, biểu diễn cách kết hợp các biểu tượng với các kiểu file, và cho phép người sử dụng chạy một ứng dụng tương ứng với kiểu của file bằng cách nhấp đúp chuột vào tên file. Chương trình này cũng biểu diễn cách phát sinh control mà không cần thiết kế giao diện.

DirectoryTreeView.vb

```
Option Strict On
Imports System.IO

Public Class ExplorerStyleViewer
    Inherits System.Windows.Forms.Form

    ' Khai báo các biến để lưu các instance của các class tự tạo
    Private dtvwDirectory As DirectoryTreeView
    Private flvFiles As FileListView
    Private mivChecked As MenuItemView
    Public Sub New()
        MyBase.New()
        InitializeComponent()

        ' Bắt đầu phần code viết thêm

        ' Tạo một instance flvFilesView
        flvFiles = New FileListView()
        flvFiles.Parent = Me
        flvFiles.Dock = DockStyle.Fill

        ' Tạo một instance Splitter (thanh chia)
        Dim split As New Splitter()
        split.Parent = Me
        split.Dock = DockStyle.Left
        split.BackColor = SystemColors.Control

        ' Tạo một instance DirectoryTreeView và thêm một event handler OnAfterSelect
        dtvwDirectory = New DirectoryTreeView()
        dtvwDirectory.Parent = Me
        dtvwDirectory.Dock = DockStyle.Left
```

```
    AddHandler dtvwDirectory.AfterSelect, _
      AddressOf DirectoryTreeViewOnAfterSelect

    ' Thêm menu View vào menu chính đã tồn tại
    Dim mnuView As New MenuItem("&View")
    mnuView.Index = 1
    Me.mnuMain.MenuItems.Add(mnuView)

    ' Thêm bốn menu con vào menu View. Bắt đầu bằng cách tạo
    ' một mảng để quy định các thuộc tính của các menu con.
    Dim astrView As String() = {"Lar&ge Icons", "S&mall Icons", "&List", "&Details"}
    Dim aview As View() = {View.LargeIcon, View.SmallIcon, View.List, View.Details}
    ' Tạo một event handler cho các menu con.
    Dim eh As New EventHandler(AddressOf MenuOnViewSelect)

    Dim i As Integer
    For i = 0 To 3
        ' Sử dụng một class tùy biến MenuItemView, class này mở
        ' rộng class MenuItem để hỗ trợ thuộc tính View.
      Dim miv As New MenuItemView()
      miv.Text = astrView(i)
      miv.View = aview(i)
      miv.RadioCheck = True
        ' Kết handler đã tạo với sự kiện Click.
      AddHandler miv.Click, eh

        ' Chọn chế độ mặc nhiên là Details
      If i = 3 Then
        mivChecked = miv
        mivChecked.Checked = True
        flvFiles.View = mivChecked.View
      End If
        ' Thêm menu con vào menu View
      mnuView.MenuItems.Add(miv)
    Next i
  End Sub

' Code phát sinh được bỏ qua
```

```
    ' Thủ tục này làm cho FileListView hiển thị nội dung của thư mục được chọn.
    Sub DirectoryTreeViewOnAfterSelect(ByVal obj As Object, _
      ByVal tvea As TreeViewEventArgs)
        flvFiles.ShowFiles(tvea.Node.FullPath)
    End Sub

    ' Thủ tục sự kiện cho các menu con của menu View
    Sub MenuOnViewSelect(ByVal obj As Object, ByVal ea As EventArgs)
        ' Xóa đánh dấu mục đang được chọn
        mivChecked.Checked = False
        ' Đánh dấu menuitem được chọn.
        mivChecked = CType(obj, MenuItemView)
        mivChecked.Checked = True

        ' Thay đổi cách hiển thị các file trong FileListView
        flvFiles.View = mivChecked.View
    End Sub
End Class
```

DirectoryTreeView.vb

```
Option Strict On
Imports System.IO
Class DirectoryTreeView
    Inherits TreeView
    Public Sub New()
        ' Nới rộng không gian hơn để hiển thị các tên thư mục dài.
        Me.Width *= 2
        ' Lấy các hình ảnh cho cây
        Me.ImageList = New ImageList()
        With Me.ImageList.Images
            .Add(New Bitmap("..\ExplorerStyleViewer\35FLOPPY.BMP"))
            .Add(New Bitmap("..\ExplorerStyleViewer\CLSDFOLD.BMP"))
            .Add(New Bitmap("..\ExplorerStyleViewer\OPENFOLD.BMP"))
        End With

        ' Xây dựng lại cây
        RefreshTree()
    End Sub
```

```
Protected Overrides Sub OnBeforeExpand(ByVal tvcea As
TreeViewCancelEventArgs)
    MyBase.OnBeforeExpand(tvcea)

    ' Đặt code cập nhật cây vào giữa hai lệnh BeginUpdate và
    ' EndUpdate để tăng hiệu xuất và để cây không bị "rung"
    Me.BeginUpdate()
    Dim tn As TreeNode
    ' Thêm các node con cho mỗi node con trong node được người sử dụng chọn.
    For Each tn In tvcea.Node.Nodes
        AddDirectories(tn)
    Next tn
    Me.EndUpdate()
End Sub

' Thủ tục này được sử dụng để thêm các node con cho một thư mục cha
' (được truyền như một đối số.)
Sub AddDirectories(ByVal tn As TreeNode)
    tn.Nodes.Clear()
    Dim strPath As String = tn.FullPath
    Dim diDirectory As New DirectoryInfo(strPath)
    Dim adiDirectories() As DirectoryInfo

    Try
        ' Lấy thông tin tất cả thư mục con vào các đối tượng DirectoryInfo
        adiDirectories = diDirectory.GetDirectories()
    Catch exp As Exception
        Exit Sub
    End Try

    Dim di As DirectoryInfo
    For Each di In adiDirectories
        ' Tạo một node con cho mọi thư mục con, truyền vào
        ' tên thư mục và các hình ảnh node của nó sẽ sử dụng.
        Dim tnDir As New TreeNode(di.Name, 1, 2)
        ' Thêm một node con mới vào node cha
        tn.Nodes.Add(tnDir)
```

```
        ' Đến đây có thể tạo toàn bộ cây bằng cách gọi đệ quy
        ' AddDirectories(), nhưng cách này khá chậm.
        '   AddDirectories(tnDir)
    Next
  End Sub

  ' Thủ tục này xóa nội dung các đối tượng TreeNode đã tồn tại và
  ' xây dựng lại DirectoryTreeView, hiển thị các đĩa logic.
  Public Sub RefreshTree()
    BeginUpdate()
    Nodes.Clear()

    ' Chọn các đĩa làm các node gốc
    Dim astrDrives As String() = Directory.GetLogicalDrives()

    Dim strDrive As String
    For Each strDrive In astrDrives
      Dim tnDrive As New TreeNode(strDrive, 0, 0)
      Nodes.Add(tnDrive)
      AddDirectories(tnDrive)

    ' Chọn đĩa C làm đĩa mặc nhiên
      If strDrive = "C:\" Then
        Me.SelectedNode = tnDrive
      End If
    Next
    EndUpdate()
  End Sub
End Class
```

FileListView.vb

```
Option Strict On
' Sử dụng cho Process.Start
Imports System.Diagnostics
Imports System.IO

Class FileListView
  Inherits ListView
```

```
Private strDirectory As String
  Public Sub New()
    ' Chọn chế độ mặc nhiên là Details
  Me.View = View.Details

    ' Nhận vào các hình ảnh làm các biểu tượng cho các kiểu file
  Dim img As New ImageList()
  With img.Images
    .Add(New Bitmap("..\ExplorerStyleViewer\DOC.BMP"))
    .Add(New Bitmap("..\ExplorerStyleViewer\EXE.BMP"))
  End With
    ' Danh sách với Small image và Large image sử dụng cùng các hình ảnh
  Me.SmallImageList = img
  Me.LargeImageList = img

    ' Tạo các cột
  With Columns
    .Add("Name", 100, HorizontalAlignment.Left)
    .Add("Size", 100, HorizontalAlignment.Right)
    .Add("Modified", 100, HorizontalAlignment.Left)
    .Add("Attribute", 100, HorizontalAlignment.Left)
  End With
End Sub

' Overrides event handler OnItemActivate. Cho phéo chạy bất kỳ
' chương trình .exe hoặc file có chương trình kết hợp.
Protected Overrides Sub OnItemActivate(ByVal ea As EventArgs)
  MyBase.OnItemActivate(ea)

  Dim lvi As ListViewItem
  For Each lvi In SelectedItems
    Try
      Process.Start(Path.Combine(strDirectory, lvi.Text))
    Catch
        ' Không làm gì cả, tiếp tục với lvi kế tiếp
      Exit Try
    End Try
  Next lvi
End Sub
```

```
' Thủ tục này hiển thị một danh sách của tất cả các file trong
' thư mục đang được chọn trong TreeView.
Public Sub ShowFiles(ByVal strDirectory As String)
    ' Lưu tên của thư mục
    Me.strDirectory = strDirectory
    Items.Clear()
    Dim diDirectories As New DirectoryInfo(strDirectory)
    Dim afiFiles() As FileInfo
    Try
        ' Gọi phương thức GetFile để lấy một mảng tất cả các file trong thư mục
        afiFiles = diDirectories.GetFiles()
    Catch
        Return
    End Try

    Dim fi As FileInfo
    For Each fi In afiFiles
        ' Tạo một ListViewItem.
        Dim lvi As New ListViewItem(fi.Name)

        ' Gán ImageIndex dựa vào phần mở rộng của tên file.
        Select Case Path.GetExtension(fi.Name).ToUpper()
            Case ".EXE"
                lvi.ImageIndex = 1
            Case Else
                lvi.ImageIndex = 0
        End Select

        ' Thêm các thông tin "kích thước" và "ngày thay đổi cuối cùng"
        lvi.SubItems.Add(fi.Length.ToString("N0"))
        lvi.SubItems.Add(fi.LastWriteTime.ToString())

        ' Thêm thông tin thuộc tính
        Dim strAttr As String = ""
        If (fi.Attributes And FileAttributes.Archive) <> 0 Then
            strAttr += "A"
        End If
```

```
        If (fi.Attributes And FileAttributes.Hidden) <> 0 Then
          strAttr += "H"
        End If
        If (fi.Attributes And FileAttributes.ReadOnly) <> 0 Then
          strAttr += "R"
        End If

        If (fi.Attributes And FileAttributes.System) <> 0 Then
          strAttr += "S"
        End If
        lvi.SubItems.Add(strAttr)

        ' Thêm ListViewItem đã hoàn tất vào FileListView
        Items.Add(lvi)
      Next fi
    End Sub
End Class
```

MenuItemView.vb

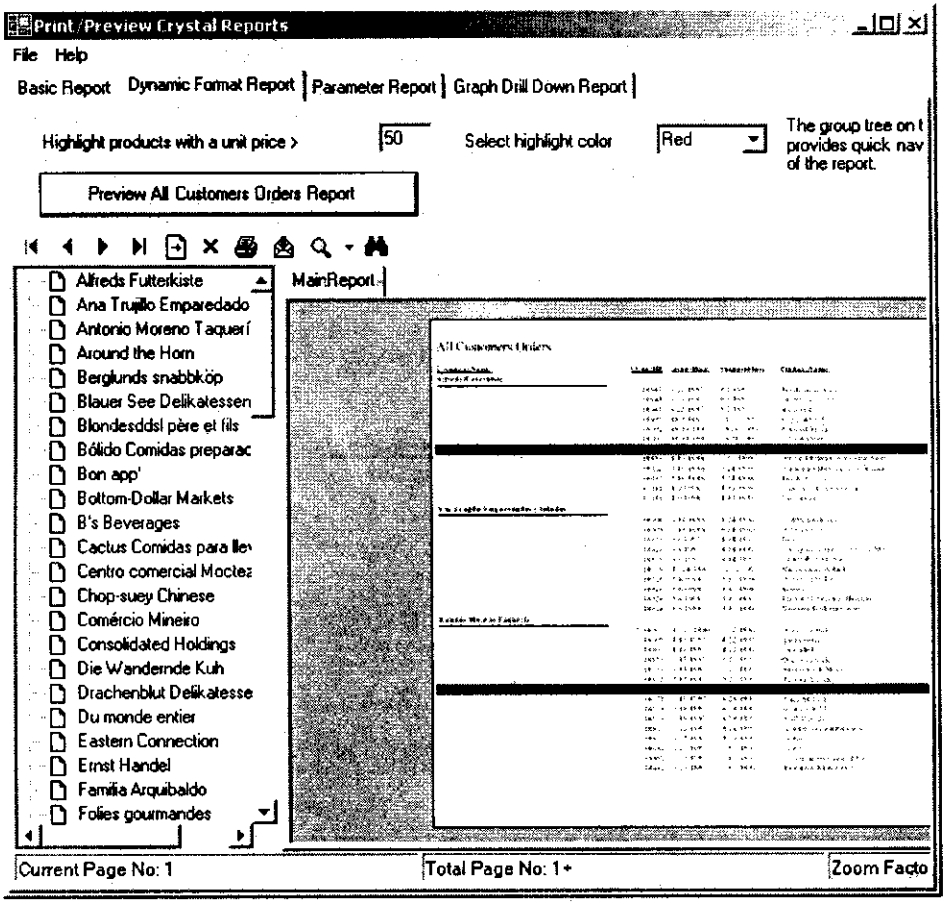
```
Option Strict On
' Class này mở rộng class MenuItem để thêm thuộc tính View
Class MenuItemView
  Inherits MenuItem
  Public View As View
End Class
```

frmMain.vb

```
  Private Sub btnSimple_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSimple.Click
    Dim sdv As New DirectoryScanner()
    sdv.Show()
  End Sub

    Private Sub btnExplorer_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExplorer.Click
    Dim esfv As New ExplorerStyleViewer()
    esfv.Show()
  End Sub
```

3. Crystal Report Print Preview

Chương trình này biểu diễn cách sử dụng trình xem Crystal Report trước khi in.



Các chức năng chính:

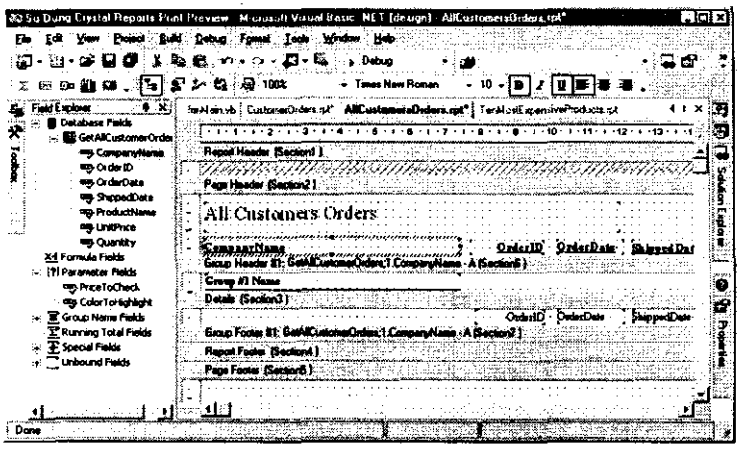
- Cách hiển thị một report (báo cáo) sử dụng Crystal Reports viewer.
- Cách truyền tham số vào một report và hiển thị nó trong viewer.
- Cách sử dụng cây duyệt trong viewer.
- Cách thay đổi report lúc chương trình đang chạy.
- Cách sử dụng chức năng "drill down" dùng một đồ thị.

Yêu cầu:

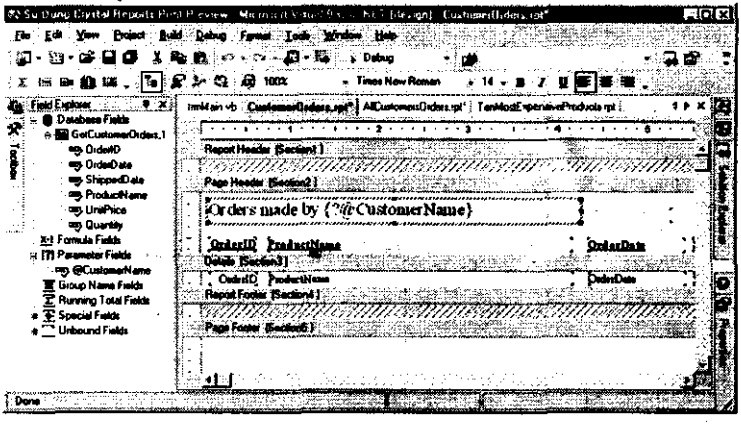
Cần có database Northwind trong SQL Server hoặc MSDE (Microsoft Data Engine, chương trình này có trong phần cài đặt Visual Studio .NET).

Project: **Su Dung Crystal Reports Print Preview**

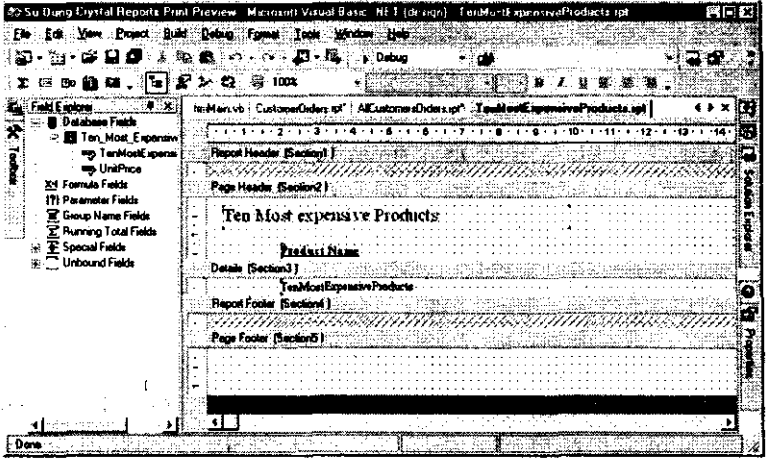
AllCustomersOrders.rpt



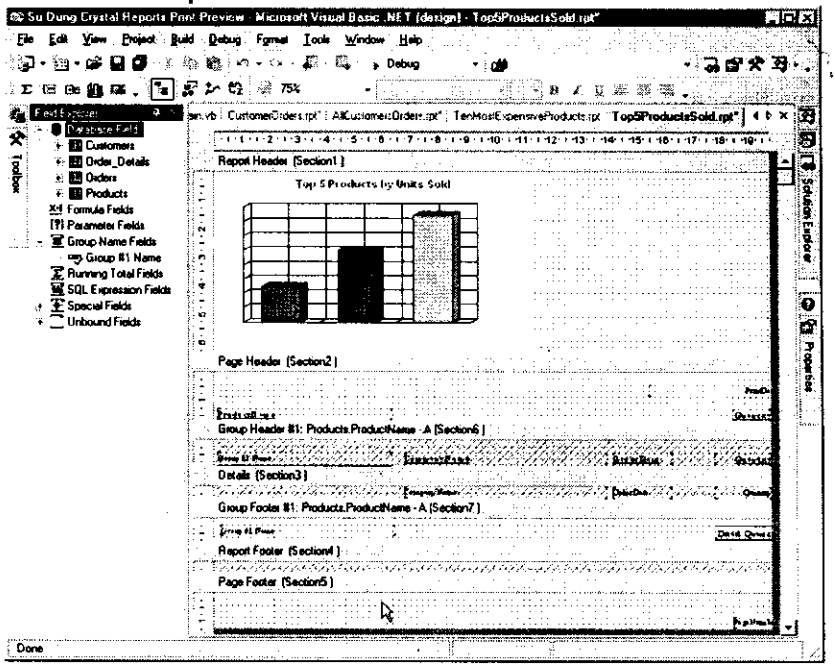
CustomerOrders.rpt



TenMostExpensiveProducts.rpt



Top5ProductsSold.rpt



frmMain.vb

```
Option Strict On
Option Explicit On

Imports System.Data
Imports System.Data.SqlClient
Imports CrystalDecisions.CrystalReports.Engine

Public Class frmMain
    Inherits System.Windows.Forms.Form
    Protected Const MSDE_SERVER As String = "(local)\VSdotNET"

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI;Connect Timeout=5"

    Protected Const MSDE_CONNECTION_STRING As String = _
      "Server=" & MSDE_SERVER & ";" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI;Connect Timeout=5"
```

```
Private ConnectionString As String = SQL_CONNECTION_STRING
Private HasConnected As Boolean = False
Private ServerName As String = "localhost"

Public Sub New()
  MyBase.New()

  InitializeComponent()

  crvBasic.DisplayToolbar = True
  crvParameter.DisplayToolbar = True
  crvDynamicFormat.DisplayToolbar = True
  crvGraphDrillDown.DisplayToolbar = True

  ' Thiết lập combobox chứa danh sách tên khách hàng lấy từ
  ' bảng customers trong database Northwind
  Dim cnSQL As SqlConnection
  Dim cmSQL As SqlCommand

  ' Tạo một đối tượng DataReader để đọc dữ liệu từ đối tượng command
  Dim drSQL As SqlDataReader

  ' Hiển thị một hộp trạng thái báo cho biết chương trình đang kết nối.
  ' Hộp thoại này cần khi thiết lập kết nối, sau khi xác định được MSDE
  ' hoặc SQL Server được cài đặt, không cần đến thông báo này nữa.
  Dim frmStatusMessage As New frmStatus()
  If Not HasConnected Then
    frmStatusMessage.Show("Connecting to SQL Server")
  End If

  ' Tiến hành kết nối đến SQL Server hoặc MSDE
  Dim IsConnecting As Boolean = True
  While IsConnecting
    Try
      ' Định nghĩa chuỗi kết nối. Cần thay đổi nó cho phù hợp
      ' với môi trường chạy ứng dụng này.
      cnSQL = New SqlConnection(ConnectionString)
      cnSQL.Open()
```

```
        ' Đối tượng Command dùng để thực hiện các lệnh SQL
        cmSQL = New SqlCommand()

        ' Kết command vào kết nối
        cmSQL.Connection = cnSQL

        ' Quy định kiểu của command là Text (văn bản)
        cmSQL.CommandType = CommandType.Text

      ' Xóa Stored Procedure (thủ tục lưu trữ nội) GetAllCustomerOrders nếu đã có.
        cmSQL.CommandText = "IF EXISTS (SELECT * FROM dbo.sysobjects
WHERE id = object_id(N'[dbo].[GetAllCustomerOrders]') and
OBJECTPROPERTY(id, N'IsProcedure') = 1) " & _
    "DROP PROCEDURE [dbo].[GetAllCustomerOrders] "

        ' Thực hiện câu lệnh vừa tạo
        cmSQL.ExecuteNonQuery()

        ' Tạo thủ tục lưu trữ nội  GetAllCustomerOrders
        cmSQL.CommandText = _
          "CREATE PROCEDURE dbo.GetAllCustomerOrders " & _
          "AS " & _
          "SELECT CUST.CompanyName, " & _
                    "ORD.OrderID, " & _
                    "ORD.OrderDate, " & _
                    "ORD.ShippedDate, " & _
                    "PROD.ProductName, " & _
                    "ORD_D.UnitPrice, " & _
                    "ORD_D.Quantity " & _
           "FROM Customers CUST " & _
                    "INNER JOIN Orders ORD " & _
                "ON CUST.CustomerID = ORD.CustomerID " & _
                "INNER JOIN [Order Details] ORD_D " & _
                "ON ORD.OrderID = ORD_D.OrderID " & _
                "INNER JOIN Products PROD " & _
                "ON ORD_D.ProductID = PROD.ProductID " & _
                "ORDER BY ORD.OrderDate " & _
                 "Return"
```

```
        ' Thực hiện câu lệnh trên
        cmSQL.ExecuteNonQuery()

        ' Xóa sp GetCustomerOrder
        cmSQL.CommandText = "IF EXISTS (select * from dbo.sysobjects where id
= object_id(N'[dbo].[GetCustomerOrders]') and OBJECTPROPERTY(id,
N'IsProcedure') = 1) " & _
            "DROP PROCEDURE [dbo].[GetCustomerOrders] "

        cmSQL.ExecuteNonQuery()
        ' Tạo SP GetCustomerOrder
        cmSQL.CommandText = _
        "CREATE PROCEDURE dbo.GetCustomerOrders " & _
                "@CustomerName nvarchar(50) " & _
        "AS " & _
        "SELECT ORD.OrderID, " & _
                    "ORD.ShippedDate, " & _
                    "ORD.OrderDate, " & _
                    "PROD.ProductName, " & _
                    "ORD_D.UnitPrice, " & _
                    "ORD_D.Quantity " & _
        "FROM Customers CUST " & _
                "INNER JOIN Orders ORD " & _
                "ON CUST.CustomerID = ORD.CustomerID " & _
                 "INNER JOIN [Order Details] ORD_D " & _
                 "ON ORD.OrderID = ORD_D.OrderID " & _
                 "INNER JOIN Products PROD " & _
                 "ON ORD_D.ProductID = PROD.ProductID " & _
        "WHERE CUST.CompanyName = @CustomerName " & _
                 "ORDER BY ORD.OrderDate " & _
                 "RETURN"

        cmSQL.ExecuteNonQuery()

        ' Câu lệnh lấy tất cả tên công ty trong bảng Customers
        cmSQL.CommandText = "SELECT CompanyName " & _
                 "FROM Customers"

        ' Thực hiện câu lệnh trên và trả dữ liệu vào DataReader
        drSQL = cmSQL.ExecuteReader()
```

```
            ' Đưa tất cả record vào combobox
            Do While drSQL.Read()
              cbCustomers.Items.Add(drSQL("CompanyName").ToString())
            Loop

              ' Chọn phần tử đầu tiên trong combobox
            cbCustomers.SelectedIndex = 0

            IsConnecting = False
            HasConnected = True

              'Đóng kết nối.
            drSQL.Close()
            cnSQL.Close()
              'Giải phóng tài nguyên.
            cnSQL.Dispose()
            cmSQL.Dispose()

          Catch Err As SqlException
            If ConnectionString = SQL_CONNECTION_STRING Then
                ' Không thể kết nối vào SQL Server. Chuyển qua MSDE
              ConnectionString = MSDE_CONNECTION_STRING
              ServerNAme = MSDE_SERVER
              frmStatusMessage.Show("Connecting to MSDE")
            Else
                ' Không thể kết nối vào SQL Server và MSDE
              frmStatusMessage.Close()
              MsgBox("To run this sample you must have SQL Server" _
                  " or MSDE with the Northwind database installed.", _
                  MsgBoxStyle.Critical, "SQL Server/MSDE not found")
                ' Thoát chương trình nếu không có kết nối nào thành công
              End
            End If
          Catch Err As Exception
              ' Thông báo lỗi không phải lỗi SQL
            MsgBox(Err.ToString(), MsgBoxStyle.Critical, "General Error")
          End Try
        End While
      End Sub
```

```
' Code phát sinh form được bỏ qua

Private Sub btnPreviewBasicReport_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnPreviewBasicReport.Click
    ' Trong thủ tục này, report "Ten Most Expensive Products" được
    ' load và hiển thị trong crystal report viewer.

    ' Các đối tượng dùng để gán thông tin kết nối database đúng
    Dim tbCurrent As CrystalDecisions.CrystalReports.Engine.Table
    Dim tliCurrent As CrystalDecisions.Shared.TableLogOnInfo

    ' Tạo một instance ReportDocument lưu giữ report
    Dim rptExpensiveProducts As New ReportDocument()

    Try
        ' Load report
        rptExpensiveProducts.Load("..\TenMostExpensiveProducts.rpt")

' Gán thông tin kết nối cho tất cả các bảng được sử dụng trong report
        For Each tbCurrent In rptExpensiveProducts.Database.Tables
            tliCurrent = tbCurrent.LogOnInfo
            With tliCurrent.ConnectionInfo
                .ServerName = ServerName
                .UserID = ""
                .Password = ""
                .DatabaseName = "Northwind"
            End With
            tbCurrent.ApplyLogOnInfo(tliCurrent)
        Next tbCurrent

        ' Gán report cho crystal report viewer
        crvBasic.ReportSource = rptExpensiveProducts
        ' Thu nhỏ trình hiển thị thành toàn trang để có thể thấy toàn bộ
        crvBasic.Zoom(2)

    Catch Exp As LoadSaveReportException
        MsgBox("Incorrect path for loading report.", _
            MsgBoxStyle.Critical, "Load Report Error")
```

```
    Catch Exp As Exception
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "General Error")

    End Try
End Sub

Private Sub btnPreviewCustomerReport_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnPreviewCustomerReport.Click
    ' Trong thủ tục này Customer Orders Report được load và hiển thị.
    ' Report này được gọi với một tham số được lấy từ combobox cbCustomer.

    ' Các đối tượng dùng để đưa tham số vào report
    Dim pvCollection As New CrystalDecisions.Shared.ParameterValues()
    Dim pdvCustomerName As New _
            CrystalDecisions.Shared.ParameterDiscreteValue()
    Dim tbCurrent As CrystalDecisions.CrystalReports.Engine.Table
    Dim tliCurrent As CrystalDecisions.Shared.TableLogOnInfo

    ' Tạo một instance của ReportDocument
    Dim rptCustomersOrders As New ReportDocument()
    Try
        ' Load report
        rptCustomersOrders.Load("..\CustomerOrders.rpt")
        For Each tbCurrent In rptCustomersOrders.Database.Tables
            tliCurrent = tbCurrent.LogOnInfo
            With tliCurrent.ConnectionInfo
                .ServerName = ServerName
                .UserID = ""
                .Password = ""
                .DatabaseName = "Northwind"
            End With
            tbCurrent.ApplyLogOnInfo(tliCurrent)
        Next tbCurrent

        ' Gán giá trị cụ thể vào pdvCustomerName
        pdvCustomerName.Value = cbCustomers.Text

        ' Cho nó vào tập hợp các tham số
        pvCollection.Add(pdvCustomerName)
```

```
        ' Áp dụng các giá trị tham số hiện hành
        rptCustomersOrders.DataDefinition.ParameterFields( _
            "@CustomerName").ApplyCurrentValues(pvCollection)

        ' Ẩn cây chọn nhóm của report này
        crvParameter.DisplayGroupTree = False

        crvParameter.ReportSource = rptCustomersOrders
        crvParameter.Zoom(2)

    Catch Exp As LoadSaveReportException
        MsgBox("Incorrect path for loading report.", _
            MsgBoxStyle.Critical, "Load Report Error")

    Catch Exp As Exception
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "General Error")
    End Try
End Sub

Private Sub btnPreviewDrillDownReport_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnPreviewDrillDownReport.Click
    ' Load và hiển thị Top 5 Products Report. Report này có một đồ thị có
    ' thể được sử dụng để truy xuất đến chi tiết của report.
    Dim rptDrillDown As New ReportDocument()
    Dim tbCurrent As CrystalDecisions.CrystalReports.Engine.Table
    Dim tliCurrent As CrystalDecisions.Shared.TableLogOnInfo

    Try
        rptDrillDown.Load("..\Top5ProductsSold.rpt")
        For Each tbCurrent In rptDrillDown.Database.Tables
            tliCurrent = tbCurrent.LogOnInfo
            With tliCurrent.ConnectionInfo
                .ServerName = ServerName
                .UserID = ""
                .Password = ""
                .DatabaseName = "Northwind"
            End With
            tbCurrent.ApplyLogOnInfo(tliCurrent)
        Next tbCurrent
```

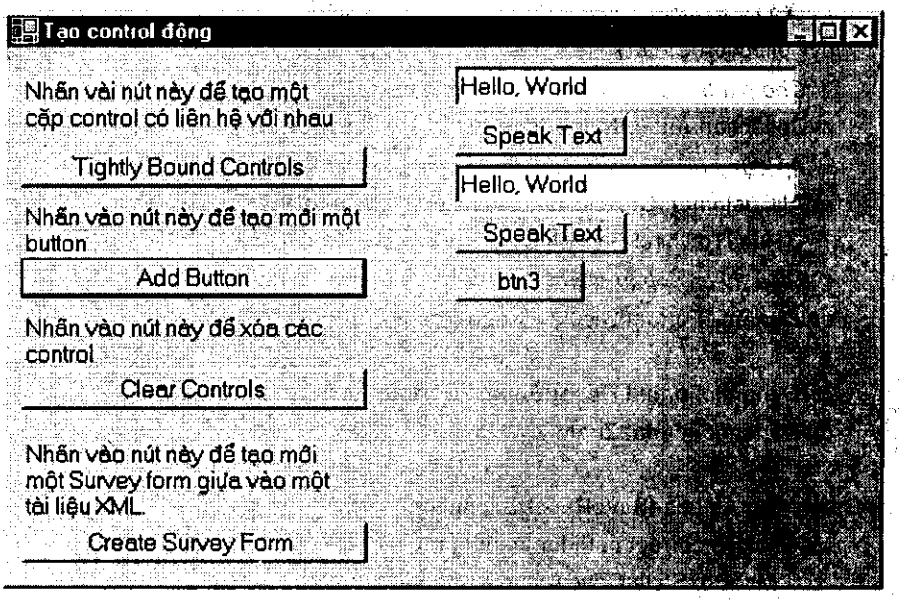
```
        crvGraphDrillDown.ReportSource = rptDrillDown

        crvParameter.DisplayGroupTree = False
        crvGraphDrillDown.Zoom(2)

    Catch Exp As LoadSaveReportException
        MsgBox("Incorrect path for loading report.", _
            MsgBoxStyle.Critical, "Load Report Error")

    Catch Exp As Exception
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "General Error")
    End Try
End Sub

Private Sub btnPreviewDynamicReport_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnPreviewDynamicReport.Click
    ' Trong thủ tục này report All Customers Orders Report được load
    ' và được hiển thị trong crystal report viewer. Trên report này, người sử
    ' dụng có thể quy định giá tiền và màu sắc. Các tham số này được
    ' truyền vào report để xác định dòng nào được làm nổi bật lên bằng
    ' màu được chọn. Trong quá trình thiết kế report một formula được thêm
    ' vào phần details để điều khiển màu nền. Nếu đơn vị giá tiền lớn hơn
    ' giá trị được truyền vào thì chi tiết đó có màu nền là màu được truyền vào.

    ' Đây cũng là report in theo chiều ngang (landscape). Để hiển thị và in một
    ' landscape report đúng, cần tạo một đối tượng report document vào lúc chạy và
    ' sau đó cần phải gán report vào nó.

    Dim rptAllCustomersOrders As New ReportDocument()

    ' Các đối tượng dùng để truyền tham số vào report
    Dim pvCollection As New CrystalDecisions.Shared.ParameterValues()
    Dim pdvColor As New CrystalDecisions.Shared.ParameterDiscreteValue()
    Dim pdvUnitPrice As New CrystalDecisions.Shared.ParameterDiscreteValue()

    Dim tbCurrent As CrystalDecisions.CrystalReports.Engine.Table
    Dim tliCurrent As CrystalDecisions.Shared.TableLogOnInfo
```

```
    ' Quy định các giá trị màu sắc
Dim red As Integer = RGB(255, 0, 0)
Dim green As Integer = RGB(0, 255, 0)
Dim blue As Integer = RGB(0, 0, 255)
If Not IsNumeric(txtUnitsToHighlight.Text) Then
   MsgBox("Please enter a number into the unit price text box.", _
        MsgBoxStyle.Exclamation, Me.Text)
   Exit Sub
End If

Try
    ' Load report
   rptAllCustomersOrders.Load("..\AllCustomersOrders.rpt")

   For Each tbCurrent In rptAllCustomersOrders.Database.Tables
      tliCurrent = tbCurrent.LogOnInfo
      With tliCurrent.ConnectionInfo
         .ServerName = ServerName
         .UserID = ""
         .Password = ""
         .DatabaseName = "Northwind"
      End With
      tbCurrent.ApplyLogOnInfo(tliCurrent)
   Next tbCurrent

    ' Quy định giá trị màu cụ thể
   Select Case cbHighlightColor.Text
      Case "Red"
         pdvColor.Value = red
      Case "Green"
         pdvColor.Value = green
      Case "Blue"
         pdvColor.Value = blue
   End Select

    ' Quy định giá trị của Unit Price (Giá tiền)
   pdvUnitPrice.Value = CInt(txtUnitsToHighlight.Text)

    ' Cho giá trị màu sắc vào tập hợp tham số
   pvCollection.Add(pdvColor)
```

```
        ' Áp dụng giá trị màu sắc
        rptAllCustomersOrders.DataDefinition.ParameterFields( _
            "ColorToHighlight").ApplyCurrentValues(pvCollection)
        ' Xóa các tham số để phục vụ tham số kế tiếp
        pvCollection.Clear()
        ' Cho giá trị unit price vào tập hợp tham số
        pvCollection.Add(pdvUnitPrice)
        ' Áp dụng giá trị unit price
        rptAllCustomersOrders.DataDefinition.ParameterFields( _
            "PriceToCheck").ApplyCurrentValues(pvCollection)
        ' Hiển thị cây phân nhóm
        crvDynamicFormat.DisplayGroupTree = True

        crvDynamicFormat.ReportSource = rptAllCustomersOrders
        crvDynamicFormat.Zoom(2)

    Catch Exp As LoadSaveReportException
        MsgBox("Incorrect path for loading report.", _
            MsgBoxStyle.Critical, "Load Report Error")

    Catch Exp As Exception
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "General Error")
    End Try

  End Sub

End Class
```

4. Tạo control động

Chương trình này chỉ cách tạo control bằng cách lập trình trong Visual Basic .NET.



Các chức năng chính

Chương trình này biểu diễn nhiều tính năng quan trọng trong việc tạo các control động (dynamically creating controls), kết hợp với các event handler thích hợp và cho chúng vào tập hợp các Control của một Form. Các tính năng bao gồm:

- Tạo các Windows control, và thay đổi thuộc tính của chúng.
- Thêm một control đã tạo vào tập hợp các control.
- Kết hợp các sự kiện trong control mới tạo vào các event handler (thủ tục xử lý sự kiện) đã có.

Project: **Tao Control Dong**

frmMain.vb

```
· Option Strict On
· Imports System.Xml
· Public Class frmMain
·     Inherits System.Windows.Forms.Form
```

```
' Khai báo các hằng số để sử dụng trong form
Private Const CONTROL_WIDTH As Integer = 300
Private Const CHARS_PER_LINE As Integer = 30
Private Const HEIGHT_PER_LINE As Integer = 19

' Các biến class
Private m_ControlCount As Integer = 0
Private m_Location As New Point(10, 10)

' Code phát sinh form được bỏ qua

' Thủ tục này thêm một button vào form, và quy định các event
' handler cho nó, sẽ được gọi khi có sự kiện Click và MouseHover
Private Sub btnAddButton_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddButton.Click
    ' Tăng biến đếm control
    m_ControlCount += 1
    ' Chỉ cho phép 5 control để đơn giản việc vẽ giao diện
    If m_ControlCount <= 5 Then
        ' Tạo một button mới
        Dim x As New Button()

        ' gán các thuộc tính vào button
        x.Name = "btn" + m_ControlCount.ToString()
        x.Text = "btn" + m_ControlCount.ToString()
        x.Location = New Point(Me.m_Location.X + 250, Me.m_Location.Y)
        m_Location.Y += x.Height + 5

        ' Thêm hai event handler
        AddHandler x.Click, AddressOf myButtonHandler_Click
        AddHandler x.MouseHover, AddressOf myButtonHandler_MouseHover
        ' Cho control vào tập hợp các control
        Controls.Add(x)
    Else
        ' Chỉ cho phép 5 control
        MsgBox("You've reached 5 controls. Clear controls to start again.", _
                MsgBoxStyle.OKOnly, Me.Text)
    End If
End Sub
```

```
' Thủ tục này xóa tất cả các control phát sinh tự động. Nó thực hiện
' điều này bằng cách loại bỏ tất cả các control, sau đó gọi thủ tục
' InitializeComponent(), đây là cách dễ nhất để đưa một form về trạng thái gốc.
Private Sub btnClearControls_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearControls.Click

    ' Xóa tất cả các control
    Controls.Clear()
    ' Tạo lại tất cả các control gốc
    InitializeComponent()
    ' khởi tạo lại m_Location bằng vị trí gốc.
    m_Location = New Point(10, 10)
    ' Khởi tạo lại số các control
    m_ControlCount = 0
    ' Hiển thị lại form
    Show()

End Sub

' Thủ tục này xử lý sự kiện btnCreateSurvey.Click và tạo một
' frmSurveyForm mới. Các control được phát sinh và thêm vào form
' survey vừa tạo. Không có thủ tục xử lý sự kiện kết hợp với các
' control mới tạo.

' Form vừa tạo khá tổng quát, nó được tạo dựa trên một tài liệu XML
' Question.xml. Bằng cách thay đổi, thêm, hoặc xóa các node trong
' tài liệu xml, ta có thể thay đổi cấu trúc của form survey.

Private Sub btnCreateSurvey_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCreateSurvey.Click

    ' Tạo mới một Survey Form để hiển thị cho người sử dụng
    Dim survey As New frmSurveyForm()

    ' Lấy tập hợp các control của form Survey
    Dim surveyControls As Control.ControlCollection = survey.SurveyFormControls
    ' Khởi tạo lại m_Location trong trường hợp tạo nhiều form
    m_Location = New Point(10, 10)
```

```
' Tạo một XML document để đọc các câu hỏi trong file xml
Dim xr As New Xml.XmlDocument()
xr.Load("..\Questions.xml")
' Lấy Tag được sử dụng cho mỗi control trong các control ta sẽ tạo
Dim myTag As String = xr.SelectSingleNode("//survey").Attributes("name").Value

' Gán thanh tiêu đề của form survey bằng Display Name
survey.SurveyTitle = _
            xr.SelectSingleNode("//survey").Attributes("displayName").Value

' Tạo một XmlNodeList chứa các câu hỏi.
Dim nodeList As Xml.XmlNodeList
nodeList = xr.GetElementsByTagName("question")

' Tạo một node XML tạm thời để sử dụng khi lấy thông tin về các
' node trong nodeList vừa tạo.
Dim myNode As XmlNode
For Each myNode In nodeList
  If Not myNode.Attributes Is Nothing Then
    ' Xác định kiểu của control sẽ tạo
    Select Case myNode.Attributes("type").Value
      Case "dropdown"
        m_Location = Survey_AddComboBox(myNode, _
                     surveyControls, m_Location, myTag)
      Case "multilist"
        m_Location = Survey_AddListBox(myNode, _
                     surveyControls, m_Location, myTag, True)
      Case "text"
        m_Location = Survey_AddTextBox(myNode, _
                     surveyControls, m_Location, myTag)
      Case "radio"
        m_Location = Survey_AddRadioButtons(myNode, _
                     surveyControls, m_Location, myTag)
    End Select
  End If
Next
```

```
    ' Qui định kích thước của form, dựa vào số lượng các control
    ' được đặt trên form, và kích thước của chúng.
    survey.Width = m_Location.X + CONTROL_WIDTH + 30
    ' Thêm một khoảng trống để đặt nút OK và Cancel
    survey.Height = m_Location.Y + 75

    ' Hiển thị form. Có thể dùng Show() nếu muốn.
    survey.ShowDialog()
    ' Hiển thị câu trả lời cho người sử dụng
    MsgBox(survey.SurveyResponse, MsgBoxStyle.OKOnly, Me.Text)
End Sub

' Thủ tục này xử lý sự kiện Click của btnTightlyBoundControls và tạo
' hai control có liên hệ chặt chẽ. Nó sử dụng các event handler đã
' được định nghĩa trước để xử lý các sự kiện. Các event handler này
' phải được định nghĩa sẵn sàng trước, trừ khi sử dụng
' Reflection.Emit. Hai control được tạo là một Button và một TextBox.
' Khi Button được nhấn, nội dung trong TextBox được hiển thị trong
' một MsgBox. Để bảo đảm là ta biết TextBox nào được sử dụng, nó
' được thêm vào thuộc tính Tag của Button (Ta không biết tên của
' TextBox trong lúc tạo event handler, bởi vì TextBox được tạo một
' cách tự động)
Private Sub btnTightlyBoundControls_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnTightlyBoundControls.Click

    ' Tăng số đếm các control (chỉ cộng một, mặc dù hai control được thêm vào)
    m_ControlCount += 1

    ' Chỉ cho phép 5 control để đơn giản việc vẽ giao diện
    If m_ControlCount <= 5 Then

        ' Tạo TextBox có chứa văn bản để hiển thị
        Dim txtSpeakText As New TextBox()

        ' Quy định một số thuộc tính của TextBox
        txtSpeakText.Text = "Hello, World"
        txtSpeakText.Name = "txtSpeakText"
        txtSpeakText.Location = New Point(m_Location.X + 250, m_Location.Y)
        txtSpeakText.Size = New Size(200, txtSpeakText.Height)
```

```
        ' Cho TextBox vào tập hợp các control
        Controls.Add(txtSpeakText)

        ' Tăng m_LocationY để control kế tiếp không chồng lên nó
        m_Location.Y += txtSpeakText.Height + 5

        ' Tạo một button sẽ được sử dụng tương tác với nó.
        ' Thêm TextBox đã tạo ở trên làm một Tag của Button này.
        Dim btnSpeakText As New Button()

        ' Qui định một số thuộc tính cho Button
        btnSpeakText.Text = "Speak Text"
        btnSpeakText.Name = "btnSpeakText"
        btnSpeakText.Location = New Point(m_Location.X + 250, m_Location.Y)
        btnSpeakText.Size = New Size(100, btnSpeakText.Height)

        ' Thêm TextBox vào button này
        btnSpeakText.Tag = txtSpeakText

        ' Thêm button vào tập hợp các control
        Controls.Add(btnSpeakText)

        m_Location.Y += btnSpeakText.Height + 5

        ' Thêm event hadler xử lý sự kiện khi button được nhấn.
        AddHandler btnSpeakText.Click, AddressOf SpeakTextClickHandler
    Else
        MsgBox("You've reached 5 controls. Clear controls to start again.", _
                MsgBoxStyle.OKOnly, Me.Text)
    End If
End Sub

' Thủ tục này xử lý các sự kiện Click của tất cả các button (được tạo
' tự động). Nó được kết vào tất cả các button bằng cách sử
' dụng hàm AddHandler vào lúc tạo button.
Private Sub myButtonHandler_Click(ByVal sender As Object, ByVal e As
EventArgs)
    If TypeOf sender Is Button Then
        ' Cho người sử dụng biết Button nào đã được nhấn
```

```
        MsgBox(CType(sender, Button).Text + " was pressed!", _
            MsgBoxStyle.OKOnly, Me.Text)
    End If
End Sub

' Thủ tục này xử lý các sự kiện MouseHover của tất cả các button
' được tạo một cách tự động. Nó được kết vào các button bằng
' cách sử dụng hàm AddHandler vào lúc tạo button.
Private Sub myButtonHandler_MouseHover(ByVal sender As Object, ByVal e As
EventArgs)
    If TypeOf sender Is Button Then
        ' Cho người sử dụng biết Button nào đã được di chuyển chuột qua
        MsgBox(CType(sender, Button).Text + " was hovered over!", _
            MsgBoxStyle.OKOnly, Me.Text)
    End If
End Sub

' Thủ tục này xử lý sự kiện Click của Button được tạo trong ví dụ các control có
' liên hệ chặt chẽ. Nó hiển thị nội dung của TextBox trong Tag của Button
' bằng một MsgBox.
Private Sub SpeakTextClickHandler(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    If TypeOf sender Is Button Then
        Dim myButton As Button = CType(sender, Button)
        ' Kiểm tra xem Button có thuộc tính Tag là một TextBox
        If TypeOf myButton.Tag Is TextBox Then
            ' Hiển thị văn bản trong TextBox
            MsgBox(CType(myButton.Tag, TextBox).Text, _
                MsgBoxStyle.OKOnly, Me.Text)
        End If
    End If
End Sub

' Hàm này thêm một ComboBox vào tập hợp control,
' kèm với một control Label để hiển thị câu hỏi.
Private Function Survey_AddComboBox(ByVal inNode As XmlNode, _
    ByVal inControls As Control.ControlCollection, _
    ByVal location As Point, ByVal tag As String) As Point
```

```
    ' Trả về vị trí cho control kế tiếp
    Return location
End Function

' Hàm này thêm một ListBox vào tập hợp các control, kèm
' với một Label kết hợp với nó để  hiển thị câu hỏi.
Private Function Survey_AddListBox(ByVal inNode As XmlNode, _
    ByVal inControls As Control.ControlCollection, _
    ByVal location As Point, ByVal tag As String, _
    ByVal isMultiSelect As Boolean) As Point

    ' Tạo mới một control
    Dim myList As New ListBox()

    myList.Text = ""
    myList.Name = inNode.Attributes("name").Value
    myList.Tag = tag
    myList.Width = CONTROL_WIDTH

    ' Bởi vì hàm này có thể được sử dụng với các ListBox cho phép
    ' chọn nhiều phần tử hoặc chỉ chọn một phần tử, quy định thuộc
    ' tính SelectionMode dựa vào biến isMultiSelect được truyền vào.
    If isMultiSelect Then
        myList.SelectionMode = SelectionMode.MultiSimple
    Else
        myList.SelectionMode = SelectionMode.One
    End If

    Dim myNode As XmlNode
    ' Cho InnerText của các node response vào danh sách
    For Each myNode In inNode.SelectNodes("responses/response")
        myList.Items.Add(myNode.InnerText)
    ' Nếu có chỉ định giá trị mặc nhiên, sử dụng nó làm giá trị hiện hành
        If Not myNode.Attributes("default") Is Nothing Then
            If myNode.Attributes("default").Value = "true" Then
                myList.Text = myNode.InnerText
            End If
        End If
    Next
```

```
    ' Tạo một Label và cho nó và tập hợp
    Dim myLabel As New Label()

    myLabel.Name = myList.Name & "Label"
    myLabel.Text = inNode.SelectSingleNode("text").InnerText
    myLabel.Width = CONTROL_WIDTH

    myLabel.Location = location
    inControls.Add(myLabel)
    location.Y += myLabel.Height

    myList.Location = location
    inControls.Add(myList)
    location.Y += myList.Height + 10

    Return location
End Function

' Hàm này thêm một GroupBox vào tập hợp control được truyền vào,
' cùng với các radio button thích hợp, mỗi radio button tương ứng cho
' mỗi trả lời. Nó cũng tạo một Label để hiển thị câu hỏi.
Private Function Survey_AddRadioButtons(ByVal inNode As XmlNode, ByVal
inControls As Control.ControlCollection, _
    ByVal location As Point, ByVal tag As String) As Point

    ' Phải tạo một GroupBox để chứa các radio button để nhóm chúng
    ' và tách biệt với các radio button khác trên form.
    Dim myGroupBox As New GroupBox()

    myGroupBox.Text = ""
    myGroupBox.Name = inNode.Attributes("name").Value
    myGroupBox.Tag = tag
    myGroupBox.Width = CONTROL_WIDTH + 20

    ' Tạo các biến cần sử dụng
    Dim myRadio As RadioButton
    Dim myRadioPoint As New Point(5, 10)
    Dim myNode As XmlNode
```

```
    ' Duyệt qua các câu trả lời, và cho nó vào form như là một radio button mới
    For Each myNode In inNode.SelectNodes("responses/response")
        ' Tạo radio button
        myRadio = New RadioButton()

        ' Quy định các thuộc tính thích hợp
        myRadio.Text = myNode.InnerText
        myRadio.Location = myRadioPoint
        myRadioPoint.Y += myRadio.Height

        ' Quy định giá trị mặc nhiên cho radio button, nhưng chỉ khi
        ' thuộc tính "default" tồn tại và nó có giá trị là true
        If Not myNode.Attributes("default") Is Nothing Then
            If myNode.Attributes("default").Value = "true" Then
                myRadio.Checked = True
            End If
        End If
        ' Thêm radio button vào GroupBox
        myGroupBox.Controls.Add(myRadio)
    Next
    myGroupBox.Height = myRadioPoint.Y + 5

    'Tạo một Label và cho nó vào tập hợp.
    Dim myLabel As New Label()

    myLabel.Name = myGroupBox.Name & "Label"
    myLabel.Text = inNode.SelectSingleNode("text").InnerText
    myLabel.Width = CONTROL_WIDTH

    myLabel.Location = location
    inControls.Add(myLabel)
    location.Y += myLabel.Height - 5

    myGroupBox.Location = location
    inControls.Add(myGroupBox)
    location.Y += myGroupBox.Height + 10

    Return location
End Function
```

```
' Hàm này thêm một TextBox và một Label
Private Function Survey_AddTextBox(ByVal inNode As XmlNode, _
  ByVal inControls As Control.ControlCollection, _
  ByVal location As Point, ByVal tag As String ) As Point

  Dim myText As New TextBox()
  If Not inNode.SelectSingleNode("defaultResponse") Is Nothing Then
    myText.Text = inNode.SelectSingleNode("defaultResponse").InnerText
  End If

  If Not inNode.Attributes("name") Is Nothing Then
    myText.Name = inNode.Attributes("name").Value
  End If

  myText.Tag = tag
  myText.Width = CONTROL_WIDTH
  If Not inNode.SelectSingleNode("maxCharacters") Is Nothing Then
    myText.MaxLength = _
        Integer.Parse(inNode.SelectSingleNode("maxCharacters").InnerText)
  End If

  If myText.MaxLength > 0 Then
    Dim numLines As Integer = (myText.MaxLength \ CHARS_PER_LINE) + 1

    If numLines = 1 Then
      myText.Multiline = False
    Else
      If numLines >= 4 Then
        myText.Multiline = True
        myText.Height = 4 * HEIGHT_PER_LINE
        myText.ScrollBars = ScrollBars.Vertical
      Else
        myText.Multiline = True
        myText.Height = numLines * HEIGHT_PER_LINE
        myText.ScrollBars = ScrollBars.None
      End If
    End If
  End If
```

```
        Dim myLabel As New Label()
        myLabel.Name = myText.Name & "Label"
        If Not inNode.SelectSingleNode("text") Is Nothing Then
            myLabel.Text = inNode.SelectSingleNode("text").InnerText
        End If
        myLabel.Width = CONTROL_WIDTH

        myLabel.Location = location
        inControls.Add(myLabel)
        location.Y += myLabel.Height

        myText.Location = location
        inControls.Add(myText)
        location.Y += myText.Height + 10

        Return location
    End Function
End Class
```

frmSurveyForm.vb

```
Option Explicit On

' Class này tương ứng một Survey form. Tuy nhiên, tất cả các control
' được tạo tự động trong code
Public Class frmSurveyForm
    Inherits System.Windows.Forms.Form

    Private m_Title As String = "Survey"
    Private m_SurveyResponse As String = "Survey Not Completed"

' Code phát sinh form được bỏ qua

    ' Tạo một thuộc tính để có thể dễ dàng lấy tập hợp các control
    Public ReadOnly Property SurveyFormControls() As Control.ControlCollection
        Get
            Return Me.Controls
        End Get
    End Property
```

```
' Tạo thuộc tính height để dễ dàng thay đổi chiều cao
Public Property SurveyHeight() As Integer
  Get
    Return Me.Height
  End Get

  Set(ByVal Value As Integer)
    Me.Height = Value
  End Set
End Property

' Tạo một thuộc tính để dễ dàng lấy trả lời
Public ReadOnly Property SurveyResponse() As String
  Get
    Return m_SurveyResponse
  End Get
End Property

' Thuộc tính thay đổi thanh tiêu đề
Public Property SurveyTitle() As String
  Get
    Return m_Title
  End Get
  Set(ByVal Value As String)
    m_Title = Value
    Me.Text = m_Title
  End Set
End Property

Public Property SurveyWidth() As Integer
  Get
    Return Me.Width
  End Get
  Set(ByVal Value As Integer)
    Me.Width = Value
  End Set
End Property
```

```
' Thủ tục này chỉ khởi tạo lại chuỗi SurveyResponse và đóng form.
Private Sub btnCancel_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnCancel.Click
    m_SurveyResponse = "Survey Not Completed"
    Me.Close()
End Sub

' Nút này thiết lập chuỗi SurveyResponse và đóng form
Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOK.Click
    ' Tạo các control sẽ dùng trong các vòng lặp
    Dim myControl As Control
    Dim myGroupControl As Control
    Dim myGroupBox As GroupBox
    Dim myObject As Object
    Dim myRadio As RadioButton
    ' Xóa nội dung trong chuỗi SurveyResponse
    Me.m_SurveyResponse = ""
' Duyệt qua từng control và đưa trả lời từ người sử dụng vào chuỗi SurveyResponse.
    For Each myControl In Me.Controls
      Select Case TypeName(myControl)
        Case "ComboBox"
          m_SurveyResponse += myControl.Name + " - "
          m_SurveyResponse += myControl.Text
          m_SurveyResponse += vbCrLf
        Case "TextBox"
          m_SurveyResponse += myControl.Name + " - "
          m_SurveyResponse += myControl.Text
          m_SurveyResponse += vbCrLf
        Case "GroupBox"
          For Each myGroupControl In CType(myControl, GroupBox).Controls
           If TypeOf myGroupControl Is RadioButton Then
             If CType(myGroupControl, RadioButton).Checked Then
                m_SurveyResponse += myControl.Name + " - "
                m_SurveyResponse += myGroupControl.Text
                m_SurveyResponse += vbCrLf
             End If
           End If
          Next
```

```
            Case "ListBox"
                m_SurveyResponse += myControl.Name + " - "
                For Each myObject In CType(myControl, ListBox).SelectedItems
                    If TypeOf myObject Is String Then
                        m_SurveyResponse += vbTab + CStr(myObject)
                        m_SurveyResponse += vbCrLf
                    End If
                Next
            End Select
        Next
        Me.Close()
    End Sub
End Class
```

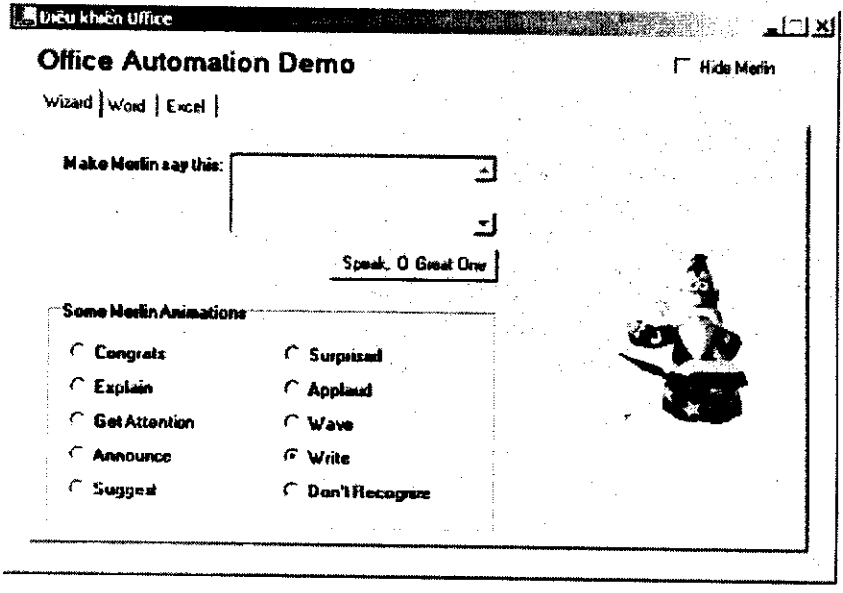
Questions.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<survey name="SurveyApp" displayName="Personal Survey">
  <description>This survey is a survey of personal information.</description>
  <question type="dropdown" name="FruitQuestion">
   <text>Which fruit is orange and circular?</text>
   <responses>
     <response default = 'false'>Apple</response>
     <response default = 'false'>Banana</response>
     <response default = 'false'>Orange</response>
     <response default = 'false'>Kiwi</response>
   </responses>
  </question>
  <question type="dropdown" name="SkyQuestion">
   <text>What color is the sky?</text>
   <responses>
     <response default = 'true'>Blue</response>
     <response default = 'false'>Red</response>
     <response default = 'false'>Orange</response>
     <response default = 'false'>Yellow</response>
   </responses>
  </question>
  <question type="text" name="MusicianQuestion">
   <text>Describe your favorite type of music.</text>
   <defaultResponse></defaultResponse>
```

```
    <maxCharacters>75</maxCharacters>
   </question>
   <question type="multilist" name="SoupQuestion">
    <text>Which soups do you like? (Select all that apply)</text>
    <responses>
      <response>Clam Chowder</response>
      <response default="true">Tomato</response>
      <response>Chicken Noodle</response>
      <response>Split Pea</response>
      <response>Cream of Mushroom</response>
    </responses>
   </question>
   <question type="radio" name="GenderQuestion">
    <text>What is your gender?</text>
    <responses>
      <response>Male</response>
      <response>Female</response>
    </responses>
   </question>
</survey>
```

5. Điều khiển Office

Chương trình này chỉ cách điều khiển Office từ .NET.



Các chức năng chính:

Chương trình này làm việc với:

- **Microsoft Agent**: Cách làm cho Merlin xuất hiện trong ứng dụng và làm một số thứ như nói, di chuyển, và hiển thị các hoạt hình của nó. Merlin và các đặc tính Office Agent khác là một cách tốt để cung cấp Help cho ứng dụng.
- **Microsoft Word**: Tạo một ứng dụng Word và gọi trình kiểm lỗi chính tả để kiểm lỗi cho một tài liệu trong một RichTextBox.
- **Microsoft Excel**: Tạo và nhập dữ liệu từ một tài liệu XML vào một DataSet và sau đó kết nó vào một DataGrid. Sau đó xuất nội dung của DataSet ra một Excel spreadsheet và chạy một hàm của Excel để tính giá trị trung bình của một cột.

Nếu muốn nghe được Merlin nói, máy tính cần cài đặt Speech Recognition, chức năng này có trong Microsoft Office XP nhưng không được mặc nhiên cài đặt.

Project: **Dieu Khien Office**

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private agentController As AgentObjects.Agent
    Private agentCharacter As AgentObjects.IAgentCtlCharacter
    Private dsMenu As DataSet

    ' Xử lý nút Browser trên tab Word, cho phép người sử dụng chọn
    ' một tài liệu văn bản (.txt) và đưa nội dung của nó vào RichTextBox.
    Private Sub btnBrowseWord_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnBrowseWord.Click
        With dlgOpenWordFile
            .InitialDirectory = "C:\"
            .Filter = "Text Format (*.txt)|*.txt"
            .FilterIndex = 1
```

```
    If .ShowDialog() = DialogResult.OK Then
        Try
            rtfDocument.LoadFile(.FileName, RichTextBoxStreamType.PlainText)
        Catch exp As ArgumentException
            MessageBox.Show("The document you are attempting to load " & _
                "is not in the correct format.", "Document Load Error", _
                MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End If
  End With
End Sub

' Xử lý nút "Export to Excel>>", cho phép người sử dụng xuất nội
' dung của DataSet qua một Excel spreadsheet và sau đó chạy hàm
' Average để xác định giá trị trung bình.
Private Sub btnExport_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnExport.Click
    ' Một Excel spreadsheet liên quan đến một cây phân cấp các đối
    ' tượng, từ Application đến Workbook đến Worksheet.
  Dim excelApp As New Excel.Application()
  Dim excelBook As Excel.Workbook = excelApp.Workbooks.Add

  Dim excelWorksheet As Excel.Worksheet = _
            CType(excelBook.Worksheets(1), Excel.Worksheet)

    ' Hiển thị trang tính cho người sử dụng thấy dữ liệu được nhập vào
  excelApp.Visible = True
  With excelWorksheet
      ' Quy định các dòng đầu (header) và định dạng trước dữ liệu
    .Columns().ColumnWidth = 21.71
    .Range("A1").Value = "Item"
    .Range("A1").Font.Bold = True
    .Range("B1").Value = "Price"
    .Range("B1").Font.Bold = True
    .Range("C1").Value = "Calories"
    .Range("C1").Font.Bold = True

      ' Bắt đầu biến đếm từ dòng thứ hai, sau dòng header
    Dim i As Integer = 2
```

```
        ' Duyệt qua tập hợp các dòng (Rows) của DataSet và
        ' ghi dữ liệu trong mỗi dòng vào các ô trong Excel.
        Dim dr As DataRow
        For Each dr In dsMenu.Tables(0).Rows
            .Range("A" & i.ToString).Value = dr("Item")
            .Range("B" & i.ToString).Value = dr("Price")
            .Range("C" & i.ToString).Value = dr("Calories")
            i += 1
        Next

        ' Chọn và áp dụng định dạng cho ô hiển thị giá trị trung bình,
        ' sau đó gọi hàm Average.
        .Range("C7").Select()
        .Range("C7").Font.Color = RGB(255, 0, 0)
        .Range("C7").Font.Bold = True
        excelApp.ActiveCell.FormulaR1C1 = "=AVERAGE(R[-5]C:R[-1]C)"
    End With
End Sub

' Xử lý nút "Get Menu", tạo và nhập dữ liệu vào DataSet từ một tài
' liệu XML, đưa dữ liệu vào một DataGrid, sau đó áp dụng định dạng cho DataGrid.
Private Sub btnGetMenu_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetMenu.Click
    ' Nhập dữ liệu vào DataSet bằng cách lấy từ một file XML với các
    ' phần tử là một thực đơn.
    dsMenu = New DataSet()
    dsMenu.ReadXml("..\menu.xml")
    ' Qui định đầu đề của DataGrid và kết nó vào DataSet.
    With grdMenu
        .CaptionText = "Today's Menu"
        .DataSource = dsMenu.Tables(0)
    End With

    ' Sử dụng một đối tượng kiểu table để áp dụng định dạng tùy biến cho DataGrid
    Dim grdTableStyle1 As New DataGridTableStyle()
    With grdTableStyle1
        .AlternatingBackColor = Color.Lavender
        .BackColor = Color.WhiteSmoke
        .ForeColor = Color.MidnightBlue
```

```
        .GridLineColor = Color.Gainsboro
        .GridLineStyle = System.Windows.Forms.DataGridLineStyle.None
        .HeaderBackColor = Color.MidnightBlue
        .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
        .HeaderForeColor = Color.WhiteSmoke
        .LinkColor = Color.Teal
        ' Đừng quên quy định thuộc tính MappingName. Nếu không
        ' có nó, các thuộc tính DataGridTableStyle và bất kỳ đối tượng
        ' DataGridColumnStyle nào được kết hợp sẽ không có ảnh hưởng.
        .MappingName = "Food"
        .SelectionBackColor = Color.CadetBlue
        .SelectionForeColor = Color.WhiteSmoke
    End With

    ' Sử dụng các đối tượng kiểu Column để định dạng từng cột
    Dim grdColStyle1 As New DataGridTextBoxColumn()
    With grdColStyle1
        .HeaderText = "Item"
        .MappingName = "Item"
        .Width = 125
    End With

    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
        .HeaderText = "Price"
        .MappingName = "Price"
        .Width = 50
    End With

    Dim grdColStyle3 As New DataGridTextBoxColumn()
    With grdColStyle3
        .HeaderText = "Calories"
        .MappingName = "Calories"
        .Width = 70
    End With
    grdTableStyle1.GridColumnStyles.AddRange (New DataGridColumnStyle() _
                                        {grdColStyle1, grdColStyle2, grdColStyle3})
    grdMenu.TableStyles.Add(grdTableStyle1)
    btnExport.Visible = True
End Sub
```

```
' Thủ tục xử lý sự kiện Click của button Speak, làm cho Merlin nói bất cứ nội
' dung nào do người sử dụng nhập vào TextBox
Private Sub btnSpeak_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnSpeak.Click

    agentCharacter.Speak(txtSpeak.Text)
End Sub

' Thủ tục xử lý nút Spellcheck, cho phép người sử dụng chạy
' chương trình kiểm lỗi để kiểm lỗi văn bản trong RichTextBox.
' Chú ý là không cần phải nhìn thấy Word.
Private Sub btnSpellCheck_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSpellCheck.Click
    Dim wordApp As New Word.Application()
    Dim bolHasNoSpellingErrors As Boolean = _
        wordApp.CheckSpelling(rtfDocument.Text)
   Dim strSpellCheck As String = "Your document has spelling errors."

    If bolHasNoSpellingErrors Then
        strSpellCheck = "Congratulations, your document is free of spelling errors."
    End If
    MessageBox.Show(strSpellCheck, "Spell Check Results", _
        MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub

' Xử lý checkbox hiển thị hay ẩn Merlin.
Private Sub chkMerlinOnOff_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkMerlinHide.CheckedChanged
    With agentCharacter
        If chkMerlinHide.Checked Then
            .StopAll()
            .Hide()
        Else
            .Show()
        End If
    End With
End Sub
```

```
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
  ' Đối tượng Agent được sử dụng để mở một kết nối đến Agent server
  agentController = New AgentObjects.Agent()
  With agentController
    .Connected = True
    .Characters.Load("merlin", "merlin.acs")
    agentCharacter = .Characters("merlin")
  End With

  ' Sử dụng thuộc tính Location để đặt Merlin
  With agentCharacter
   .MoveTo(CShort(Me.Location.X + 420), CShort(Me.Location.Y + 130))
    .Show()
    .Play("Announce")
    .Speak("Hello, my name is Merlin. " & _
      "Welcome to the Office Automation Demo!")
    .Play("GestureRight")
     ' Có thể làm cho dọng nói của Merlin tự nhiên hơn bằng
     ' cách chèn các tag như Pau, Chr, Emp hoặc Spd
    .Speak("Make me say something.\pau=300\or\pau=500\...")
    .MoveTo(CShort(Me.Location.X + 340), CShort(Me.Location.Y + 250))
    .Play("GestureRight")
    .Speak("...try out some of my animations.")
  End With

  ' Ẩn nút "Export to Excel" cho đến khi DataGrid được kết dữ liệu
  btnExport.Visible = False
End Sub

' Xử lý các sự kiện click cho tất cả các kiểu hoạt hình của Mérlin
Private Sub rdoAnimations_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles optDontRecognize.Click, optWrite.Click,
optWave.Click, optApplaud.Click, optSurprised.Click, optSuggest.Click,
optAnnounce.Click, optDoMagic.Click, optExplain.Click, optCongrats.Click

  agentCharacter.StopAll()
  Dim radio As RadioButton = CType(sender, RadioButton)
  With agentCharacter
    Select Case radio.Tag
```

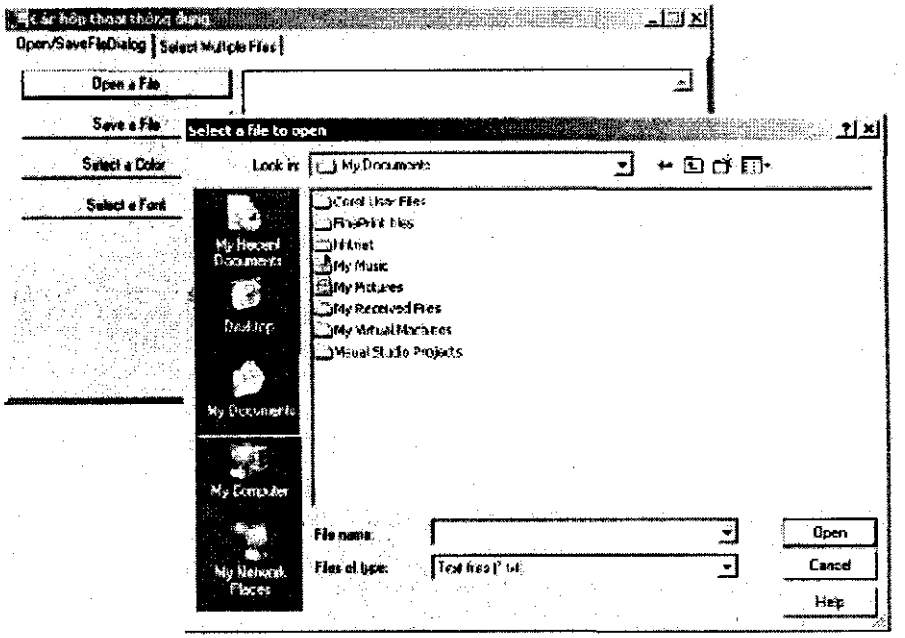
```
            Case "GetAttention"
                ' Một vài hoạt hình cần gọi "Return" để trở về trạng thái
                ' tự nhiên. Các hoạt hình khác có các return xây dựng sẵn.
                .Play("GetAttention")
                .Play("GetAttentionReturn")
            Case "Explain"
                .Play("Explain")
            Case "Congratulate"
                .Play("Congratulate")
            Case "Announce"
                .Play("Announce")
            Case "Applaud"
                .Play("Congratulate_2")
            Case "DontRecognize"
                .Play("DontRecognize")
            Case "Write"
                .Play("Write")
            Case "Surprised"
                .Play("Surprised")
            Case "Suggest"
                .Play("Suggest")
            Case "Wave"
                .Play("Wave")
        End Select
    End With
End Sub

' Thủ tục xử lý TabControl, di chuyển Merlin ra khỏi vùng làm việc
' khi người sử dụng chọn tab Word hay Excel.
Private Sub tabOfficeDemo_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles tabOfficeDemo.SelectedIndexChanged
    If tabOfficeDemo.SelectedIndex <> 0 Then
        With agentCharacter
            .StopAll()
            .MoveTo(CShort(Me.Location.X + 340), CShort(Me.Location.Y + 40))
        End With
    End If
End Sub
End Class
```

Menu.xml

```
<?xml version='1.0'?>
<lunch-menu>
 <Food>
  <Item>Cheese Pizza</Item>
  <Price>$6.95</Price>
  <Calories>800</Calories>
 </Food>
 <Food>
  <Item>Pepperoni Pizza</Item>
  <Price>$7.95</Price>
  <Calories>950</Calories>
 </Food>
 <Food>
  <Item>The "Everything" Pizza</Item>
  <Price>$9.95</Price>
  <Calories>1050</Calories>
 </Food>
 <Food>
  <Item>Hungarian Ghoulash</Item>
  <Price>$4.50</Price>
  <Calories>600</Calories>
 </Food>
 <Food>
  <Item>Maisey's Pork Sandwich</Item>
  <Price>$6.95</Price>
  <Calories>950</Calories>
 </Food>
</lunch-menu>
```

6. Các hộp thoại thông dụng

Chương trình này cho biết cách sử dụng các control **OpenFileDialog**, **SaveFileDialog**, **ColorDialog** và **FontDialog**.



Các control chính:

Loại	Thuộc tính	Giá trị
SaveFileDialog	(Name)	sdlgTextFile
OpenFileDialog	(Name)	odlgTextFile
OpenFileDialog	(Name)	odlgFileNames
FontDialog	(Name)	fdlgText
ColorDialog	(Name)	cdlgText

Các chức năng chính:

Trong chương trình chứa một tab control có hai trang. Trang đầu tiên cho phép chọn một file, mở file, lưu file, và chọn font và màu sắc cho văn bản. Tab thứ hai biểu diễn cách sử dụng các thuột tính **FileNames** và **MultiSelect** của control **OpenFileDialog**.

Project: **Cac Hop Thoai Thong Dung**

frmMain.vb

```
Option Strict On
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

    Private FileName As String

    Private Sub btnRetriveFileNames_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnRetriveFileNames.Click
        Try
            With odlgFileNames
                ' Không nhất thiết phải làm điều này, nhưng ta có thể quy
                ' định thư mục bắt đầu cho hộp thoại. Để thuộc tính
                ' DefaultFolder có tác dụng, cần phải gọi phương thức Reset.
                .Reset()
                .InitialDirectory = "C:\"

                ' Thêm phần mở rộng mặc nhiên vào tên file nếu
                ' người sử dụng không gõ nó vào. Mặc nhiên là True.
                .AddExtension = True

                ' Kiểm tra để bảo đảm rằng file được chọn tồn tại. Nếu không tồn tại
                ' hộp thoại sẽ hiển thị một thông báo. Mặc nhiên là True.
                .CheckFileExists = True

                ' Kiểm tra để bảo đảm đường dẫn đến file là đúng.
                ' Giá trị mặc nhiên là True
                .CheckPathExists = True

                ' Quy định phần mở rộng mặc nhiên. Không kèm dấu chấm
                ' Giá trị mặc nhiên là ""
                .DefaultExt = "txt"
```

```
    ' Trả về file được chỉ đến bởi một file liên kết hay không? Nếu False, chỉ
      ' đơn giản là trả về file liên kết được chọn. Nếu True, trả về
      ' file được liên kết trong LNK file. Giá trị mặc nhiên là True.
    .DereferenceLinks = True

' Bộ lọc file, sử dụng một tập các cặp được tách nhau bởi "|".
' Mỗi cặp chứa một phần mô tả đứng đầu. Sử dụng "|" ở giữa
' các cặp. Ta cũng có thể quy định thuộc tính FilterIndex để
' chọn bộ lọc mặc nhiên. Bắt đầu từ 1, và giá trị mặc nhiên là 1.
    .Filter = _
    "Text files (*.txt)|*.txt|" & _
    "All files|*.*"

' Nếu muốn cho phép người sử dụng chọn nhiều file, quy định
' thuộc tính này là True. Lấy tên các file được chọn bằng
' thuộc tính FileNames của hộp thoại. Giá trị mặc nhiên là False.
    .Multiselect = True

      ' Trở về thư mục cũ khi thực hiện chọn một file hay không? Nếu False,
      ' thư mục hiện hành đổi thành thư mục đã chọn file. Đặt thuộc
      ' tính này là True để đưa thư mục hiện hành trở lại ban đầu.
      ' Giá trị mặc nhiên là False
    .RestoreDirectory = True

      ' Hiển thị button Help và checkbox Read-Only?
    .ShowHelp = True
    .ShowReadOnly = False

      ' Khởi tạo giá trị cho checkbox Read-Only?
    .ReadOnlyChecked = False
      ' Chuỗi hiển thị trên thanh tiêu đề. Mặc nhiên là ""
    .Title = "Select a file"

      ' Chỉ chấp nhận các tên file đúng trong Win32?
      ' Mặc nhiên là True
    .ValidateNames = True
```

```
        If .ShowDialog() = DialogResult.OK Then
            ' Ta có một chọn lựa ở đây. Có thể sử dụng thuộc tính
            ' FileName hoặc FileNames để lấy tên file được chọn, hoặc có thể sử
            ' dụng phương thức OpenFile để mở file như là một Stream chỉ đọc.
          lstFiles.DataSource = .FileNames

          ' Hoặc có thể lập trình như sau, duyệt qua tên các file được chọn:
            'Dim strName As String
            'For Each strName In .FileNames
            '    lstFiles.Items.Add(strName)
            'Next
        End If
      End With
    Catch exp As Exception
      MessageBox.Show(exp.Message, Me.Text)
    End Try
  End Sub

  Private Sub btnOpenTextFile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnOpenTextFile.Click
    Dim ts As StreamReader
    Try
      With odlgTextFile
       ' Xem thủ tục btnRetriveFileNames_Click để biết các thuộc tính
        .CheckFileExists = True
        .CheckPathExists = True
        .DefaultExt = "txt"
        .DereferenceLinks = True

        .Filter = _
        "Text files (*.txt)|*.txt|" & _
        "All files|*.*"

        .Multiselect = False
        .RestoreDirectory = True
        .ShowHelp = True
        .ShowReadOnly = False
        .Title = "Select a file to open"
        .ValidateNames = True
```

```
        If .ShowDialog() = DialogResult.OK Then
          FileName = .FileName
          ts = New StreamReader(.OpenFile)
          txtFileContents.Text = ts.ReadToEnd()
        End If
      End With
    Catch exp As Exception
      MessageBox.Show(exp.Message, Me.Text)
    Finally
      If Not (ts Is Nothing) Then
        ts.Close()
      End If
    End Try
  End Sub

  Private Sub btnSaveTextFile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSaveTextFile.Click
    Dim sw As StreamWriter

    Try
      With sdlgTextFile
      ' Xem code sử dụng control OpenFileDialog để biết các thuộc tính
        .AddExtension = True

          ' Kiểm tra đường dẫn file có tồn tại hay không?.
          ' Thông báo trước khi tạo file mới?
          ' Thông báo trước khi ghi chồng lên file khác?

        .CheckPathExists = True
        .CreatePrompt = False
        .OverwritePrompt = True
        .ValidateNames = True
        .ShowHelp = True

      ' Nếu người sử dụng không cung cấp phần tên file mở rộng thì dùng giá trị này.
        .DefaultExt = "txt"

          ' Đặt tên file mặc nhiên là tên file hiện hành
        .FileName = FileName
```

```
            .Filter = _
            "Text files (*.txt)|*.txt|" & _
            "All files|*.*"
            .FilterIndex = 1
            If .ShowDialog() = DialogResult.OK Then
               FileName = .FileName
               sw = New StreamWriter(FileName)
               sw.Write(txtFileContents.Text)
            End If
         End With
      Catch exp As Exception
         MessageBox.Show(exp.Message, Me.Text)
      Finally
         If Not (sw Is Nothing) Then
            sw.Close()
         End If
      End Try
   End Sub

   Private Sub odlgTextFile_FileOk(ByVal sender As System.Object, ByVal e As
   System.ComponentModel.CancelEventArgs) Handles odlgTextFile.FileOk
      ' Sự kiện này xảy ra khi chọn một file nhưng trước khi đóng
      ' hộp thoại. Nếu gán giá trị của thuộc tính
      ' CancelEventArgs.Cancel là True thì hộp thoại sẽ đóng lại.
      If MessageBox.Show( _
      "Do you want to open a new file?", _
      Me.Text, MessageBoxButtons.YesNo, _
      MessageBoxIcon.Question) <> DialogResult.Yes Then

         e.Cancel = True
      End If
   End Sub

   Private Sub odlgTextFile_HelpRequest(ByVal sender As System.Object, ByVal e
   As System.EventArgs) Handles odlgTextFile.HelpRequest
      ' Thủ tục này thực hiện khi nhấn vào nút Help
      MessageBox.Show("Display your own help for the OpenFileDialog control here.",
   Me.Text)
   End Sub
```

```
Private Sub sdlgTextFile_HelpRequest(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles sdlgTextFile.HelpRequest
    ' Thủ tục này thực hiện khi nhấn vào nút Help
   MessageBox.Show("Display your own help for the SaveFileDialog control here.",
Me.Text)
End Sub

Private Sub btnSelectColor_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSelectColor.Click

    ' Khởi tạo CustomColors bằng một mảng số nguyên.
    ' Chú ý những màu này không có cùng định dạng với các màu
    ' trong .NET – Chúng là các giá trị màu theo kiểu VB6. Có thể
    ' dùng hàm RGB để tạo các giá trị này.

    Static CustomColors() As Integer = _
    {RGB(255, 0, 0), RGB(0, 255, 0), RGB(0, 0, 255)}

    Try
      With cdlgText
          ' Khởi tạo màu được chọn bằng với màu đang được sử dụng
          ' trong TextBox. Giá trị mặc nhiên là black.
        .Color = txtFileContents.ForeColor

          ' Nhập các màu custom vào hộp thoại với mảng đã tạo.
        .CustomColors = CustomColors

          ' Có cho phép người sử dụng tạo các màu tùy ý?
        .AllowFullOpen = True

          ' Hiển thị tất cả các màu cơ bản?
        .AnyColor = True

          ' Có mở hộp thoại với phần chọn màu tùy ý?
        .FullOpen = False

          ' Giới hạn chỉ sử dụng các màu đặc (solid).
        .SolidColorOnly = True
        .ShowHelp = True
```

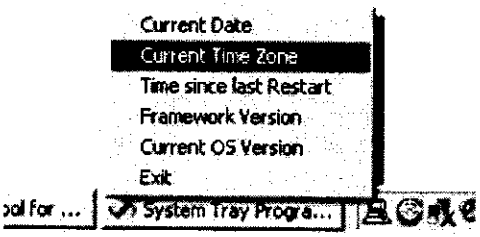
```
            If .ShowDialog() = DialogResult.OK Then
                txtFileContents.ForeColor = .Color

                ' Lưu lại các màu custom để sử dụng lại
                CustomColors = .CustomColors
            End If

            ' Khởi tạo lại tất cả các màu trong hộp thoại. Thủ tục này
            ' không cần thiết, nhưng gọi nó để bảo đảm là hộp thoại
            ' trong một trạng thái xác định để sử dụng trong lần kế tiếp.
            cdlgText.Reset()
        End With

        ' Có thể lập trình với code ngắn hơn
        'With cdlgText
        '  .Color = txtFileContents.ForeColor
        '  If .ShowDialog() = DialogResult.OK Then
        '    txtFileContents.ForeColor = .Color
        '  End If
        'End With
    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
    End Try
End Sub

Private Sub btnSelectFont_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSelectFont.Click
    Try
        With fdlgText
            ' Khởi tạo hộp thoại font cho thích hợp với font đang sử dụng trong TextBox.
            .Font = txtFileContents.Font
            .Color = txtFileContents.ForeColor

            ' Cho phép chọn màu. Giá trị mặc nhiên là False
            .ShowColor = True

            ' Hiển thị nút Apply trên hộp thoại
            .ShowApply = True
```

```
        ' Cho phép chọn hiệu ứng
      .ShowEffects = True

      .ShowHelp = True

        ' Cho phép thay đổi tập ký tự, nhưng không cho phép
        ' các font giả lập, font vector và font đứng.
      .AllowScriptChange = True
      .AllowSimulations = False
      .AllowVectorFonts = False
      .AllowVerticalFonts = False

      ' Cho phép các font cố định (fixed), và chỉ cho phép các font
      ' đã tồn tại. Quy định các kích thước font nhỏ nhất và lớn nhất.

      .FixedPitchOnly = False
      .FontMustExist = True

        ' Giá trị mặc nhiên cho cả hai thuộc tính này là 0
      .MaxSize = 48
      .MinSize = 8

        ' Hiển thị hộp thoại và áp dụng các yêu cầu đã chọn
      If .ShowDialog = DialogResult.OK Then
        ApplyFontAndColor()
      End If
    End With

    ' Có thể lập trình ngắn gọn như sau:
    'With fdlgText
    '  .Font = txtFileContents.Font
    '  If .ShowDialog() = DialogResult.OK Then
    '    txtFileContents.Font = .Font
    '  End If
    'End With
  Catch exp As Exception
    MessageBox.Show(exp.Message, Me.Text)
  End Try
End Sub
```

```
    Private Sub ApplyFontAndColor()
        Try
            With fdlgText
                ' Áp dụng font của TextBox, bao gồm font, kích thước...
                txtFileContents.Font = .Font
                ' Nếu có chọn màu sắc, thì áp dụng màu
                If .ShowColor Then
                    txtFileContents.ForeColor = .Color
                End If
            End With
        Catch exp As Exception
            MessageBox.Show(exp.Message, Me.Text)
        End Try
    End Sub

    Private Sub fdlgText_Apply(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles fdlgText.Apply
        ' Thủ tục xử lý nút Apply của hộp thoại Font
        ApplyFontAndColor()
    End Sub
End Class
```

7. Tạo một chương trình trên System Tray

Mục đích của chương trình này là chỉ cách viết một chương trình mà nó hiển thị một biểu tượng trong khay hệ thống (kế bên đồng hồ hệ thống).



Các control chính:

Loại	Thuộc tính	Giá trị
NotifyIcon	(Name)	ntfSystemInfo
	ContextMenu	mnuCtx
ContextMenu	(Name)	mnuCtx

Các chức năng chính:

Ngoài việc biểu diễn cách tạo một NotifyIcon và kết hợp nó với một ContextMenu. Chương trình này còn chỉ cách truy xuất hệ thống và các thông tin khác từ các class trong framework. Bao gồm:

- Thời gian sử dụng máy tính kể từ lúc khởi động máy.
- Time zone hiện hành của máy tính
- Ngày hiện hành
- Phiên bản của framework được cài đặt trên máy tính
- Phiên bản của hệ điều hành

Các thông tin này được hiển thị khi người sử dụng chọn các mục trong menu kết hợp với biểu tượng trên khay hệ thống.

Project: **System Tray Program**

frmMain.vb

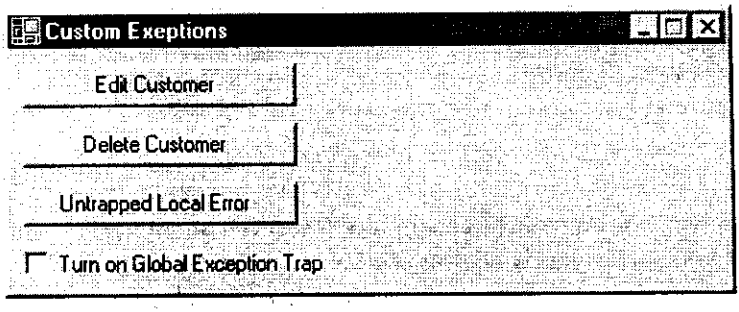
```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Code phát sinh form được bỏ qua
    Private Sub btnTray_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnTray.Click
        ' Ẩn form chính
        Me.Hide()
        ntfSystemInfo.Visible = True
        ntfSystemInfo.Text = "System Information"
    End Sub

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        ' Bảo đảm là biểu tượng không hiển thị trên system tray khi
        ' chưa nhấn vào button Tray
        ntfSystemInfo.Visible = False
    End Sub

    Private Sub mnuCurrentOSVersion_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles mnuCurrentOSVersion.Click
        ' Lấy thông tin hệ điều hành.
        MessageBox.Show("OS Version: " + _
```

```
        Environment.OSVersion.ToString(), _
        "Operating System", MessageBoxButtons.OK, _
        MessageBoxIcon.Information)
    End Sub

    Private Sub mnuCurrentTime_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuCurrentDate.Click
        ' Hiển thị ngày tháng hiện hành
        MessageBox.Show("Today's Date is: " + _
          DateTime.Now.ToLongDateString(), _
          "Date", MessageBoxButtons.OK, _
          MessageBoxIcon.Information)
    End Sub

    Private Sub mnuCurrentTimeZone_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles mnuCurrentTimeZone.Click

        ' Sử dụng class TimeZone để hiển thị tên timezone hiện hành
        If (TimeZone.CurrentTimeZone.IsDaylightSavingTime(DateTime.Now)) Then
          MessageBox.Show( _
                      "The current time zone is: " + _
                      TimeZone.CurrentTimeZone.DaylightName, _
                      "Time Zone", _
                      MessageBoxButtons.OK, _
                      MessageBoxIcon.Information)
        Else
          MessageBox.Show( _
                "The current time zone is: " + _
                TimeZone.CurrentTimeZone.StandardName, _
                "Time Zone", _
                MessageBoxButtons.OK, _
                MessageBoxIcon.Information)
        End If
    End Sub

    Private Sub mnuExitApp_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuExitApp.Click
        Shutdown()
    End Sub
```

```
Private Sub mnuFrameworkVersion_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuFrameworkVersion.Click
    ' Lấy thông tin về phiên bản của .NET Framework.
    MessageBox.Show("Framework Version: " + _
      Environment.Version.ToString(), ".NET Framework Version", _
      MessageBoxButtons.OK, _
      MessageBoxIcon.Information)
End Sub

Private Sub mnuTimeSinceReboot_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuTimeSinceReboot.Click
    ' Hiển thị thời gian sử dụng máy tính kể từ lúc khởi động máy.
    ' Thời gian lấy được từ TickCount được tính theo Milisecond.
    ' Giá trị này được chuyển sang phút.

    Dim timeSinceLastRebootMinutes As Double = _
                    ((Environment.TickCount / 1000) / 60)
    MessageBox.Show("It has been : " + _
      CInt(timeSinceLastRebootMinutes).ToString() + _
      " minutes since your last reboot.", "Last Reboot", _
      MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub

Private Sub ntfSystemInfo_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ntfSystemInfo.DoubleClick
    ' Khi người sử dụng nhấn đúp chuột vào biểu tượng trên system
    ' tray, hiển thị form chính, và ẩn biểu tượng khi hiển thị form.
    ntfSystemInfo.Visible = False
    Me.Show()
End Sub

Protected Sub Shutdown()
    ' Biểu tượng phải mất đi trước khi ứng dụng kết thúc, nếu không nó
    ' vẫn nằm đó cho dù ứng dụng đã kết thúc.
    ntfSystemInfo.Visible = False
    Application.Exit()
End Sub

End Class
```

8. Các Exception tùy biến

Chương trình này biểu diễn hai kỹ thuật cao cấp quản lý các ngoại lệ (exception).



Các chức năng chính:

Chương trình này chứa một Windows Forms client và một class library.

Class Library

Class Library định nghĩa một lớp ứng dụng, **Customer**. Ngoài ra, nó còn thể hiện một tập các class exception tùy biến (tự tạo), chúng được sử dụng để thể hiện các lỗi được định nghĩa trong ứng dụng. Trong ứng dụng này, có một cây phân cấp các exception. Cấp cao nhất là CRMSystemException dành cho một hệ thống **SomeCompany CRM** giả tưởng. Nó thừa kế từ exception System.Application. Từ đây, hai class khác thừa kế từ nó:

- CustomerException
- EmployeeException

Không có chổ nào trong ví dụ này có liên quan đến employee, nhưng ở đây chỉ cách làm việc với một sản phẩm lớn hơn có thể hình thành từ nhiều module hoặc component. Có hai exception khác thừa kế từ CustomerException:

- CustomerNotFoundException
- CustomerNotDeleteException

Mỗi class cung cấp thông tin ở các cấp khác nhau và có chức năng khác nhau. Xem thêm trong source code để được rõ hơn. Chú ý là có thể đặt tên cho các class exception tùy ý, nhưng quy ước là kết thúc với phần "Exception".

Client

Ứng dụng Client có một tham chiếu (reference) đến class library. Hai nút đầu cho thấy cách thực thi các phương thức với class customer và bắt (catch) các custom exception. Nút thứ ba gây một ngoại lệ không được bắt. Ngoài ra, có một checkbox **Turn on Global Exception Trap**, đánh dấu vào sẽ cho phép

bắt exception toàn cục (chung). Khi được cho phép, code định nghĩa trong **OnThreadException** sẽ được thực hiện thay vì thủ tục mặc nhiên của Windows Form.

Project: **Custom Exceptions**

Chạy ứng dụng

Chương trình tốt nhất là chạy ngoài trình bắt lỗi để thấy được kết quả của thục tục bắt lỗi chung. Chạy chương trình và nhấn vào ba nút. Chú ý cách thông báo lỗi "lịch sự" của nút **Edit Customer** và **Delete Customer**. Mặc dù mặc nhiên khi nhấn vào nút thứ ba **"Untrapped Local Error"**, một hộp thoại chọn lựa xuất hiện. Hộp thoại này là thủ tục bắt lỗi chung được định nghĩa bởi **Windows Form**. Đánh dấu vào checkbox **Turn on Global Exception Trap** để thấy bẩy lỗi tùy biến hoạt động.

Server

Thuộc tính của Project:

Output Type: Class Library

Root NameSpace: SomeCompany.CRMSystem

Customer.vb

```
Option Strict On
Public Class Customer
    ' Trong ứng dụng thực tế, nên dùng các thủ tục thuộc tính
    Public Id As Integer
    Public FirstName As String
    Public LastName As String

    Public Shared Function EditCustomer(ByVal Id As Integer) As Customer
        ' Ta giả sử tìm thông tin Customer từ database, nhưng không tìm
        ' ra. Do đó, thông báo cho thục tục gọi class này một cách để không bỏ qua.
        Dim strMsg As String
        strMsg = String.Format("The customer you requested by Id {0} could not be
found.", Id)
        ' Tạo exception
        Dim exp As New CustomerNotFoundException(strMsg)
        ' Đẩy nó ra cho thủ tục gọi
        Throw exp
    End Function
```

```
    Public Shared Sub DeleteCustomer(ByVal Id As Integer)
        ' Giả sử ta tìm thấy Customer trong database nhưng không thể xóa
        ' nó được vì lý do quyền hạn.

        Dim c As New Customer()
        With c
            .Id = Id
            .FirstName = "Joe"
            .LastName = "Smith"
        End With

        ' Đoạn code này có thể có lỗi nếu không có các quyền hạn
        Dim strUser As String
        Try
            strUser = System.Environment.UserDomainName & "\" & _
                                System.Environment.UserName
        Catch pexp As Exception
            strUser = "Unavailable"
        End Try

        Dim strMsg As String
        strMsg = String.Format("The customer you requested {0} {1} could not be
    deleted. Your account '{2}' does not have permission.", c.FirstName, c.LastName,
    strUser)

        Dim exp As New CustomerNotDeletedException(strMsg, c, strUser)
        exp.LogError()
        Throw exp
    End Sub
End Class

' Exception này cung cấp nền tảng cho hệ thống CRM.
Public Class CRMSystemException
    Inherits System.ApplicationException
    Private m_AppSource As String
    Public Sub New(ByVal Message As String)
        MyBase.New(Message)
        Me.m_AppSource = "SomeCompany CRM System"
    End Sub
```

```
    ' Ta chỉ muốn phương thức này thấy được trong cùng assembly. Tuy
    ' nhiên, có thể thay đổi scope nếu class này có sử dụng trong một assembly khác.
    Friend Sub LogError()
        Dim e As System.Diagnostics.EventLog
        e = New System.Diagnostics.EventLog("Application")
        e.Source = Me.AppSource
        e.WriteEntry(Me.Message, System.Diagnostics.EventLogEntryType.Error)
        e.Dispose()
    End Sub

    Public Overridable ReadOnly Property AppSource() As String
        Get
            Return m_AppSource
        End Get
    End Property
End Class

' Exception nền tảng của module Customer. Tất cả các exception
' Customer sẽ đưa ra một tham chiếu đến Customer thông qua thuộc
' tính chỉ đọc CustomerInfo. Trong một số trường hợp nó có giá trị Nothing.
Public Class CustomerException
    Inherits CRMSystemException

    Private m_AppSource As String

    Private m_Customer As Customer

    Public Sub New(ByVal Message As String, ByVal ReqCustomer As Customer)
        MyBase.New(Message)
        Me.m_Customer = ReqCustomer
        Me.m_AppSource = "SomeCompany CRM Customer Module"
    End Sub

    Public ReadOnly Property CustomerInfo() As Customer
        Get
            Return MyClass.m_Customer
        End Get
    End Property
```

```
    Public Overrides ReadOnly Property AppSource() As String
        Get
            Return Me.m_AppSource
        End Get
    End Property
End Class

' Exception đơn giản
Public Class CustomerNotFoundException
    Inherits CustomerException
    Public Sub New(ByVal Message As String)
        ' Truyền Nothing vào Customer vì không tìm thấy
        MyBase.New(Message, Nothing)
    End Sub
End Class

' Exception này cung cấp một thuộc tính id được khởi tạo lúc xây dựng
Public Class CustomerNotDeletedException
    Inherits CustomerException
    Private m_UserId As String
    Public Sub New(ByVal Message As String, ByVal ReqCustomer As Customer,
    ByVal UserId As String)
        MyBase.New(Message, ReqCustomer)
        Me.m_UserId = UserId
    End Sub

    Public ReadOnly Property UserId() As String
        Get
            Return MyClass.m_UserId
        End Get
    End Property
End Class

' Các exception tùy biến không cần phải phức tạp.
Public Class EmployeeException
    Inherits CRMSystemException
    Public Sub New(ByVal Message As String)
        MyBase.New(Message)
    End Sub
End Class
```

Client

frmMain.vb

```
Option Strict On

' Import assembly bên ngoài, assembly này định nghĩa class Customer,
' và ba exception tùy biến. Chú ý là yêu cầu có tham chiếu trong phần
' References của project.
Imports SomeCompany.CRMSystem

Public Class frmMain
  Inherits System.Windows.Forms.Form

'Code phát sinh form được bỏ qua

  Private Sub chkGET_CheckedChanged(ByVal sender As System.Object, ByVal e
  As System.EventArgs) Handles chkGET.CheckedChanged
    Dim ctl As CheckBox = CType(sender, CheckBox)

    If ctl.CheckState = CheckState.Checked Then
      ' Bật tính năng bắt exception chung. Sẽ không thấy trình bắt lỗi mặc nhiên.
      AddHandler Application.ThreadException, AddressOf Me.OnThreadException
    Else
      ' Tắt tính năng bắt exception chung và trả lại mặc nhiên cho Windows Form.
      RemoveHandler Application.ThreadException, AddressOf Me.OnThreadException
    End If
  End Sub

  Private Sub cmdEdit_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdEdit.Click
    Dim c As Customer
    Try
      Dim i As Integer = 14213
      c = Customer.EditCustomer(i)

      ' Thực hiện công việc nếu nhận được Customer

    Catch exp As CustomerNotFoundException
      MessageBox.Show(exp.Message, exp.AppSource, _
                      MessageBoxButtons.OK, MessageBoxIcon.Error)
```

```
    Catch exp As CustomerException
        MessageBox.Show(exp.Message, exp.AppSource, _
                        MessageBoxButtons.OK, MessageBoxIcon.Error)

    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, _
                        MessageBoxButtons.OK, MessageBoxIcon.Error)

    End Try
End Sub

Private Sub cmdDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdDelete.Click

    Try
        Dim i As Integer = 14213
        Customer.DeleteCustomer(i)

        ' Ta sẽ không bao giờ thấy thông báo này, nhưng để cho đầy đủ
        MessageBox.Show(String.Format("Customer Id {0} was deleted.", i), _
                Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Information)

    Catch exp As CustomerNotDeletedException
        Dim c As Customer
        c = exp.CustomerInfo

        ' Có thể làm gì đó với Customer nếu muốn.

        MessageBox.Show(exp.Message, exp.AppSource, _
                MessageBoxButtons.OK, MessageBoxIcon.Error)

    Catch exp As CustomerException
        MessageBox.Show(exp.Message, exp.AppSource, _
                    MessageBoxButtons.OK, MessageBoxIcon.Error)
    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, _
                        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

```
Private Sub cmdUntrapped_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUntrapped.Click
    ' Thông thường một lỗi không được bắt trong VB6 và các phiên
    ' bản cũ sẽ cho một thông báo ngắn trong MessageBox và sau đó
    ' thoát chương trình
    ' Tuy nhiên Windows Form có một thủ tục bắt lỗi cấp cao nhất giữa
    ' chương trình và CLR. Ta có thấy được hộp thoại của chúng hay
    ' không tùy vào ba thứ:
    ' 1) Có trình bắt lỗi (debugger) nào đang chạy hay không?
    ' 2) Chương trình có các thủ tục bắt exception riêng không?
    ' 3) Có bật tính năng bắt lỗi JIT lên trong file cấu hình?

    ' Nếu một lỗi, không được bắt, xảy ra và trả lời không cho ba câu hỏi
    ' trên, thì sẽ thấy một hộp thoại Windows Forms cung cấp cho
    ' người sử dụng các khả năng tiếp tục (Continue) hoặc thoát (Quit).
    ' Để thấy được thủ tục bắt lỗi kiểu này, cần chạy chương trình đã
    ' được biên dịch

    Dim i As Short = 1234
    Dim j As Short = 0
    Dim k As Short = -1

    k = CShort(i / j)

    ' Chú ý là không bao giờ thấy được thông báo bên dưới
    MessageBox.Show("Your reults are: " & k.ToString(), Me.Text, _
                    MessageBoxButtons.OK, MessageBoxIcon.Information)
End Sub

Friend Sub OnThreadException(ByVal sender As Object, ByVal t As
System.Threading.ThreadExceptionEventArgs)
' Tại đây có thể miêu tả những gì xảy ra. Có thể ghi lại các lỗi, hiển
' thị thông báo, gửi email...
    ' Chú ý: Nếu có một lỗi không được bắt ở đây, ta sẽ thấy
    ' hộp thoại bắt lỗi JIT, không phải là hộp thoại bắt lỗi của
    ' Windows Forms.

    ' Xóa ghi chú dòng bên dưới để thấy được
    Throw New ArgumentException("Oopss! I did it again!")
```

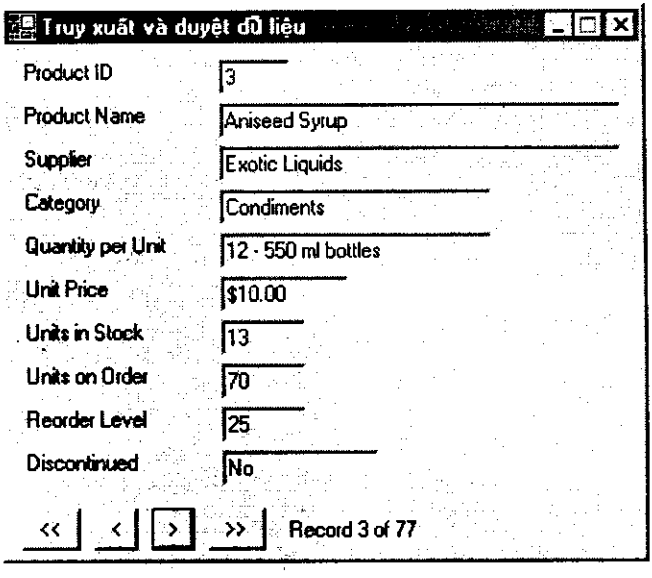
```
    Dim exp As Exception = t.Exception

    If TypeOf exp Is CustomerException Then
        ' Kiểm tra có phải là một trong các kiểu CustomerException

    ElseIf TypeOf exp Is ApplicationException Then
        ' Kiểm tra có phải là các kiểu exception khác của ứng dụng

    ElseIf TypeOf exp Is ArithmeticException Then
        ' Sẽ thấy được hộp thông báo này nếu đã chạy code
        ' trong cmdUntrapped_Click
        ' Xóa ghi chú bên dưới để kiểm tra lại
        ' MessageBox.Show(exp.Message, exp.Source, _
        '                 MessageBoxButtons.OK, MessageBoxIcon.Error)

    ElseIf TypeOf exp Is SystemException Then
        ' Kiểm tra có phải exception hệ thống
    Else

    End If

    Dim strMsg As String
    strMsg = String.Format("We're sorry, an untrapped error occurred.{0}The error
message was:{0}{1}", vbCrLf, exp.Message)

    MessageBox.Show(strMsg, "Global Exception Trap", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
  End Sub
End Class
```

app.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <!-- Đổi thành true để thấy hộp thoại bắt lỗi JIT-->
  <system.windows.forms jitDebugging="false" />
</configuration>
```

9. Truy xuất và duyệt dữ liệu

Chương trình này chỉ cách tạo một form điển hình có kết dữ liệu. Thông tin được hiển thị trên các TextBox. Khi người sử dụng duyệt qua các mẫu tin (record), nội dung bên trong các TextBox sẽ thay đổi.



Các chức năng chính:

- Ví dụ này chỉ cách kết dữ liệu từ các field cụ thể vào các control. Chương trình này cũng chỉ cách duyệt qua các record như: về đầu, lùi về một vị trí, tiến tới một vị trí, và về cuối cùng. Ngoài ra vị trí của record hiện hành và tổng số record được thể hiện trên một label.
- Đối với một số field (các giá trị Boolean, và Currency), sử dụng định dạng tùy biến bằng cách sử dụng sự kiện Format của class Binding.
- Người sử dụng có thể duyệt qua các record bằng cách sử dụng các phím điều khiển cursor. Mũi tên phải: về record trước, mũi tên trái: tiến tới một record, Home: về record đầu, End: đến record cuối. Page down nhảy qua 10 record, Page Up nhảy ngược về 10 record.
- Thông tin được lấy vào một Typed DataSet. Đối tượng này đáp ứng nhiều điểm mạnh hơn DataSet thông thường, bao gồm việc viết code đơn giản, và ít có cơ hội xảy ra lỗi.

Yêu cầu:

Chương trình sử dụng database Northwind trên SQL Server hoặc MSDE.

Project: **Truy xuat va Duyet du lieu**

ProductDataSet.xsd

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="ProductDataSet"
  targetNamespace="http://tempuri.org/ProductDataSet.xsd"
  elementFormDefault="qualified" attributeFormDefault="qualified"
  xmlns="http://tempuri.org/ProductDataSet.xsd"
  xmlns:mstns="http://tempuri.org/ProductDataSet.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
  microsoft-com:xml-msdata">
  <xs:element name="ProductDataSet" msdata:IsDataSet="true">
   <xs:complexType>
     <xs:choice maxOccurs="unbounded">
       <xs:element name="Products">
         <xs:complexType>
           <xs:sequence>
             <xs:element name="ProductID" msdata:ReadOnly="true"
  msdata:AutoIncrement="true" type="xs:int" />
             <xs:element name="ProductName" type="xs:string" />
             <xs:element name="SupplierID" type="xs:int" minOccurs="0" />
             <xs:element name="CategoryID" type="xs:int" minOccurs="0" />
             <xs:element name="QuantityPerUnit" type="xs:string" minOccurs="0" />
             <xs:element name="UnitPrice" type="xs:decimal" minOccurs="0" />
             <xs:element name="UnitsInStock" type="xs:short" minOccurs="0" />
             <xs:element name="UnitsOnOrder" type="xs:short" minOccurs="0" />
             <xs:element name="ReorderLevel" type="xs:short" minOccurs="0" />
             <xs:element name="Discontinued" type="xs:boolean" />
             <xs:element name="SupplierName" type="xs:string" />
             <xs:element name="CategoryName" type="xs:string" />
           </xs:sequence>
         </xs:complexType>
       </xs:element>
     </xs:choice>
   </xs:complexType>
  </xs:element>
</xs:schema>
```

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected productInfo As New ProductDataSet()

    Protected Const SQL_CONNECTION_STRING As String = _
      "Server=localhost;" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
      "Server=(local)\NetSDK;" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI"
' Code phát sinh form được bỏ qua

    ' Khi kết dữ liệu vào một giá trị boolean, thủ tục này chuyển nó
    ' thành chuỗi True/False hoặc Yes/No
    Protected Sub BooleanToYesNo(ByVal sender As Object, ByVal e As
  ConvertEventArgs)
        ' Phương thức này chỉ chuyển sang dạng chuỗi. Kiểm tra bằng DesiredType
      If Not e.DesiredType Is GetType(String) Then
        Exit Sub
      End If
        ' Nếu giá trị là "true" thì chuyển thành "Yes", nếu khác thì chuyển thành "No"
      e.Value = IIf(CType(e.Value, Boolean), "Yes", "No")
    End Sub

    Protected Sub DecimalToCurrencyString(ByVal sender As Object, ByVal e As
  ConvertEventArgs)
      If Not e.DesiredType Is GetType(String) Then
        Exit Sub
      End If
      ' Sử dụng phương thức ToString để định dạng các giá trị là tiền tệ ("c")
      e.Value = CType(e.Value, Decimal).ToString("c")
    End Sub
```

```
' Lùi lại 10 record
Public Sub Back10()
    Me.BindingContext(productInfo.Products).Position -= 10
End Sub

' Về record đầu tiên
Public Sub FirstRecord()
    ' Vị trí record đầu tiên là 0 (không phải 1)
    Me.BindingContext(productInfo.Products).Position = 0
End Sub

' Tiến tới 10 record
Public Sub Forward10()
    Me.BindingContext(productInfo.Products).Position += 10
End Sub

' Đến record cuối cùng
Public Sub LastRecord()
    ' Sử dụng ProductInfo.Products.Count để lấy tổng số record.
    Me.BindingContext(productInfo.Products).Position = _
        productInfo.Products.Count - 1
End Sub

' Di chuyển đến record kế tiếp
Public Sub NextRecord()
    Me.BindingContext(productInfo.Products).Position += 1
End Sub

' Lùi về một record
Public Sub PreviousRecord()
    Me.BindingContext(productInfo.Products).Position -= 1
End Sub

' In ra số record
Protected Sub ShowCurrentRecord()
    lblRecordNumber.Text = "Record " & _
     Me.BindingContext(productInfo.Products).Position + 1 & " of " & _
     productInfo.Products.Count
End Sub
```

```
Private Sub btnFirst_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFirst.Click
    ' Di chuyển đến record đầu tiên
    FirstRecord()
End Sub

Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLast.Click
    ' Di chuyển đến record cuối
    LastRecord()
End Sub

Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNext.Click
    ' Di chuyển đến record kế tiếp
    NextRecord()
End Sub

Private Sub btnPrevious_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrevious.Click
    ' Lùi về một record
    PreviousRecord()
End Sub

Private Sub frmMain_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles MyBase.KeyDown

    ' Cho phép người sử dụng sử dụng các phím điều khiển cursor.
    If e.KeyCode = Keys.Right Then NextRecord()
    If e.KeyCode = Keys.Left Then PreviousRecord()
    If e.KeyCode = Keys.Home Then FirstRecord()
    If e.KeyCode = Keys.End Then LastRecord()
    If e.KeyCode = Keys.PageDown Then Forward10()
    If e.KeyCode = Keys.PageUp Then Back10()

End Sub
```

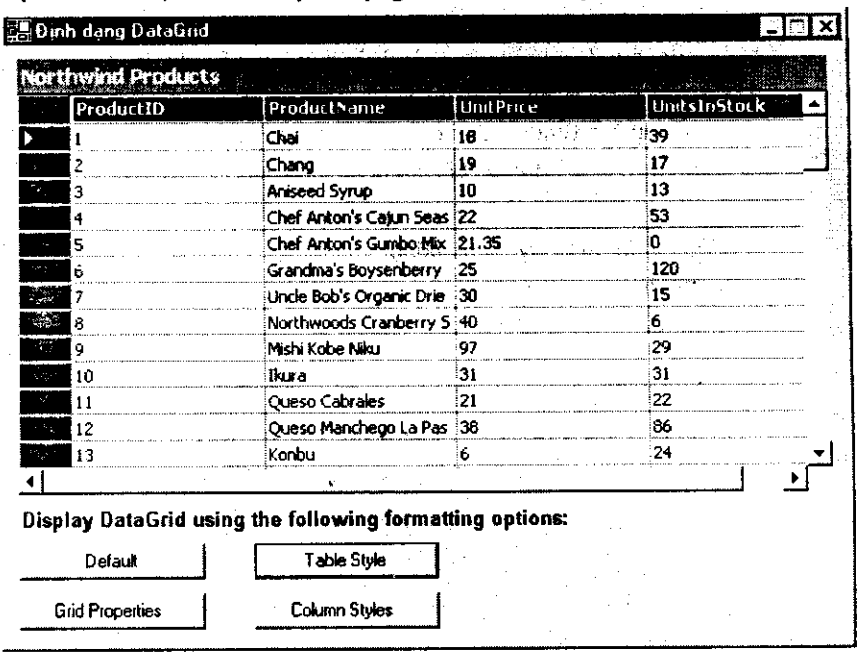
```
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Static ConnectionString As String = SQL_CONNECTION_STRING
    Static DidPreviouslyConnect As Boolean = False

    Dim frmStatusMessage As New frmStatus()
    If Not DidPreviouslyConnect Then
      frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim IsConnecting As Boolean = True
    While IsConnecting
      Try
        Dim northwindConnection As New SqlConnection(ConnectionString)
          ' Câu lệnh truy vấn này lấy tất cả product cùng với
          ' CategoryName và SupplierName cho mỗi product
        Dim selectCommand As String = _
         "SELECT Products.ProductID, Products.ProductName, " & _
             "Products.SupplierID, Products.CategoryID, " & _
             "Products.QuantityPerUnit, Products.UnitPrice, " & _
             "Products.UnitsInStock, Products.UnitsOnOrder, " & _
             "Products.ReorderLevel, Products.Discontinued, " & _
             "Suppliers.CompanyName AS SupplierName, " & _
             "Categories.CategoryName " & _
          "FROM Products INNER JOIN " & _
             "Suppliers ON Products.SupplierID =" & _
             Suppliers.SupplierID INNER JOIN " & _
             "Categories ON Products.CategoryID = " & _
             Categories.CategoryID"

          ' SqlDataAdapter thực sự phát câu lệnh đến database.
        Dim productAdapter As New _
            SqlDataAdapter(selectCommand, northwindConnection)

        productAdapter.Fill(productInfo.Products)

        txtProductID.DataBindings.Add("Text", productInfo.Products, "ProductID")
        txtProductName.DataBindings.Add("Text", productInfo.Products, _
                                        "ProductName")
```

```
        txtSupplier.DataBindings.Add("Text", productInfo.Products, _
                                     "SupplierName")
        txtCategory.DataBindings.Add("Text", productInfo.Products, _
                                     "CategoryName")
        txtQuantityPerUnit.DataBindings.Add("Text", productInfo.Products, _
                                     "QuantityPerUnit")
        txtUnitsInStock.DataBindings.Add("Text", productInfo.Products, _
                                     "UnitsInStock")
        txtUnitsOnOrder.DataBindings.Add("Text", productInfo.Products, _
                                     "UnitsOnOrder")
        txtReorderLevel.DataBindings.Add("Text", productInfo.Products, _
                                     "ReorderLevel")
        Dim UnitPriceBinding As New Binding("Text", productInfo.Products, _
                                     "UnitPrice")
        AddHandler UnitPriceBinding.Format, _
                            AddressOf DecimalToCurrencyString
        txtUnitPrice.DataBindings.Add(UnitPriceBinding)

        Dim DiscontinuedBinding As New Binding("Text", _
                        productInfo.Products, "Discontinued")
        AddHandler DiscontinuedBinding.Format, AddressOf BooleanToYesNo

        txtDiscontinued.DataBindings.Add(DiscontinuedBinding)

        AddHandler Me.BindingContext(productInfo.Products).PositionChanged, _
                        AddressOf ProductInfo_PositionChanged

        ' Dữ liệu đã được lấy thành công, do đó thoát vòng lặp
        IsConnecting = False
        DidPreviouslyConnect = True

    Catch exc As Exception
        If ConnectionString = SQL_CONNECTION_STRING Then
            ' Không thể kết nối vào SQL Server, chuyển qua MSDE
            ConnectionString = MSDE_CONNECTION_STRING
            frmStatusMessage.Show("Connecting to MSDE")
        Else
            ' Không thể kết nối vào SQL Server và MSDE
            frmStatusMessage.Close()
```

```
            MessageBox.Show("To run this sample, you must " & _
                            "have SQL or MSDE. ")
        End
    End If
  End Try
 End While

 frmStatusMessage.Close()

 ShowCurrentRecord()
End Sub

Protected Sub ProductInfo_PositionChanged(ByVal sender As Object, ByVal e As
System.EventArgs)
 ShowCurrentRecord()
End Sub
End Class
```

10. Định dạng DataGrid

Chương trình này cho thấy cách lấy dữ liệu từ database, hiển thị dữ liệu trong một DataGrid, và cách định dạng cách trình bày của DataGrid.



Các chức năng chính

ADO.NET được sử dụng để lấy thông tin từ SQL Server. Dữ liệu được đưa vào một DataSet, và DataGrid thể hiện dữ liệu này bằng cách kết vào DataSet này.

Định dạng cách trình bày của DataGrid được thực hiện trong code tại bốn thủ tục xử lý bốn nút bấm. Các nút có tên như sau:

1. **Default**: Sử dụng định dạng mặc nhiên
2. **Grid Properties:** Chỉ sử dụng các thuộc tính định dạng của DataGrid.
3. **Table Style:** Sử dụng đối tượng **DataGridTableStyle** để định dạng DataGrid.
4. **ColumnStyles:** Đối tượng **DataGridColumnStyle** được sử dụng để định dạng **DataGrid**.

Yêu cầu:

Sử dụng database Northwind trong SQL Server hoặc MSDE.

Project: **Dinh dang DataGrid**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class frmMain
    Inherits Form

    ' ProductData sẽ chứa thông tin được trả về từ SQL Server
    Protected ProductData As New DataSet()

    ' Sử dụng để tham chiếu đến bảng chứa thông tin sản phẩm trong ProductData
    Protected Const PRODUCT_TABLE_NAME As String = "Products"

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"
```

```
    ' Các giá trị mặc nhiên cần được lưu lại
    Protected DefaultGridBorderStyle As BorderStyle

    Public Sub New()
        MyBase.New()
        InitializeComponent()

        'Lấy thông tin mặc nhiên phục vụ cho mục đích reset định dạng
        DefaultGridBorderStyle = grdProducts.BorderStyle
    End Sub

' Code phát sinh form được bỏ qua

    Private Sub BindDataGrid()

        Static ConnectionString As String = SQL_CONNECTION_STRING
        Static DidPreviouslyConnect As Boolean = False

        Dim frmStatusMessage As New frmStatus()
        If Not DidPreviouslyConnect Then
            frmStatusMessage.Show("Connecting to SQL Server")
        End If

        Dim IsConnecting As Boolean = True
        While IsConnecting
            Try
                Dim northwindConnection As New SqlConnection(ConnectionString)

                ' SqlDataAdapter được sử dụng để di chuyển dữ liệu giữa
                ' SQL Server và DataSet.
                Dim ProductAdapter As New SqlDataAdapter( _
                    "SELECT ProductID, ProductName, UnitPrice," & _
                    " UnitsInStock FROM products", northwindConnection)

                ' Thiết lập DataSet với thông tin từ bảng Products. Bởi vì
                ' DataSet có thể chứa nhiều tập kết quả, tốt nhất là đặt tên
                ' cho tập kết quả khi thiết lập DataSet. Trong trường hợp này,
                ' tập kết quả được đặt tên là "Products".
                ProductAdapter.Fill(ProductData, PRODUCT_TABLE_NAME)
```

```
            ' Đã lấy dữ liệu thành công, thoát vòng lặp
            IsConnecting = False
            DidPreviouslyConnect = True
        Catch exc As Exception
            If ConnectionString = SQL_CONNECTION_STRING Then
                ' Không thể kết nối vào SQL, chuyển qua MSDE
                ConnectionString = MSDE_CONNECTION_STRING
                frmStatusMessage.Show("Connecting to MSDE")
            Else
                ' Không thể kết nối vào SQL và MSDE
                frmStatusMessage.Close()
                MessageBox.Show("To run this sample, you must " & _
                                "have SQL or MSDE.")
                End
            End If
        End Try
    End While
    frmStatusMessage.Close()

    ' Kết DataGrid vào một bảng biết trước trong DataSet. Thông tin
    ' các sản phẩm sẽ được hiển thị
    grdProducts.DataSource = ProductData.Tables(PRODUCT_TABLE_NAME)
End Sub

Private Sub FormatGridWithBothTableAndColumnStyles()
' Trực tiếp quy định các thuộc tính của DataGrid, nhưng chỉ các
' thuộc tính không có trong các thuộc tính của DataGridTableStyle.
    With grdProducts
        .BackColor = Color.GhostWhite
        .BackgroundColor = Color.Lavender
        .BorderStyle = BorderStyle.None
        .CaptionBackColor = Color.RoyalBlue
        .CaptionFont = New Font("Tahoma", 10.0!, FontStyle.Bold)
        .CaptionForeColor = Color.Bisque
        .CaptionText = "Northwind Products"
        .Font = New Font("Tahoma", 8.0!)
        .ParentRowsBackColor = Color.Lavender
        .ParentRowsForeColor = Color.MidnightBlue
    End With
```

```
    ' Thêm các khả năng định dạng khác
    Dim grdTableStyle1 As New DataGridTableStyle()
    With grdTableStyle1
      .AlternatingBackColor = Color.GhostWhite
      .BackColor = Color.GhostWhite
      .ForeColor = Color.MidnightBlue
      .GridLineColor = Color.RoyalBlue
      .HeaderBackColor = Color.MidnightBlue
      .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
      .HeaderForeColor = Color.Lavender
      .SelectionBackColor = Color.Teal
      .SelectionForeColor = Color.PaleGreen
       ' Đừng quên quy định thuộc tính MappingName. Nếu không,
       ' các thuộc tính DataGridTableStyle và các đối tượng
       ' DataGridColumnStyle kết hợp với nó sẽ không có tác dụng.
      .MappingName = PRODUCT_TABLE_NAME
      .PreferredColumnWidth = 125
      .PreferredRowHeight = 15
    End With

    ' Định dạng từng cột muốn xuất hiện trong DataGrid. Trong đa số
    ' trường hợp, class DataGridTextBoxColumn là thích hợp. Tuy
    ' nhiên, cũng có thể dùng class DataGridBoolColumn. Cả hai
    ' class này đều mở rộng class DataGridColumnStyle.
    Dim grdColStyle1 As New DataGridTextBoxColumn()
    With grdColStyle1
      .HeaderText = "ID"
      .MappingName = "ProductID"
      .Width = 50
    End With

    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
      .HeaderText = "Name"
      .MappingName = "ProductName"
    End With
```

```
    Dim grdColStyle3 As New DataGridTextBoxColumn()
    With grdColStyle3
        .HeaderText = "Price"
        .MappingName = "UnitPrice"
        .Width = 75
        .ReadOnly = True
    End With

    Dim grdColStyle4 As New DataGridTextBoxColumn()
    With grdColStyle4
        .HeaderText = "# In Stock"
        .MappingName = "UnitsInStock"
        .Width = 75
        .Alignment = HorizontalAlignment.Center
    End With

    ' Thêm các đối tượng kiểu trình bày vào tập hợp các kiểu bảng.
    grdTableStyle1.GridColumnStyles.AddRange (New DataGridColumnStyle() _
        {grdColStyle1, grdColStyle2, grdColStyle3, grdColStyle4})
    grdProducts.TableStyles.Add(grdTableStyle1)
End Sub

Private Sub FormatGridWithoutTableStyles()
    ' Thay đổi cách trình bày của DataGrid bằng cách quy định trực tiếp
    ' các thuộc tính định dạng của nó.
    With grdProducts
        .AlternatingBackColor = Color.GhostWhite
        .BackColor = Color.GhostWhite
        .BackgroundColor = Color.Lavender
        .BorderStyle = BorderStyle.None
        .CaptionBackColor = Color.RoyalBlue
        .CaptionFont = New Font("Tahoma", 10.0!, FontStyle.Bold)
        .CaptionForeColor = Color.Bisque
        .CaptionText = "Northwind Products"
        .Font = New Font("Tahoma", 8.0!)
        .ForeColor = Color.MidnightBlue
        .GridLineColor = Color.RoyalBlue
        .HeaderBackColor = Color.MidnightBlue
        .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
        .HeaderForeColor = Color.Lavender
```

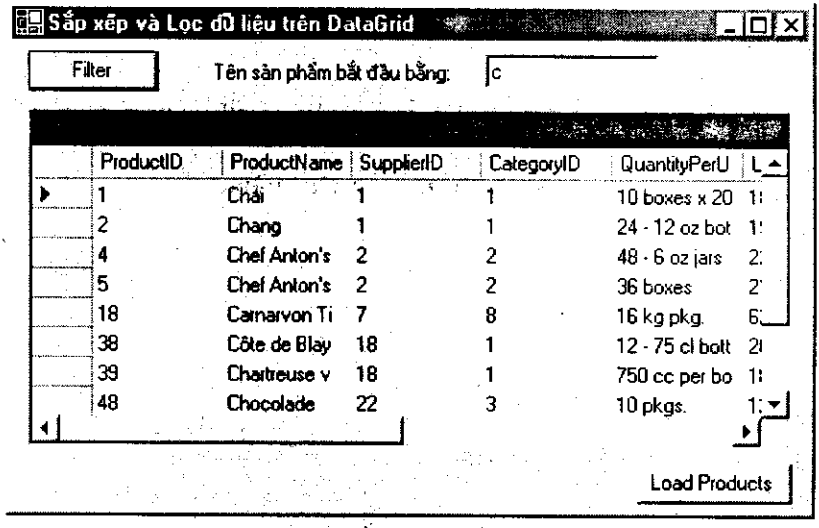
```
        .ParentRowsBackColor = Color.Lavender
        .ParentRowsForeColor = Color.MidnightBlue
        .SelectionBackColor = Color.Teal
        .SelectionForeColor = Color.PaleGreen
    End With
End Sub

Private Sub FormatGridWithTableStyles()
    ' Quy định các thuộc tính trực tiếp của DataGrid, nhưng chỉ những
    ' thuộc tính không thuộc các thuộc tính của DataGridTableStyle
    With grdProducts
        .BackColor = Color.GhostWhite
        .BackgroundColor = Color.Lavender
        .BorderStyle = BorderStyle.None
        .CaptionBackColor = Color.RoyalBlue
        .CaptionFont = New Font("Tahoma", 10.0!, FontStyle.Bold)
        .CaptionForeColor = Color.Bisque
        .CaptionText = "Northwind Products"
        .Font = New Font("Tahoma", 8.0!)
        .ParentRowsBackColor = Color.Lavender
        .ParentRowsForeColor = Color.MidnightBlue
    End With

    ' Quy định các thuộc tính định dạng khác sử dụng đối tượng
    ' DataGridTableStyle. Chú ý là một số thuộc tính định dạng của
    ' DataGrid có thể chồng chéo lên DataGridTableStyle. Do đó,
    ' thường quy định các thuộc tính của DataGrid nhiều đến mức có
    ' thể, sau đó hiệu chỉnh bằng DataGridTableStyle.

    ' Bởi vì một DataGrid có thể chứa nhiều đối tượng DataTable, một
    ' đối tượng DataGridTableStyle khác có thể áp dụng cho dữ liệu
    ' trong mỗi DataTable. Để làm điều này, đơn giản là gán thuộc tính
    ' MappingName bằng tên của DataTable chứa dữ liệu muốn
    ' định dạng.

    Dim grdTableStyle1 As New DataGridTableStyle()

    With grdTableStyle1
        .AlternatingBackColor = Color.GhostWhite
```

```
        .BackColor = Color.GhostWhite
        .ForeColor = Color.MidnightBlue
        .GridLineColor = Color.RoyalBlue
        .HeaderBackColor = Color.MidnightBlue
        .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
        .HeaderForeColor = Color.Lavender
        .SelectionBackColor = Color.Teal
        .SelectionForeColor = Color.PaleGreen

        .MappingName = PRODUCT_TABLE_NAME
        .PreferredColumnWidth = 125
        .PreferredRowHeight = 15
    End With
    ' Cho style vào tập các table style của DataGrid.
    ' Nếu không, các style sẽ không có tác dụng.
    grdProducts.TableStyles.Add(grdTableStyle1)
End Sub

Private Sub ResetDemo()
    ' Dọn dẹp DataTable để các lần nhấn nút kế tiếp không thêm dữ liệu chồng vào
    If Not IsNothing(ProductData.Tables(PRODUCT_TABLE_NAME)) Then
        ProductData.Tables(PRODUCT_TABLE_NAME).Clear()
    End If
    ' Xóa các TableStyle đã tồn tại và trả về mặc nhiên
    With grdProducts
        .BackgroundColor = SystemColors.InactiveCaptionText
        .CaptionText = ""
        .CaptionBackColor = SystemColors.ActiveCaption
        .TableStyles.Clear()
        .ResetAlternatingBackColor()
        .ResetBackColor()
        .ResetForeColor()
        .ResetGridLineColor()
        .ResetHeaderBackColor()
        .ResetHeaderFont()
        .ResetHeaderForeColor()
        .ResetSelectionBackColor()
        .ResetSelectionForeColor()
        .ResetText()
        .BorderStyle = DefaultGridBorderStyle
```

```
    End With
  End Sub

  Private Sub btnColumnStyles_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnColumnStyles.Click
    ResetDemo()
    BindDataGrid()
  ' Nếu cần định dạng ở cấp độ cột, thêm các đối tượng
  ' DataGridColumnStyle vào một một đối tượng DataGridTableStyle.
    FormatGridWithBothTableAndColumnStyles()
  End Sub

  Private Sub btnDefaultFormatting_Click(ByVal sender As System.Object, ByVal e
  As System.EventArgs) Handles btnDefaultFormatting.Click
    ResetDemo()
     ' Sử dụng định dạng mặc nhiên của Visual Studio .NET
    BindDataGrid()
  End Sub

  Private Sub btnGridProperties_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnGridProperties.Click
    ResetDemo()
    BindDataGrid()
     ' Nếu không muốn điều khiển trên từng cột cụ thể thì chỉ cần dùng
     ' các thuột tính của DataGrid.
    FormatGridWithoutTableStyles()
  End Sub


  Private Sub btnTableStyle_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnTableStyle.Click
    ResetDemo()
    BindDataGrid()
    FormatGridWithTableStyles()
  End Sub
End Class
```

11. Sắp xếp và lọc dữ liệu trên DataGrid

Ví dụ này biểu diễn cách lấy dữ liệu từ database và hiển thị dữ liệu trong một DataGrid.



Các chức năng chính:

ADO.NET được sử dụng để lấy dữ liệu từ SQL Server. Dữ liệu được đưa vào một DataSet, và dùng DataGrid kết vào DataSet để hiển thị dữ liệu. ADO.NET còn được sử dụng để lọc (filter) dữ liệu. Khi người sử dụng nhấn nút Filter, một DataView được sử dụng để loại trừ một số dòng. Ta cũng có thể sắp xếp lại dữ liệu bằng cách nhấn vào các tiêu đề (heading) của các cột trong DataGrid.

Việc kiểm tra hợp lệ được thực hiện trong thủ tục lọc dữ liệu. Nếu người sử dụng nhập vào một điều kiện không thỏa bất kỳ dòng nào, người sử dụng sẽ nhận được thông báo.

Debug.Assert được sử dụng trong thủ tục lọc dữ liệu để bảo đảm rằng DataSet có chứa dữ liệu trước khi cho kết quả lọc. Để thấy nó hoạt động, làm như sau:

1. Đổi thuộc tính Enabled của **btnFilter** thành True.
2. Nhấn **F5** để chạy ứng dụng.
3. Nhấn **Filter** trước khi nhấn **Load Data**.

Debug.Assert chỉ hoạt động trong chế độ debug, do đó người lập trình không cần lo nó hiển thị lỗi trong code đã phát hành (released code).

1. Trên thanh công cụ **Visual Studio**, đổi **Solution Configuarations** từ Debug sang Release.
2. Nhấn **F5**.
3. Nhấn **Load Products** trước khi nhấn **Filter**.

Yêu cầu:

Chương trình sử dụng database Northwind có sẵn trong SQL Server hoặc MSDE.

Project: **Sap xep va Loc du lieu tren DataGrid**

frmMain.vb

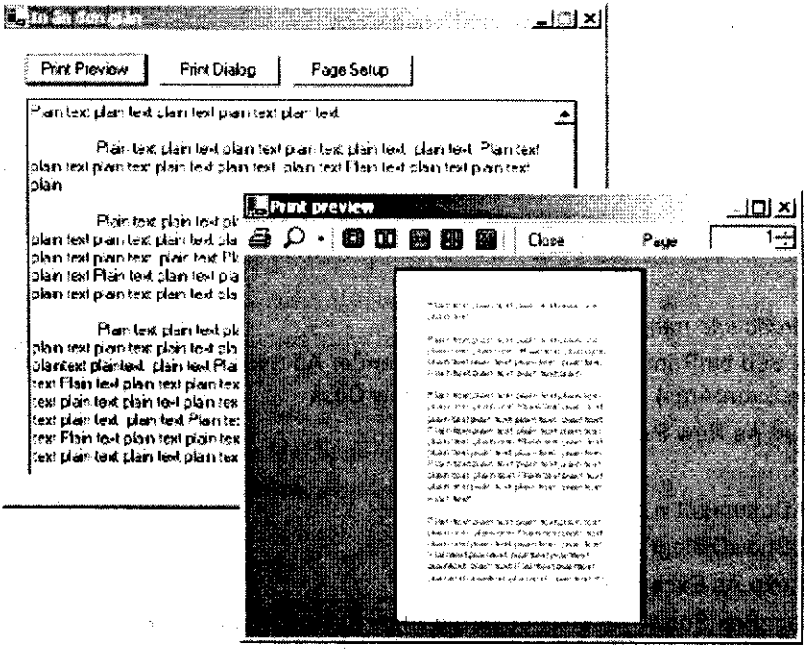
```
Option Strict On

Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' ProductData sẽ chứa thông tin trả về từ SQL Server
    Protected ProductData As New DataSet()

    ' Ví dụ này giả sử máy tính đã có cài đặt SQL Server và chứa
    ' database Northwind. Nếu không, ít nhất phải có MSDE được cài đặt
    ' trên máy tính với database Northwind. Hoặc thay đổi chuỗi kết nối
    ' bên dưới để chỉ đến một SQL Server khác bằng cách đổi "Server="

    Protected Const SQL_CONNECTION_STRING As String = _
     "Server=localhost;" & _
     "DataBase=Northwind;" & _
     "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
     "Server=(local)\NetSDK;" & _
     "DataBase=Northwind;" & _
     "Integrated Security=SSPI"

    Protected Const CONNECTION_ERROR_MSG As String = _
     "To run this sample, you must have SQL " & _
     "or MSDE with the Northwind database installed. "
```

```
Protected didPreviouslyConnect As Boolean = False
Protected connectionString As String = SQL_CONNECTION_STRING
' Sử dụng để tham chiếu bảng chứa thông tin sản phẩm
' trong ProductData
Protected Const PRODUCT_TABLE_NAME As String = "Products"

' Code phát sinh form được bỏ qua

Private Sub btnFilter_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFilter.Click
 Const MESSAGEBOX_CAPTION As String = "Filter"

 ' Kiểm tra tính đúng đắn để bảo đảm có dữ liệu trước khi tiến hành lọc
 Debug.Assert(Not ProductData.Tables(PRODUCT_TABLE_NAME) Is Nothing, _
  "No product data loaded in ProductData.Tables(PRODUCT_TABLE_NAME)")

 With ProductData.Tables(PRODUCT_TABLE_NAME)
   ' Lọc dữ liệu trong DataGrid để chỉ hiển thị các tên sản phẩm bắt đầu là
   ' một chuỗi cho trước trong TextBox.
   .DefaultView.RowFilter = "ProductName like '" & txtFilter.Text & "%'"

   ' Có sản phẩm nào thỏa hay không?
   If .DefaultView.Count = 0 Then
     MessageBox.Show("No matching rows.", _
      MESSAGEBOX_CAPTION, _
      MessageBoxButtons.OK, _
      MessageBoxIcon.Information)
   End If

   ' Bằng cách kết DataGrid vào DataView, nó sẽ hiển thị
   ' những dòng thỏa điều kiện
   grdProducts.DataSource = .DefaultView
 End With
End Sub

Private Sub btnLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLoad.Click

 Dim frmStatusMessage As New frmStatus()
```

```
    If Not didPreviouslyConnect Then
      frmStatusMessage.Show("Connecting to SQL Server")
    End If
    frmStatusMessage.Close()

    Dim isConnecting As Boolean = True
    While isConnecting
      Try
        Dim northwindConnection As New SqlConnection(connectionString)
        ' SqlDataAdapter được sử dụng để di chuyển dữ
          ' liệu giữa SQL Server và DataSet.
        Dim ProductAdapter As New SqlDataAdapter("select * from products", _
                                    northwindConnection)
        ' Xóa tất cả dữ liệu đã được lấy trước đó ra khỏi DataSet.
        ProductData.Clear()
        ' Thiết lập DataSet với thông tin từ bảng Products.
        ProductAdapter.Fill(ProductData, PRODUCT_TABLE_NAME)

' Kết DataGrid vào một bảng trong DataSet. Lần này sẽ hiển thị dữ liệu lên DataGrid
        grdProducts.DataSource = ProductData.Tables(PRODUCT_TABLE_NAME)
        ' Dữ liệu đã có, cho phép người sử dụng lọc dữ liệu.
        btnFilter.Enabled = True
        isConnecting = False
      Catch exc As Exception
        If connectionString = SQL_CONNECTION_STRING Then
          ' Không thể kết nối SQL Server. Chuyển qua MSDE
          connectionString = MSDE_CONNECTION_STRING
          frmStatusMessage.Show("Connecting to MSDE")
        Else
          frmStatusMessage.Close()
          MessageBox.Show(CONNECTION_ERROR_MSG, _
           "Connection Failed!", MessageBoxButtons.OK, _
           MessageBoxIcon.Error)
          End
        End If
      End Try
    End While
  End Sub
End Class
```

12. In ấn đơn giản

Chương trình này biểu diễn cách in văn bản thuần túy (plain text).



Các chức năng chính:

- Sử dụng hộp thoại Page Setup
- Sử dụng hộp thoại Print Preview

Project: **In an don gian**

frmMain.vb

```
Option Strict On
Imports System.Drawing.Printing
Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Chú ý là tất cả các thủ tục sự kiện cùng làm việc trên
    ' cùng đối tượng PrintDocument
    Private WithEvents pdoc As New PrintDocument()

' Code phát sinh form được bỏ qua
```

```
' Hộp thoại PrintDialog cho phép người sử dụng chọn máy in để in
' hoặc chỉnh các thông số.
Private Sub btnPrintDialog_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrintDialog.Click
 Dim dialog As New PrintDialog()
 dialog.Document = pdoc
 If dialog.ShowDialog = DialogResult.OK Then
   pdoc.Print()
 End If
End Sub

' Xem trước các trang in
Private Sub btnPrintPreview_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrintPreview.Click
 Dim ppd As New PrintPreviewDialog()
 Try
   ppd.Document = pdoc
   ppd.ShowDialog()
 Catch exp As Exception
 MessageBox.Show("An error occurred while trying to load the " & _
    "document for Print Preview. Make sure you currently have " & _
    "access to a printer. A printer must be connected and " & _
    "accessible for Print Preview to work.", Me.Text, _
     MessageBoxButtons.OK, MessageBoxIcon.Error)
 End Try
End Sub

' Page setup cho phép người sử dụng chọn kích thước giấy, hướng
' giấy đứng (portrait), hoặc hướng giấy nằm ngang (landscape),...
Private Sub btnPageSetup_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPageSetup.Click
 Dim psd As New PageSetupDialog()
 With psd
   .Document = pdoc
   .PageSettings = pdoc.DefaultPageSettings
 End With
 If psd.ShowDialog = DialogResult.OK Then
   pdoc.DefaultPageSettings = psd.PageSettings
 End If
End Sub
```

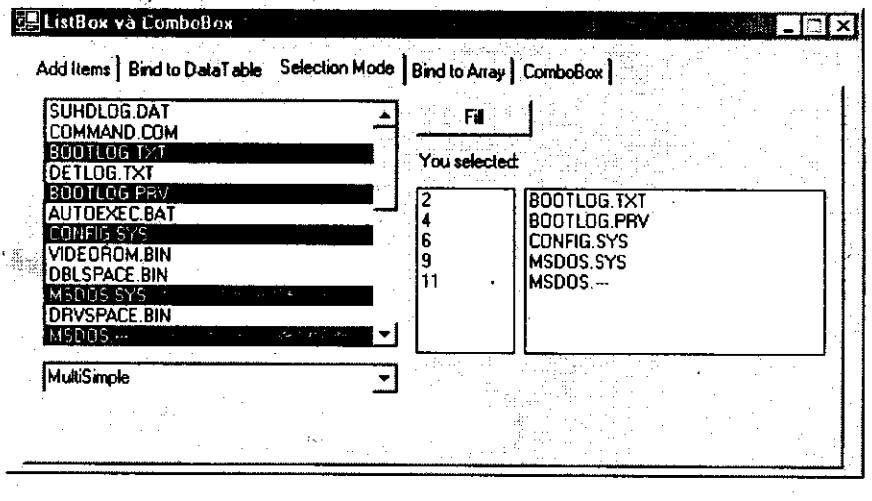
```
' Khởi tạo TextBox một ít văn bản để in.
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
 txtDocument.Text = "Plain text plain text plain text plain text plain text " & _
   vbCrLf & vbCrLf & vbTab & _
  "Plain text plain text plain text plain text plain text  plain text  " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain." & _
  vbCrLf & vbCrLf & vbTab & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text." & _
  vbCrLf & vbCrLf & vbTab & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plaintext plaintext plaintext plaintext plaintext  plain text " & _
  "Plaintext plaintext plaintext plaintext plaintext  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain text plain text  plain text " & _
  "Plain text plain text plain text plain."
End Sub

' Sự kiện PrintPage được kích hoạt khi bất cứ trang nào được in.
' Ta cũng có thể sử dụng các sự kiện BeginPrint và EndPrint để điều
' khiển nhiều hơn.
Private Sub pdoc_PrintPage(ByVal sender As Object, ByVal e As
System.Drawing.Printing.PrintPageEventArgs) Handles pdoc.PrintPage
' Khai báo một biến để lưu vị trí của ký tự cuối cùng được in. Khai báo
' là biến tĩnh để các sự kiện PrintPage sau có thể tham chiếu đến nó.
```

```
Static intCurrentChar As Int32
' Khởi tạo font dùng để in
Dim font As New font("Microsoft Sans Serif", 24)
Dim intPrintAreaHeight, intPrintAreaWidth, marginLeft, marginTop As Int32
With pdoc.DefaultPageSettings
  ' Khởi tạo các biến cục bộ chứa các giới hạn của vùng in.
  intPrintAreaHeight = .PaperSize.Height - .Margins.Top - .Margins.Bottom
  intPrintAreaWidth = .PaperSize.Width - .Margins.Left - .Margins.Right

  ' Khởi tạo các biến cục bộ lưu các giá trị lề để phục vụ làm
  ' tọa độ X và Y của góc trên bên trái của vùng in.
  ' Tọa độ X
  marginLeft = .Margins.Left
  ' Tọa độ Y
  marginTop = .Margins.Top
End With

' Nếu người sử dụng chọn Landscape mode, hoán chuyển chiều cao
' và chiều rộng của vùng in.
If pdoc.DefaultPageSettings.Landscape Then
  Dim intTemp As Int32
  intTemp = intPrintAreaHeight
  intPrintAreaHeight = intPrintAreaWidth
  intPrintAreaWidth = intTemp
End If

' Tính toán tổng số dòng trong tài liệu dựa trên chiều cao của vùng in
' và chiều cao của font.
Dim intLineCount As Int32 = CInt(intPrintAreaHeight / font.Height)
' Khởi tạo cấu trúc định nghĩa vùng in
Dim rectPrintingArea As New RectangleF(marginLeft, marginTop,
intPrintAreaWidth, intPrintAreaHeight)

' Sử dụng class StringFormat để lưu giữ thông tin trình bày văn bản
' (như canh lề và khoảng cách dòng), và các chức năng OpenType....
' Sử dụng StringFormat khiến cho MeasureString và DrawString sử
' dụng chỉ một số các dòng khi in mỗi trang.
Dim fmt As New StringFormat(StringFormatFlags.LineLimit)
```

```
    ' Gọi MeasureString để xác định số ký tự sẽ nằm trong vùng in.
    Dim intLinesFilled, intCharsFitted As Int32

    e.Graphics.MeasureString(Mid(txtDocument.Text, intCurrentChar + 1), font, _
            New SizeF(intPrintAreaWidth, intPrintAreaHeight), fmt, _
        intCharsFitted, intLinesFilled)
    ' In văn bản ra
    e.Graphics.DrawString(Mid(txtDocument.Text, intCurrentChar + 1), _
                          font, Brushes.Black, rectPrintingArea, fmt)
    intCurrentChar += intCharsFitted

    ' HasMorePages báo cho module in biết có phát sinh sự kiện
    ' PrintPage khác hay không.
    If intCurrentChar < txtDocument.Text.Length Then
      e.HasMorePages = True
    Else
      e.HasMorePages = False
    ' Phải khởi tạo lại intCurrentChar bằng 0 bởi vì nó là biến static
      intCurrentChar = 0
    End If
  End Sub
End Class
```

13. ListBox và ComboBox

Biểu diễn cách kết dữ liệu vào các control ListBox và ComboBox, và làm việc với các phần tử được chọn.



Các chức năng chính:

Biểu diễn cách sử dụng nhiều kỹ thuật khác nhau để đưa dữ liệu vào các control ListBox và ComboBox, và nhiều phương pháp khác nhau để lấy các phần tử được chọn. Trên control Tab có các trang sau:

- **Add Items** Biểu diễn cách thêm các đối tượng vào một ListBox, sử dụng thuộc tính DisplayMember của control để xác định thuộc tính nào của đối tượng sẽ hiển thị trong danh sách.
  Chú ý: Bởi vì ví dụ này không sử dụng thuộc tính DataSource của control, thay vào đó là thêm từng phần tử cụ thể, cho nên code không thể sử dụng thuộc tính ValueMember để lấy giá trị được chọn khi tiến hành chọn phần tử nào đó. Xem thủ tục lstProcessesAddItem_SelectedIndexChanged để rỏ hơn.
- **Selection Mode** Tìm hiểu các kiểu khác nhau của các ListBoxcho phép chọn nhiều phần tử.
- **Bind to DataTable** Cho dữ liệu gồm tất cả tên các file trong thư mục C:\ vào một DataTable, sau đó kết ListBox vào DataTable.
- **Bind to Array** Kết ListBox vào một mảng chứa các chương trình đang chạy – một mảng các đối tượng Process. Cho biết cách sử dụng các thuộc tính DisplayMember và ValueMember.
- **ComboBox** Biểu diễn cách kết ComboBox vào một DataTable lấy từ SQL Server. Giả sử database **Northwind** đã có trong SQL Server ở máy cục bộ. Trang này cũng cung cấp một số thuộc tính ảnh hưởng cách làm việc của ComboBox.

Project: **ListBox va ComboBox**

frmMain.vb

```
Option Strict On
Imports System.Diagnostics
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form

'Code phát sinh form được bỏ qua

    Private Sub btnFill1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnFill1.Click
     AddItems()
    End Sub
```

```
Private Sub btnFill2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFill2.Click
 BindToArray()
End Sub

Private Sub btnFill3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFill3.Click
 ' Kết một ListBox vào một DataTable chứa thông tin về các file.
 Try
   Dim dt As DataTable = FillTable("C:\")

   If Not (dt Is Nothing) Then
     With lstFiles
       .DisplayMember = "FileName"
       .ValueMember = "Length"
       .DataSource = dt
     End With
   End If

 Catch exp As Exception
   MessageBox.Show(exp.Message, Me.Text)
 End Try
End Sub

Private Sub btnFill4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFill4.Click
 FillSelectionMode()
End Sub

Private Sub btnFill5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFill5.Click
 BindToDataSet()
End Sub

Private Sub cboDemo_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cboDemo.SelectedIndexChanged

 lblResults.Text = cboDemo.SelectedValue.ToString
End Sub
```

```
Private Sub cboDropDownStyle_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboDropDownStyle.SelectedIndexChanged
 ' Lấy giá trị được liệt kê trong cboDropDownStyle, và gán vào
 ' thuộc tính DropDownStyle của cboDemo
 cboDemo.DropDownStyle = _
CType(System.Enum.Parse(GetType(ComboBoxStyle), _
  cboDropDownStyle.Text), ComboBoxStyle)
End Sub

Private Sub cboSelectionMode_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboSelectionMode.SelectedIndexChanged
 ' Cho phép người sử dụng chọn một trong các kiểu chọn sau:
 ' One, MultipleSimple, MultipleExtended
 Try
    lstMultiSelect.ClearSelected()
    lstMultiSelect.SelectionMode = _
     CType(System.Enum.Parse(GetType(SelectionMode), _
            cboSelectionMode.Text), SelectionMode)
 Catch exp As Exception
    MessageBox.Show(exp.Message, Me.Text)
 End Try
End Sub

Private Sub lstFiles_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles lstFiles.SelectedIndexChanged
 ' Hiển thị thông tin về file được chọn
 lblFileInfo.Text = "Length: " & lstFiles.SelectedValue.ToString
End Sub

Private Sub lstMultiSelect_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles lstMultiSelect.SelectedIndexChanged

' Hiển thị danh sách các phần tử được chọn
' Thuộc tính SelectedIndices trả về đối tượng SelectedIndexCollection.
' Sử dụng phương thức CopyTo của nó để chép các phần tử vào
' một mảng để có thể kết danh sách các phần tử được chọn vào ListBox.
 Try
    Dim aIndices(lstMultiSelect.SelectedIndices.Count - 1) As Integer
```

```
    lstMultiSelect.SelectedIndices.CopyTo(aIndices, 0)
    lstSelected.DataSource = aIndices

    ' Duyệt qua danh sách các phần tử được chọn
    With lstSelectedItems
      .Items.Clear()
      ' Begin/EndUpdate bật/tắt hiển thị của control khi đang thêm
      ' các phần tử. Chỉ để thấy việc cập nhật lại rõ ràng hơn.
      .BeginUpdate()
      Dim fi As FileInfo
      For Each fi In lstMultiSelect.SelectedItems
        .Items.Add(fi.Name)
      Next
      .EndUpdate()
    End With

  Catch exp As Exception
    lstSelected.DataSource = Nothing
  End Try
End Sub

Private Sub lstProcessesAddItem_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
lstProcessesAddItem.SelectedIndexChanged
  ' Bởi vì không sử dụng thuộc tính DataSource của control, nên không thể lấy thuộc
  ' tính SelectedValue. Thay vào đó, sử dụng thuộc tính SelectedItem.
  Try
    lblFileName1.Text = CType(lstProcessesAddItem.SelectedItem, _
                              Process).MainModule.FileName
  Catch exp As Exception

    ' Không làm gì cả nếu có bất kỳ exception nào xảy ra
    lblFileName1.Text = String.Empty
  End Try

End Sub

Private Sub lstProcessesDataSource_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
lstProcessesDataSource.SelectedIndexChanged
```

```
    Try
      lblFileName2.Text = CType(lstProcessesDataSource.SelectedValue, _
                                                ProcessModule).FileName
    Catch exp As Exception
      lblFileName2.Text = String.Empty
    End Try
  End Sub

  Private Sub nudDropDownItems_ValueChanged(ByVal sender As Object, ByVal e
  As System.EventArgs) Handles nudDropDownItems.ValueChanged
    cboDemo.MaxDropDownItems = CInt(nudDropDownItems.Value)
  End Sub

  Private Sub nudDropDownWidth_ValueChanged(ByVal sender As System.Object,
  ByVal e As System.EventArgs) Handles nudDropDownWidth.ValueChanged
    cboDemo.DropDownWidth = CInt(nudDropDownWidth.Value)
  End Sub

  Private Sub AddItems()
    ' Có thể thêm từng phần tử vào một listbox hoặc combobox.
    ' Bởi vì ta có thể thêm bất kỳ kiểu đối tượng nào, và tùy vào ta sử
    ' dụng thuộc tính nào của đối tượng để hiển thị. Quy định thuộc tính
    ' DisplayMember để chỉ định tên của thuộc tính muốn hiển thị.
    ' Trong trường hợp này, sử dụng thuộc tính ProcessName để hiển thị

    Try
      Dim prc As Process
      ' Loại bỏ tất cả các phần tử đã có ra khỏi control
      lstProcessesAddItem.Items.Clear()
      ' Cho dữ liệu vào control. Chỉ định thành phần nào sẽ được hiển thị
      lstProcessesAddItem.DisplayMember = "ProcessName"
      For Each prc In Process.GetProcesses()
        lstProcessesAddItem.Items.Add(prc)
      Next
      lstProcessesAddItem.Sorted = True
    Catch exp As Exception
      MessageBox.Show(exp.Message, Me.Text)
    End Try
  End Sub
```

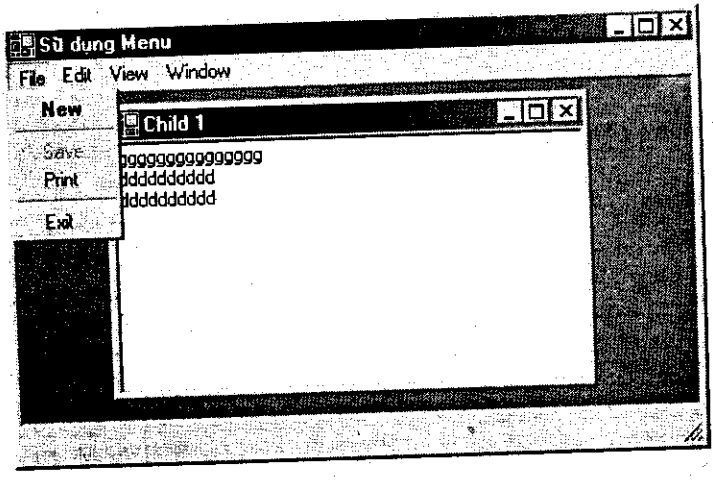
```
Private Sub BindToArray()
 ' Kết vào một mảng khá đơn giản – chỉ cần gán vào thuộc tính
 ' DataSourrce của control. Trong trường hợp này, cũng có thể sử
 ' dụng thuộc tính ValueMember.

 Try
   Dim prc As Process
   With lstProcessesDataSource
     .ValueMember = "MainModule"
     .DisplayMember = "ProcessName"
     .DataSource = Process.GetProcesses()
   End With

 Catch exp As Exception
   MessageBox.Show(exp.Message, Me.Text)
 End Try
End Sub

Private Sub BindToDataSet()
 Try
   ' Kết vào một DataTable
   Dim ds As New DataSet()
   ' SqlDataAdapter1 lấy dữ liệu từ bảng Products trong database
    ' Northwind trên SQL Server
   SqlDataAdapter1.Fill(ds)
   cboDemo.ValueMember = "ProductID"
   cboDemo.DisplayMember = "ProductName"
   cboDemo.DataSource = ds.Tables(0)

 Catch exp As Exception
   MessageBox.Show(exp.Message, Me.Text)
 End Try
End Sub

Private Sub FillSelectionMode()
 Try
   Dim di As New DirectoryInfo("C:\")
   With lstMultiSelect
     .DisplayMember = "Name"
```

```
        .ValueMember = "Length"
        .DataSource = di.GetFiles()

        ' Khởi tạo combobox chứa các kiểu chọn khác nhau
        cboSelectionMode.Text = .SelectionMode.ToString
      End With

    Catch exp As Exception
      MessageBox.Show(exp.Message, Me.Text)
    End Try
  End Sub

  Private Function FillTable(ByVal Path As String) As DataTable
    ' Xây dựng một DataTable chứa thông tin về các file trên đĩa cứng
    Dim dt As New DataTable()
    Dim dr As DataRow
    Try
      dt.Columns.Add("FileName", GetType(System.String))
      dt.Columns.Add("Length", GetType(System.Int64))
      ' Class DirectoryInfo trong namespace System.IO
      Dim di As New DirectoryInfo(Path)

      ' Lấy thông tin tất cả các file không có thuộc tính System, Hidden
      Dim fi As FileInfo
      For Each fi In di.GetFiles()
        If (fi.Attributes And (FileAttributes.Hidden Or FileAttributes.System)) = 0 Then
          dr = dt.NewRow()
          dr("FileName") = fi.Name
          dr("Length") = fi.Length
          dt.Rows.Add(dr)
        End If
      Next

    Catch exp As Exception
      MessageBox.Show(exp.Message, Me.Text)
    End Try

    Return dt
  End Function
```

```
Private Sub LoadValues()
 ' ' Cho các kiểu chọn có thể vào cboSelectionMode.
 ' cboSelectionMode.DataSource = _
 ' System.Enum.GetNames(GetType(SelectionMode))
cboDropDownStyle.DataSource = _
 System.Enum.GetNames(GetType(ComboBoxStyle))

 ' Quản lý các thuộc tính của cboDemo
 With nudDropDownWidth
   .Value = cboDemo.DropDownWidth
   .Minimum = cboDemo.Width
   .Maximum = cboDemo.Width * 2
 End With
 nudDropDownItems.Value = cboDemo.MaxDropDownItems
End Sub

Private Sub TabControl1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TabControl1.SelectedIndexChanged
 ' Bảo đảm là danh sách các phần tử  được chọn được xóa  khi rời khỏi tab
 lstSelected.DataSource = Nothing
End Sub
End Class
```

14. Sử dụng các menu

Chương trình này biểu diễn cách sử dụng các menu trong các ứng dụng Windows.



Các control và các thuộc tính chính:

Loại	Thuộc tính	Giá trị
Form	(Name)	frmMain
	IsMdiContainer	True
	Menu	mnuMain
StatusBar	(Name)	sbarEdit
MainMenu	(Name)	mnuMain
MenuItem (File)	(Name)	mnuFile
	MergeType	MergeItems
	Text	&File
MenuItem (File | New)	(Name)	mnuNew
	MergeType	MergeItems
	Text	&New
MenuItem (File | Exit)	(Name)	mnuExit
	MergeOrder	10
	MergeType	MergeItems
	Text	E&xit
MenuItem (View)	(Name)	mnuView
	MergeOrder	10
	MergeType	Add
	Text	&View
MenuItem (View | StatusBar)	(Name)	mnuStatusBar
	Checked	True
	Text	&Status Bar
MenuItem (Windows)	(Name)	mnuWindow
	MdiList	True
	MergeOrder	20
	MergeType	Add
	Text	&Window
MenuItem (Window | Cascade)	(Name)	mnuCascade
	Text	&Cascade
(Window | Title Vertical)	(Name)	mnuTitleVertical
	Text	Title &Vertical
(Window | Title Horizontal)	(Name)	mnuTitleHorizontal
	Text	Title &Horizontal
Form	(Name)	frmEdit
	Menu	MainMenu1
TextBox	(Name)	txtEdit
	Dock	Fill
MainMenu	(Name)	MainMenu1
MenuItem (File)	(Name)	mnuFile
	MergeType	MergeItems

	Text	&File
MenuItem (File I Save)	(Name)	mnuSave
	MergeOrder	2
	MergeType	MergeItems
	Text	&Save
MenuItem (File I Print)	(Name)	mnuPrint
	MergeOrder	3
	MergeType	MergeItems
	Text	&Print
MenuItem (Edit)	(Name)	mnuEdit
	MergeOrder	5
	Text	&Edit
MenuItem (Edit I Undo)	(Name)	mnuUndo
	Text	&Undo
MenuItem (Edit I Cut)	(Name)	mnuCut
	Shortcut	CtrlX
	Text	Cu&t
MenuItem (Edit I Copy)	(Name)	mnuCopy
	Shortcut	CtrlC
	Text	&Copy
MenuItem (Edit I Paste)	(Name)	mnuPaste
	Shortcut	CtrlV
	Text	&Paste
MenuItem (Edit I Text Size)	(Name)	mnuSize
	Text	Text &Size
Text Size I Small	(Name)	mnuSmall
	Checked	True
	RadioCheck	True
	Text	&Small
Text Size I Medium	(Name)	mnuMedium
	Checked	True
	RadioCheck	True
	Text	&Medium
Text Size I Large	(Name)	mnuLarge
	Checked	True
	RadioCheck	True
	Text	&Large

Các chức năng chính:

Chương trình này cho phép sử dụng control MainMenu trong ngữ cảnh là một ứng dụng MDI đơn giản. Chương trình sử dụng hầu hết các tính năng lúc thiết kế của control MainMenu, kể cả trộn menu, viết code các sự kiện (Popup, Select, và Click). Ta cũng tìm hiểu cách sử dụng các thuộc tính

MdiList, RadioCheck, và Checked. Ngoài ra, chương trình còn biểu diễn cách xử lý nhiều sự kiện, hiển thị trên thanh trạng thái thông tin về các menu,...

Chương trình này cho thấy được nhiều tính năng thú vị của các menu và thanh công cụ trong các ứng dụng Windows:

- Trộn các menu item có trong các form con với form cha. Xem các thuộc tính MergeOrder và MergeType. Một số menu được thêm vào menu cha (ví dụ menu Edit trên frmEdit) và một số khác được trộn với menu cha (ví dụ: các phần tử trong menu Edit). Xét các thuộc tính MergeType và MergeOrder của các menu File trên frmMain và frmEdit. Thuộc tính MergeType được gán là MergeItems, nghĩa là menu File trên frmMain trộn với các phần tử trong menu File của frmEdit. Menu Edit trên frmEdit có thuộc tính MergeType có giá trị là Add – nghĩa là menu Edit của nó được thêm vào menu của form cha. Thuộc tính MergeOrder điều khiển vị trí các phần tử được trộn trong cấu trúc menu. Các giá trị của thuộc tính này có liên quan đến nhau, nhưng giá trị thực sự thì không quan trọng, ngoại trừ trong mối quan hệ với các phần tử khác. Trên frmMain, menu File | New có MergeOrder là 0, và menu File | Exit có MergeOder là 10. Trên frmEdit, giá trị này của menu File | Save là 2, File | Print là 3. Các đường phân tách có giá trị là 1 và 4. Khi chọn File | New trong form chính, form con xuất hiện, và các menu của nó trộn với menu File của form chính, và menu Edit được thêm vào form chính. Thuộc tính MergeOrder của menu chính trong form cha là 0, 10, 20, 30, 40. Menu Edit trên frmEdit có thuộc tính MergeOrder có giá trị là 5. Khi form con được chọn, menu Edit xuất hiện giữa hai menu đầu tiên của form cha (bởi vì 5 lớn hơn 0 và nhỏ hơn 10).
- Sự kiện Popup của menu xảy ra khi menu chuẩn bị xuất hiện. Sự kiện này cho phép quyết định cấp phép sử dụng các menu con. Xét menu File của frmMain và menu Edit của frmEdit, mỗi menu sử dụng một sự kiện Popup để cho phép và không cho phép sử dụng các phần tử trong menu.
- Menu Window có thuộc tính MdiList được gán giá trị là True, cho nên nó hiển thị một danh sách các cửa sổ con.
- Một số menu có các phím tắt (menu Edit của frmEdit). có thể hiển thị hoặc ẩn thông tin này bằng cách sử dụng thuộc tính ShowShortCut.
- Một phần tử của một menu có thể được chọn làm phần tử mặc nhiên thông qua thuộc tính DefaultItem. Xem menu File | New trên form chính.
- Menu View|Status Bar sử dụng menu đánh dấu để bật hoặc tắt hiển thị thanh trạng thái. Xem thủ tục mnuView_Popup để biết cách đánh dấu

một menu. Xem thủ tục mnuStatusBar_Click để thấy cách chuyển đổi trạng thái.

- Xử lý các menu RadioCheck thì phụ thuộc vào code. Menu Edit | Text Size cho phép chọn các kích thước font. Chọn một phần tử thì phải đánh dấu nó và xóa đánh dấu các phần tử còn lại.
- Có thể thay đổi thuộc tính Text của một menu vào bất kỳ lúc nào. Xem xử lý của mnuUndo trong frmEdit (thủ tục HandleEditItems).

Project: **Su dung Menu**

frmMain.vb

```
Option Strict On
Public Class frmMain
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

  Private Sub mnuCascade_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuCascade.Click
   Me.LayoutMdi(MdiLayout.Cascade)
  End Sub

  Private Sub mnuNew_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuNew.Click
   ' Thực hiện tạo một form con mới
   AddNew()
  End Sub

  Private Sub AddNew()
   Try
     Dim frm As New frmEdit()
     Static intChild As Integer
     intChild += 1
     frm.MdiParent = Me
     frm.Text = "Child " & intChild
     frm.Show()
   Catch exp As Exception
     MessageBox.Show(exp.Message, Me.Text)
   End Try
  End Sub
```

```
Private Sub mnuStatusBar_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles mnuStatusBar.Click
 ' Bật/tắt hiển thị thanh trạng thái
 Dim blnShow As Boolean = Not mnuStatusBar.Checked
 mnuStatusBar.Checked = blnShow
 sbarEdit.Visible = blnShow
End Sub

Private Sub mnuTileHorizontal_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuTileHorizontal.Click
 Me.LayoutMdi(MdiLayout.TileHorizontal)
End Sub

Private Sub mnuTileVertical_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuTileVertical.Click
 Me.LayoutMdi(MdiLayout.TileVertical)
End Sub

Private Sub mnuView_Popup(ByVal sender As Object, ByVal e As
System.EventArgs) Handles mnuView.Popup
 ' Xác định trạng thái của menu StatusBar
 mnuStatusBar.Checked = sbarEdit.Visible
End Sub

Private Sub mnuWindow_Popup(ByVal sender As Object, ByVal e As
System.EventArgs) Handles mnuWindow.Popup
 ' Enable/disable các menu trong menu Window
 Dim blnEnable As Boolean = (Me.MdiChildren.Length > 0)
 mnuCascade.Enabled = blnEnable
 mnuTileHorizontal.Enabled = blnEnable
 mnuTileVertical.Enabled = blnEnable
End Sub

Private Sub ClearStatus(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuCascade.Click, mnuExit.Click, mnuFile.Click,
mnuNew.Click, mnuStatusBar.Click, mnuTileHorizontal.Click, mnuTileVertical.Click,
mnuView.Click, mnuWindow.Click
 ' Sau khi chọn một menu, xóa nội dung trên thanh trạng thái
 ClearStatusBar()
End Sub
```

```
    Public Sub ClearStatusBar()
     sbarEdit.Text = String.Empty
    End Sub

    Private Sub HandleSelect(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuStatusBar.Select, mnuCascade.Select,
    mnuExit.Select, mnuNew.Select, mnuTileHorizontal.Select, mnuTileVertical.Select,
    mnuFile.Select, mnuView.Select, mnuWindow.Select
     Dim strText As String
     If sender Is mnuStatusBar Then
       strText = "Toggle display of the status bar"
     ElseIf sender Is mnuCascade Then
       strText = "Cascade child windows"
     ElseIf sender Is mnuExit Then
       strText = "Exit demonstration"
     ElseIf sender Is mnuNew Then
       strText = "Create new child window"
     ElseIf sender Is mnuTileHorizontal Then
       strText = "Tile windows horizontally"
     ElseIf sender Is mnuTileVertical Then
       strText = "Tile windows vertically"
     Else
       strText = String.Empty
     End If
     WriteToStatusBar(strText)
    End Sub

    Public Sub WriteToStatusBar(ByVal Text As String)
     ' Thủ tục này Public để các form con có thể gọi nó
     sbarEdit.Text = Text
    End Sub
End Class
```

frmEdit.vb

```
Public Class frmEdit
    Inherits System.Windows.Forms.Form

'Code phát sinh form được bỏ qua
```

```
Private Enum FontSize
    Small = 8
    Medium = 12
    Large = 16
End Enum
Private mblnModified As Boolean = False

Private Sub mnuCopy_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuCopy.Click
    Copy()
End Sub

Private Sub mnuCut_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuCut.Click
    Cut()
End Sub

Private Sub mnuEdit_Popup(ByVal sender As Object, ByVal e As
System.EventArgs) Handles mnuEdit.Popup
    HandleEditItems()
End Sub

Private Sub mnuFile_Popup(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFile.Popup
    HandleFileItems()
End Sub

Private Sub mnuLarge_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuLarge.Click
    SetLargeFont()
End Sub

Private Sub mnuMedium_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuMedium.Click
    SetMediumFont()
End Sub

Private Sub mnuPaste_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuPaste.Click
    Paste()
End Sub
```

```
Private Sub mnuPrint_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuPrint.Click
    Print()
End Sub

Private Sub mnuSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuSave.Click
    Save()
End Sub

Private Sub mnuSmall_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuSmall.Click
    SetSmallFont()
End Sub

Private Sub mnuUndo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuUndo.Click
    Undo()
End Sub

Private Sub txtEdit_ModifiedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtEdit.ModifiedChanged
    ' Khi TextBox có sự thay đổi, thay đổi giá trị mblnModified
    mblnModified = True
End Sub

Private Sub ClearStatus(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuCopy.Click, mnuCut.Click, mnuEdit.Click,
mnuFile.Click, mnuLarge.Click, mnuMedium.Click, mnuPaste.Click, mnuPrint.Click,
mnuSave.Click, mnuSize.Click,     mnuSmall.Click, mnuUndo.Click
    ClearStatusBar()
End Sub

Private Sub ClearStatusBar()
    Try
        CType(Me.ParentForm, frmMain).ClearStatusBar()
    Catch
        ' Không làm gì cả
    End Try
End Sub
```

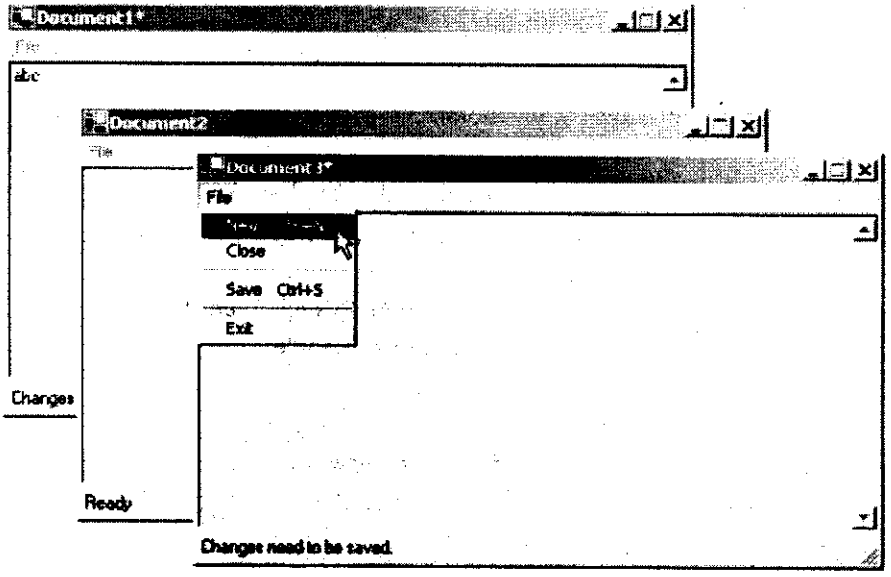
```
Public Sub Copy()
    txtEdit.Copy()
End Sub

Public Sub Cut()
    txtEdit.Cut()
End Sub

Private Sub HandleEditItems()
    ' Xác định có cho phép sử dụng các menu Cut/Copy/Paste

    ' Paste được cho phép khi đối tượng Clipboard chứa văn bản
    Dim iData As IDataObject = Clipboard.GetDataObject()
    mnuPaste.Enabled = iData.GetDataPresent(GetType(String))

    ' Cut và Copy được cho phép nếu có văn bản được chọn trong TextBox
    Dim blnEnable As Boolean = (txtEdit.SelectedText.Length > 0)
    mnuCut.Enabled = blnEnable
    mnuCopy.Enabled = blnEnable

    ' Xử lý Undo
    If txtEdit.CanUndo Then
        mnuUndo.Text = "&Undo"
        mnuUndo.Enabled = True
    Else
        mnuUndo.Text = "Nothing to Undo"
        mnuUndo.Enabled = False
    End If
End Sub

Private Sub HandleFileItems()

    mnuPrint.Enabled = (txtEdit.Text.Length > 0)
    mnuSave.Enabled = mblnModified

End Sub
```

```
Private Sub HandleStatus(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuSave.Select, mnuCopy.Select, mnuCut.Select,
mnuLarge.Select, mnuMedium.Select, mnuSmall.Select, mnuPrint.Select,
mnuPaste.Select, mnuUndo.Select, mnuSize.Select, mnuEdit.Select,
mnuFile.Select
    Dim strText As String
    If sender Is mnuSave Then
        strText = "Save the current contents"
    ElseIf sender Is mnuCopy Then
        strText = "Copy selected text to the clipboard"
    ElseIf sender Is mnuCut Then
        strText = "Cut selected text to the clipboard"
    ElseIf sender Is mnuLarge Then
        strText = "Display text using large font"
    ElseIf sender Is mnuMedium Then
        strText = "Display text using medium font"
    ElseIf sender Is mnuSmall Then
        strText = "Display text using small font"
    ElseIf sender Is mnuPrint Then
        strText = "Print the current contents"
    ElseIf sender Is mnuPaste Then
        strText = "Paste the contents of the clipboard"
    ElseIf sender Is mnuUndo Then
        strText = "Undo changes to contents"
    Else
        strText = String.Empty
    End If
    CType(Me.ParentForm, frmMain).WriteToStatusBar(strText)
End Sub

Public Sub Paste()
    txtEdit.Paste()
End Sub

Public Sub Print()
    MessageBox.Show("Add code to print your document here", Me.Text)
End Sub
```

```
Public Sub Save()
    MessageBox.Show("Add code to save your document here", Me.Text)
End Sub

Public Sub SetLargeFont()
    SetSize(FontSize.Large)
End Sub

Public Sub SetMediumFont()
    SetSize(FontSize.Medium)
End Sub

Public Sub SetSmallFont()
    SetSize(FontSize.Small)
End Sub

Private Sub SetSize(ByVal Size As FontSize)
    Dim fnt As New Font(txtEdit.Font.FontFamily, CType(Size, Integer))
    txtEdit.Font = fnt

    ' Xóa đánh dấu tất cả
    mnuSmall.Checked = False
    mnuMedium.Checked = False
    mnuLarge.Checked = False
    ' Đánh dấu phần tử thích hợp
    Select Case Size
        Case FontSize.Small
            mnuSmall.Checked = True
        Case FontSize.Medium
            mnuMedium.Checked = True
        Case FontSize.Large
            mnuLarge.Checked = True
    End Select
End Sub

Public Sub Undo()
    txtEdit.Undo()
End Sub
End Class
```

15. Quản lý nhiều form cấp cao nhất (Top-Level)

Biểu diễn một ứng dụng có nhiều form tương tự như Microsoft Word.



Các chức năng chính:

- Nhiều ứng dụng làm việc với nhiều cửa sổ tài liệu (mỗi cửa sổ mở một tài liệu), được hiển thị như là các cửa sổ cấp cao nhất (Top Level) xuất hiện trong danh sách các tác vụ (Task List). Điều này cho phép người sử dụng nhấn **Alt + Tab** để chuyển qua lại giữa các tài liệu như là chúng được mở trong các tiến trình khác nhau, nhưng thực sự không phải như vậy. Chương trình này cũng hoạt động như trên.
- Khi ứng dụng bắt đầu, thông qua Sub Main trong class **Forms**, nó tạo một form instance mới bằng cách gọi phương thức NewForm của class và thêm form này vào tập hợp các form. Ngay khi form đầu tiên được tạo, dòng lệnh Application.Run được gọi để bắt đầu tiến trình ứng dụng chính để khi đóng form đầu tiên thì không thoát ứng dụng. Mỗi tài liệu có một menu item File | New. Nhấn vào menu gọi thủ tục Forms.NewForm để mở một tài liệu mới giống như làm trong Sub Main.
- Các form còn có các menu item File | Close và File | Exit. Chọn File | Colse để bắt đầu đóng tiến trình. Mỗi form chó một sự kiện đóng form riêng để cho phép kiểm tra nội dung có thay đổi hay chưa (thông qua thuộc tính Dirty), nếu có thì hỏi có lưu lại hay không.

Project: **Quan ly nhieu Top-Level Form**

frmMain.vb

```
Option Strict On
Public Class frmMain
  Inherits System.Windows.Forms.Form

  ' Các sự kiện này được sử dụng để thông báo cho class Forms khi
  ' người sử dụng hủy thao tác Save trong sự kiện Form_Closing hoặc
  ' khi chọn Exit từ menu File.
  Public Event SaveWhileClosingCancelled As System.EventHandler
  Public Event ExitApplication As System.EventHandler

  Private m_Dirty As Boolean = False
  Private m_ClosingComplete As Boolean = False
  Private m_DocumentName As String
  Private m_FileName As String

' Code phát sinh form được bỏ qua

  Public ReadOnly Property ClosingComplete() As Boolean
   Get
     Return m_ClosingComplete
   End Get
  End Property

  Public ReadOnly Property DocumentName() As String
   Get
     Return m_DocumentName
   End Get
  End Property

  Public Property Dirty() As Boolean
   ' Thuộc tính này dùng để xác định có lưu lại dữ liệu trước khi thoát
   Get
     Return m_Dirty
   End Get
```

```
Set(ByVal Value As Boolean)
    If Value Then
      If Not Me.Text.EndsWith("*") Then
        Me.Text = Me.Text & "*"

        Me.sbDocInfo.Text = "Changes need to be saved."
      End If
    Else
      Me.sbDocInfo.Text = "Ready"
      ' Xóa dấu *
      Me.Text = Me.Text.Substring(0, (Me.Text.Length - 1))
    End If
    m_Dirty = Value
  End Set
 End Property

 Public Property FileName() As String
  Get
    Return m_FileName
  End Get
  Set(ByVal Value As String)
    m_FileName = Value
    m_DocumentName = _
      System.IO.Path.GetFileNameWithoutExtension(m_FileName)
    Me.Text = Me.DocumentName
  End Set
 End Property

 Private Sub frmMain_Closing(ByVal sender As Object, ByVal e As
 System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
  Try
    ' Gán giá trị cho thuộc tính m_ClosingComplete bằng True để
    ' class Forms biết là có thể loại bỏ nó ra khỏi tập hợp các form.
    m_ClosingComplete = True

    If Me.Dirty Then
' Hỏi người sử dụng Lưu (Yes), không lưu (No), hoặc ngừng việc đóng form (Cancel).
      Dim strDocTitle As String
```

```
        If Me.Text.EndsWith("*") Then
          strDocTitle = Me.Text.Substring(0, (Me.Text.Length - 1))
        Else
          strDocTitle = Me.Text
        End If

        Dim strMsg As String = String.Format("Do you want to save {0}?", strDocTitle)

        Select Case MessageBox.Show(strMsg, "Closing", _
                        MessageBoxButtons.YesNoCancel, _
                        MessageBoxIcon.Exclamation, _
                        MessageBoxDefaultButton.Button1)
          Case DialogResult.Yes
            ' Lưu tài liệu
            Me.SaveDocument()

          Case DialogResult.No
            ' Không lưu và thoát
            ' Viết thêm code ở đây
          Case DialogResult.Cancel
            ' Dừng việc đóng form
            e.Cancel = True
            ' Nếu hủy việc đóng form, cần để nó trong tập hợp các Forms
            m_ClosingComplete = False
            ' Phát sinh một sự kiện để ứng dụng không đóng các tài liệu khác
            RaiseEvent SaveWhileClosingCancelled(Me, Nothing)

        End Select
      End If
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, _
                MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
  End Sub

  Private Sub mnuExit_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuExit.Click
    ' Thoát ứng dụng
    RaiseEvent ExitApplication(Me, Nothing)
  End Sub
```

```
Private Sub mnuNew_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNew.Click
 ' Create a new document instnace
 Forms.NewForm()
End Sub

Private Sub mnuSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuSave.Click
 Me.SaveDocument()
End Sub

Private Sub mnuClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuClose.Click
 Me.Close()
End Sub

Private Sub txtData_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtData.TextChanged
 ' Nếu nội dung được thay đổi, gán thuộc tính Dirty của form là True
 Me.Dirty = True
End Sub

Private Sub SaveDocument()
 ' Đoạn code này không thực hiện lưu file thự sự
 Try
   ' Kiểm tra tài liệu có thay đổi hay chưa
   If Me.Dirty Then
     ' Kiểm tra xem ta đã có tên file của tài liệu chưa
     If Not Me.FileName Is Nothing Then
       ' Lưu vào tài liệu đã tồn tại
     Else
   ' Chưa có tên file, hỏi người sử dụng nhập vào tên file.
     ' Sử dụng các hộp thoại thông dụng
   ' Tạo một tên file dựa vào tên của tài liệu và  đường dẫn của ứng dụng hiện hành
       Me.FileName = AppDomain.CurrentDomain.BaseDirectory Saved" & Me.Text
     End If
     ' Khi tài liệu đã được lưu, khởi tạo lại giá trị của Dirty
     Me.Dirty = False
   End If
```

```
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, _
                  MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
  End Sub
End Class

Public Class Forms
  ' Tập hợp này dùng để quản lý danh sách các form tài liệu.
  Private Shared m_Forms As New Collection()

  ' Biến đếm dùng để tạo các tiêu đề mặc nhiên cho mỗi form
  Private Shared m_FormsCreated As Integer = 0
  ' Thuộc tính này dùng để quyết định dừng việc thoát ứng dụng nếu
  ' người sử dụng chọn Cancel trên hộp thoại hỏi có lưu (Save) hay không.
  Private Shared m_CancelExit As Boolean = False
  ' Được sử dụng để kiểm tra có đang ở trong quá trình thoát
  Private Shared m_ShutdownInProgress As Boolean = False

  ' Số form hiện đang được mở
  Public Shared ReadOnly Property Count() As Integer
   Get
     Return m_Forms.Count
   End Get
  End Property

  Public Shared Sub Main()
  ' Mở cửa sổ tài liệu đầu tiên
  Try

    Forms.NewForm()

  Catch exp As Exception
    ' Khi có vấn đề
    MessageBox.Show("Sorry, we were unable to load a document. Good Bye.", _
          "Application Main", MessageBoxButtons.OK, MessageBoxIcon.Error)
    Application.Exit()
  End Try
```

```
    ' Quy định tiến trình chính chạy bên ngoài sự điều khiển của một form cụ thể để khi
    ' đóng một tài liệu thì không thoát toàn bộ chương trình.
    Application.Run()
End Sub

Public Shared Sub NewForm()
    Try
        Forms.m_FormsCreated += 1
        Dim frm As New frmMain()
        frm.Text = "Document" & Forms.m_FormsCreated.ToString()
        m_Forms.Add(frm, frm.GetHashCode.ToString())
        ' Móc (hook) sự kiện Closed của form để ta biết
        ' khi đóng cửa sổ tài liệu
        AddHandler frm.Closed, AddressOf Forms.frmMain_Closed

        ' Hook sự kiện SaveWhileClosingCancelled để ta biết người sử
        ' dụng có nhấn nút Cacel khi được hỏi lưu.
        AddHandler frm.SaveWhileClosingCancelled, AddressOf
Forms.frmMain_SaveWhileClosingCancelled
        ' Hook sự kiện ExitApplication để biết người sử dụng có muốn
        ' thoát toàn bộ chương trình bằng cách chọn menu Exit từ một cửa sổ tài liệ
        AddHandler frm.ExitApplication, AddressOf Forms.frmMain_ExitApplication

        ' Hiển thị form
        frm.Show()

    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK,
MessageBoxIcon.Error)
        If Forms.Count = 0 Then
            ' Đẩy lỗi qua cho Main để có thể thoát chương trình
            Throw exp
        End If
    End Try

End Sub
```

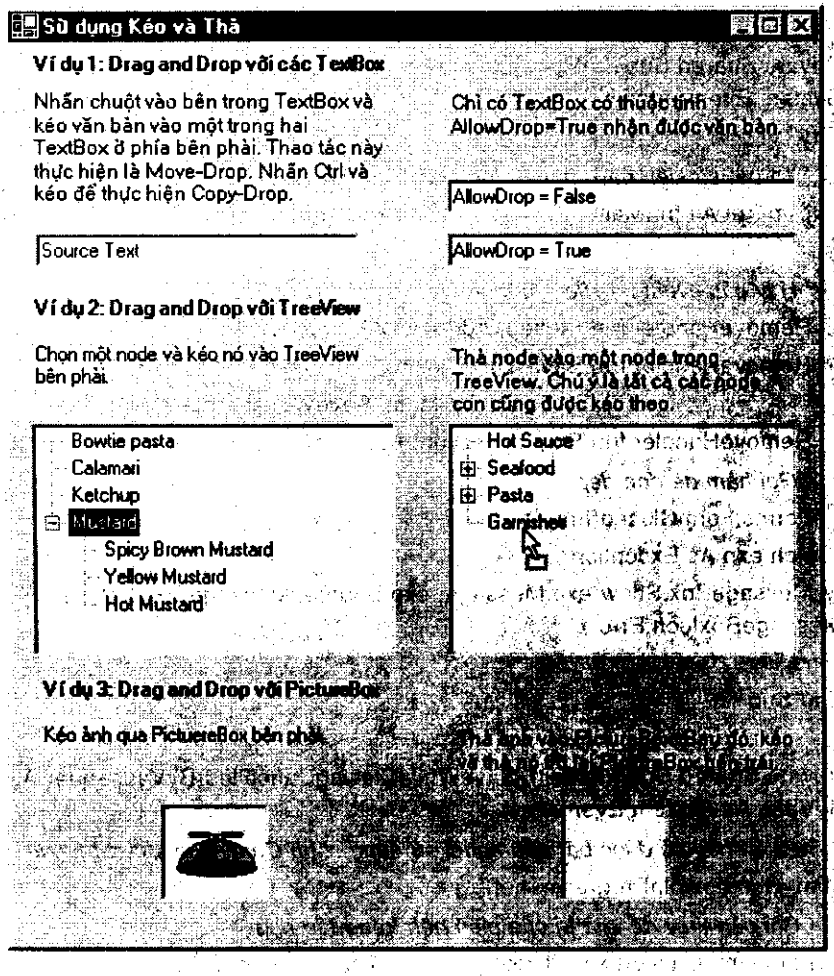
```
Private Shared Sub FormClosed(ByVal frm As frmMain)
 ' Loại bỏ form đã được đóng ra khỏi tập hợp
 m_Forms.Remove(frm.GetHashCode.ToString())
 ' Nếu không còn form nào, thoát chương trình
 If m_Forms.Count = 0 Then
   Application.Exit()
 End If
End Sub

Public Shared Sub ExitApp()
 Try
   m_ShutdownInProgress = True
   ' Thoát khi tất cả các form đã được đóng
   Dim frm As frmMain
   Dim i As Integer
   ' Duyệt qua tập hợp từng form một, ra lệnh cho mỗi form tự đóng.
    ' Chỉ hỏi lưu lại khi form có nội dung được thay đổi. Nếu người sử
    ' dụng chọn Cancel thì không đóng form đó.
   For i = m_Forms.Count To 1 Step -1
     frm = CType(m_Forms(i), frmMain)
     If frm.Dirty Then
       frm.Close()
     End If
     ' Kiểm tra biến cờ (flag) để biết người sử dụng có dừng việc thoát chương trình
     If m_CancelExit = True Then
       m_CancelExit = False
       Exit Sub
     End If
   Next
   ' Đóng tất cả các tài liệu không có thay đổi nội dung.
   If m_Forms.Count > 0 Then
     For i = m_Forms.Count To 1 Step -1
       frm = CType(m_Forms(i), frmMain)
       frm.Close()
     Next
   End If
 Catch exp As Exception
   MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK,
MessageBoxIcon.Error)
```

```
    Application.Exit()
  Finally
    m_ShutdownInProgress = False
  End Try
 End Sub

 Private Shared Sub frmMain_Closed(ByVal sender As Object, ByVal e As
 System.EventArgs)
  Try
    ' Ta bắt sự kiện này khi một form đã hoàn tất việc đóng
    Dim frm As frmMain = CType(sender, frmMain)

    ' Hủy các event handler đã thêm vào lúc tạo.
    RemoveHandler frm.Closed, AddressOf Forms.frmMain_Closed
    RemoveHandler frm.SaveWhileClosingCancelled, AddressOf _
                  Forms.frmMain_SaveWhileClosingCancelled
    RemoveHandler frm.ExitApplication, AddressOf Forms.frmMain_ExitApplication
    ' Gọi hàm để dọn dẹp
    Forms.FormClosed(frm)
  Catch exp As Exception
    MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK,
 MessageBoxIcon.Error)
  End Try
 End Sub

 Private Shared Sub frmMain_SaveWhileClosingCancelled(ByVal sender As Object,
 ByVal e As System.EventArgs)
  ' Sự kiện này sẽ được bắt nếu người sử dụng chọn Cancel khi được hỏi lưu.
  If m_ShutdownInProgress Then
    ' Chỉ cần thay đổi giá trị của biến bên trong tập hợp
    Forms.m_CancelExit = True
  End If
 End Sub

 Private Shared Sub frmMain_ExitApplication(ByVal sender As Object, ByVal e As
 System.EventArgs)
  ' Sự kiện này được bắt khi người sử dụng chọn menu Exit
  Forms.ExitApp()
 End Sub
End Class
```

16. Sử dụng Kéo và Thả (Drag and Drop)

Trong chương trình này ta học cách sử dụng chức năng kéo và thả với Visual Basic .NET.

Để sử dụng chức năng này, các control phải có thuộc tính AllowDrop=True.



Các chức năng chính:

Chương trình này có ba ví dụ về kéo và thả:

- Từ một TextBox vào hai TextBox khác, một trong hai TextBox có thuộc tính AllowDrop = False để cấm thả vào nó.
- Từ một TreeView vào một TreeView khác, biểu diễn cách di chuyển các node từ TreeView này sang TreeView khác.
- Từ một PictureBox sang một PictureBox khác, biểu diễn cách sao chép ảnh từ một control này sang một control khác.

Có thể thực hiện sao chép (Copy) hoặc di chuyển (Move) khi kéo và thả. Để Copy nhấn và giữ phím Ctrl trong khi kéo. Khi làm như vậy, chú ý con trỏ chỉ Copy-Drop (hình chữ nhật và dấu +). Con trỏ Move chỉ có hình chữ nhật.

Chú ý: Dữ liệu có thể Drag được có thể là một instance của class String, Bitmap, hoặc MetaFile, hoặc một đối tượng có cài đặt các interface ISerializable hoặc IDataObject.

Phiên bản hiện hành không cho phép gán thuộc tính AllowDrop của PictureBox trên trình thiết kế, do đó phải gán trong code.

Project: **Su Dung Drag and Drop**

frmMain.vb

```
Option Strict On
Public Class frmMain
  Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

  ' Hằng số sử dụng để phát hiện có nhấn phím Ctrl trong quá trình kéo
  Const CtrlMask As Byte = 8
  Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles MyBase.Load
    ' Không có cách nào để quy định thuộc tính AllowDrop cho
    ' PictureBox trong trình thiết kế, do đó phải viết code.
   picLeft.AllowDrop = True
   picRight.AllowDrop = True
  End Sub

  ' Xử lý sự kiện MouseDown cho TextBox bên trái.
  Private Sub TextBoxLeft_MouseDown(ByVal sender As System.Object, ByVal e As
  System.Windows.Forms.MouseEventArgs) Handles txtLeft.MouseDown

   If e.Button = MouseButtons.Left Then
     txtLeft.SelectAll()
     ' Gọi hành động drag and drop
     txtLeft.DoDragDrop(txtLeft.SelectedText, DragDropEffects.Move Or
  DragDropEffects.Copy)
   End If
  End Sub
```

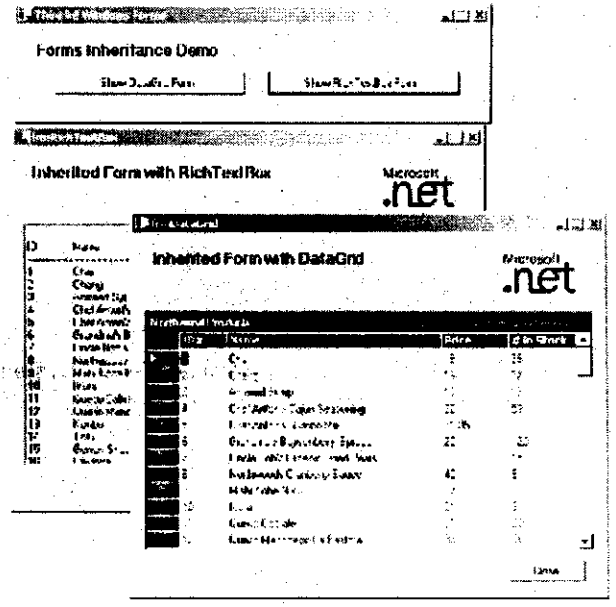
```
' Xử lý sự kiện DragDrop cho TextBox ở dưới phía bên phải. Sự kiện
' này thực hiện khi thả nút chuột, kết thúc thao tác drag and drop
Private Sub TextBoxLowerRight_DragDrop(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.DragEventArgs) Handles txtLowerRight.DragDrop
 txtLowerRight.Text = e.Data.GetData(DataFormats.Text).ToString

 ' Nếu không nhấn phím Ctrl, xóa nội dung nguồn để thực hiện di chuyển.
 If (e.KeyState And CtrlMask) <> CtrlMask Then
   txtLeft.Text = ""
 End If
End Sub

' Xử lý sự kiện DragEnter cho TextBox ở dưới, bên phải. Sự kiện này
' được gọi khi có một đối tượng được kéo vào trong khung của control
Private Sub TextBoxLowerRight_DragEnter(ByVal sender As System.Object, ByVal
e As System.Windows.Forms.DragEventArgs) Handles txtLowerRight.DragEnter

 ' Kiểm tra nội dung kéo vào là hợp lệ. Nếu không thì không cho thả
 If (e.Data.GetDataPresent(DataFormats.Text)) Then
    ' Nếu phím Ctrl được nhấn trong quá trình kéo thì thực hiện Copy.
    ' Nếu không, thực hiện Move.
   If (e.KeyState And CtrlMask) = CtrlMask Then
     e.Effect = DragDropEffects.Copy
   Else
     e.Effect = DragDropEffects.Move
   End If
 Else
   e.Effect = DragDropEffects.None
 End If
End Sub

' Xử lý sự kiện MouseDown cho các control PictureBox.
Private Sub PictureBox_MouseDown(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles picLeft.MouseDown,
picRight.MouseDown
 If e.Button = MouseButtons.Left Then
   Dim pic As PictureBox = CType(sender, PictureBox)
   'Gọi thao tác drag and drop
   If Not pic.Image Is Nothing Then
     pic.DoDragDrop(pic.Image, DragDropEffects.Move Or DragDropEffects.Copy)
```

```
    End If
  End If
End Sub

' Xử lý sự kiện DragEnter cho các PictureBox.
Private Sub PictureBox_DragEnter(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DragEventArgs) Handles picLeft.DragEnter,
picRight.DragEnter
  ' Kiểm tra tính hợp lệ của nội dung kéo vào
  If (e.Data.GetDataPresent(DataFormats.Bitmap)) Then
    ' Kiểm tra có nhấn Ctrl
    If (e.KeyState And CtrlMask) = CtrlMask Then
      e.Effect = DragDropEffects.Copy
    Else
      e.Effect = DragDropEffects.Move
    End If
  Else
    e.Effect = DragDropEffects.None
  End If
End Sub

' Xử lý sự kiện DragDrop cho cả hai PictureBox. Một thủ tục có thể
' dùng cho hai PictureBox bằng cách ép kiểu sender và kiểm tra
' thuộc tính Name để xác định control nào cần loại bỏ hình ảnh.
Private Sub PictureBox_DragDrop(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DragEventArgs) Handles picLeft.DragDrop,
picRight.DragDrop
  Dim pic As PictureBox = CType(sender, PictureBox)
  pic.Image = CType(e.Data.GetData(DataFormats.Bitmap), Bitmap)
  ' Làm cho hình ảnh trong PictureBox khác được loại bỏ nếu không có nhấn Ctrl.
  If (e.KeyState And CtrlMask) <> CtrlMask Then
    If pic.Name = "picLeft" Then
      picRight.Image = Nothing
    Else
      picLeft.Image = Nothing
    End If
  End If

End Sub
```

```
' Xử lý sự kiện DragDrop cho cả hai control TreeView
Private Sub TreeView_DragDrop(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DragEventArgs) Handles tvwLeft.DragDrop,
tvwRight.DragDrop
 ' Khởi tạo biến dùng để chứa node được kéo
 Dim OriginationNode As TreeNode =
CType(e.Data.GetData("System.Windows.Forms.TreeNode"), TreeNode)

 ' Gọi GetDataPresent khác đi một chút đối với TreeView vì TreeNode
  ' không phải là một thành phần của class DataFormats. Nghĩa là nó
  ' không phải là một kiểu xác định trước. Vì vậy cần dùng một overload khác
 If e.Data.GetDataPresent("System.Windows.Forms.TreeNode", False) Then
   Dim pt As Point
   Dim DestinationNode As TreeNode

   ' Sử dụng PointToClient để tính vị trí của chuột trên TreeView đích
   pt = CType(sender, TreeView).PointToClient(New Point(e.X, e.Y))
   ' Sử dụng Point này để lấy node gần nhất trên TreeView đích
   DestinationNode = CType(sender, TreeView).GetNodeAt(pt)

   ' Lệnh If bảo đảm là người sử dụng không mất các node nếu thả
    ' chuột không đúng vị trí hợp lệ
 If Not DestinationNode.TreeView Is OriginationNode.TreeView Then
     DestinationNode.Nodes.Add(CType(OriginationNode.Clone, TreeNode))
     ' Mở node cha khi thêm node mới để thấy rõ kết quả của thao tác thả chuột.
     DestinationNode.Expand()
     ' Nếu không có nhấn phím Ctrl, loại bỏ node gốc
     If (e.KeyState And CtrlMask) <> CtrlMask Then
       OriginationNode.Remove()
     End If
   End If
 End If
End Sub

' Xử lý sự kiện DragEnter cho cả hai control TreeView.
Private Sub TreeView_DragEnter(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DragEventArgs) Handles tvwRight.DragEnter,
tvwLeft.DragEnter
 ' Kiểm tra tính hợp lệ của nội dung được kéo
 If (e.Data.GetDataPresent("System.Windows.Forms.TreeNode")) Then
```

```
    ' Kiểm tra có nhấn Ctrl
    If (e.KeyState And CtrlMask) = CtrlMask Then
      e.Effect = DragDropEffects.Copy
    Else
      e.Effect = DragDropEffects.Move
    End If
  Else
    e.Effect = DragDropEffects.None
  End If
 End Sub

 ' Xử lý sự kiện ItemDrag cho cả hai control TreeView
 Private Sub TreeView_ItemDrag(ByVal sender As System.Object, ByVal e As
 System.Windows.Forms.ItemDragEventArgs) Handles tvwRight.ItemDrag,
 tvwLeft.ItemDrag
  If e.Button = MouseButtons.Left Then
    ' Gọi thao tác drag and drop
    DoDragDrop(e.Item, DragDropEffects.Move Or DragDropEffects.Copy)
  End If
 End Sub
End Class
```

17. Thừa kế Windows Forms



Ví dụ này cho biết cách sử dụng thừa kế Windows Forms. Một Form thực sự chỉ là một class, giống với tất cả các class khác trong .NET, ngoại trừ nó có khả năng hiển thị trực quan cho người sử dụng. Để sử dụng kiểu thừa kế này chỉ đơn giản tạo một class có nguồn gốc từ một class tạo ra form System.Windows.Forms.Form.

Các chức năng chính:

Chương trình này sử dụng frmMain để gọi hai form khác được phát sinh từ **frmBase.vb**: frmDataGrid và frmRichTextBox. Các class này thừa kế từ class frmBase bao gồm một Label, một Button, và các thuộc tính ảnh hưởng đến cách trình bày của form, và một hàm trả về một DataSet. Form được phát sinh sẽ thêm vào vùng ở giữa một DataGrid hoặc một TextBox, hiển thị cùng một DataSet.

Hai Label khác được sử dụng, với thuộc tính Text là một chuỗi trống, và được đặt trên frmBase. Một được khai báo là Protected và cái còn lại là Private. (Các control được đặt vào form bằng Visual Studio Designer được khai báo mặc nhiên là Friend). Trên các form được thừa kế, thử truy xuất đến các thành phần của từng Label. Chú ý là chỉ có Label khai báo là Protected mới truy xuất được.

Yêu cầu: Sử dụng database Northwind trong SQL Server hoặc MSDE.

Để xem được các form phát sinh từ frmBase trong trình thiết kế, cần phải biên dịch project. Nếu mở form thừa kế trước khi biên dịch thì đóng nó, biên dịch, và mở lại.

Project: **Thua ke Windows Forms**

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Code phát sinh form được bỏ qua

    Private Sub btnShowDataGridForm_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnShowDataGridForm.Click
        Dim frmDG As New frmDataGrid()
        frmDG.Show()
    End Sub
```

```
Private Sub btnShowRichTextBoxForm_Click(ByVal sender As System.Object, ByVal
  e As System.EventArgs) Handles btnShowRichTextBoxForm.Click
   Dim frmRTB As New frmRichTextBox()
   frmRTB.Show()
  End Sub
End Class
```

frmBase.vb

```
Option Strict On
Imports System.Data.SqlClient
Public Class frmBase
  Inherits System.Windows.Forms.Form

  Protected Const SQL_CONNECTION_STRING As String = _
   "Server=localhost;" & _
   "DataBase=Northwind;" & _
   "Integrated Security=SSPI"
  Protected Const MSDE_CONNECTION_STRING As String = _
   "Server=(local)\NetSDK;" & _
   "DataBase=Northwind;" & _
   "Integrated Security=SSPI"

  Protected Const CONNECTION_ERROR_MSG As String = _
   "To run this sample, you must have SQL or MSDE."

  Protected DidPreviouslyConnect As Boolean = False
  Protected DidCreateTable As Boolean = False
  Protected connectionString As String = SQL_CONNECTION_STRING

' Code phát sinh form được bỏ qua

  Protected Function GetDataSource() As DataSet
   Static ConnectionString As String = SQL_CONNECTION_STRING
   Static DidPreviouslyConnect As Boolean = False

   Dim frmStatusMessage As New frmStatus()
   If Not DidPreviouslyConnect Then
     frmStatusMessage.Show("Connecting to SQL Server")
   End If
```

```
    Dim IsConnecting As Boolean = True
    While IsConnecting

      Try
        Dim cnNorthwind As New SqlConnection(ConnectionString)

        Dim da As New SqlDataAdapter( _
      "SELECT ProductID, ProductName, UnitPrice, UnitsInStock " & _
          "FROM products", cnNorthwind)

        Dim dsProducts As New DataSet()

        da.Fill(dsProducts, "Products")

        IsConnecting = False
        DidPreviouslyConnect = True
        frmStatusMessage.Close()

        Return dsProducts

      Catch exc As Exception
        If ConnectionString = SQL_CONNECTION_STRING Then
          ConnectionString = MSDE_CONNECTION_STRING
          frmStatusMessage.Show("Connecting to MSDE")
        Else
          frmStatusMessage.Close()
          MessageBox.Show("To run this sample, you must have SQL or MSDE.",
            "Connection Problem", MessageBoxButtons.OK, MessageBoxIcon.Error)
          End
        End If
      End Try
    End While

    frmStatusMessage.Close()

  End Function
End Class
```

frmDataGrid.vb

```
Option Strict On
Public Class frmDataGrid
  Inherits frmBase

' Code phát sinh form được bỏ qua

  Private Sub btnClose_Click(ByVal sender As Object, ByVal e As
  System.EventArgs) Handles btnClose.Click
   Me.Close()
  End Sub

  Private Sub frmDataGrid_Load(ByVal sender As Object, ByVal e As
  System.EventArgs) Handles MyBase.Load

   With dgProducts
     .CaptionText = "Northwind Products"
     .DataSource = GetDataSource().Tables(0)
   End With

   ' Sử dụng đối tượng TableStyle để áp dụng định dạng cho DataGrid.
   Dim dgTableStyle1 As New DataGridTableStyle()
   With dgTableStyle1
     .AlternatingBackColor = Color.Lavender
     .BackColor = Color.WhiteSmoke
     .ForeColor = Color.MidnightBlue
     .GridLineColor = Color.Gainsboro
     .GridLineStyle = System.Windows.Forms.DataGridLineStyle.None
     .HeaderBackColor = Color.MidnightBlue
     .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
     .HeaderForeColor = Color.WhiteSmoke
     .LinkColor = Color.Teal
     .MappingName = "Products"
     .SelectionBackColor = Color.CadetBlue
     .SelectionForeColor = Color.WhiteSmoke
   End With

   ' Sử dụng đối tượng ColumnStyle để định dạng cho mỗi cột
   Dim grdColStyle1 As New DataGridTextBoxColumn()
```

```
    With grdColStyle1
      .HeaderText = "ID#"
      .MappingName = "ProductID"
      .Width = 50
    End With

    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
      .HeaderText = "Name"
      .MappingName = "ProductName"
      .Width = 225
    End With

    Dim grdColStyle3 As New DataGridTextBoxColumn()
    With grdColStyle3
      .HeaderText = "Price"
      .MappingName = "UnitPrice"
      .Width = 70
    End With

    Dim grdColStyle4 As New DataGridTextBoxColumn()
    With grdColStyle4
      .HeaderText = "# in Stock"
      .MappingName = "UnitsInStock"
      .Width = 70
    End With

    dgTableStyle1.GridColumnStyles.AddRange _
      (New DataGridColumnStyle() _
      {grdColStyle1, grdColStyle2, grdColStyle3, grdColStyle4})
    dgProducts.TableStyles.Add(dgTableStyle1)

    ' Gán tựa đề vào Label thừa kế
    lblTitle.Text = "Inherited Form with DataGrid"

  End Sub

End Class
```

frmRichTextBox.vb

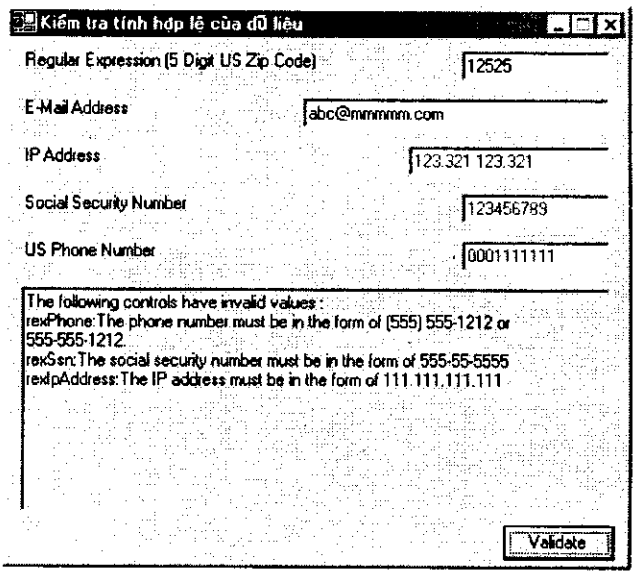
```
Option Strict On
Imports System.Text

Public Class frmRichTextBox
  Inherits frmBase

' Code phát sinh form được bỏ qua

  Private Sub btnClose_Click(ByVal sender As Object, ByVal e As
  System.EventArgs) Handles btnClose.Click
   Me.Close()
  End Sub

  Private Sub frmRichTextBox_Load(ByVal sender As Object, ByVal e As
  System.EventArgs) Handles MyBase.Load

   ' Lấy DataSet
   Dim ds As DataSet = GetDataSource()

   ' Thay vì sử dụng ghép chuỗi thông thường, sử dụng class
    ' StringBuilder để tối ưu việc ghép chuỗi.
   Dim sb As New StringBuilder()

   ' Dòng đầu làm tựa đề các cột
   sb.Append("=========================================")
   sb.Append(vbCrLf)
   sb.Append("ID")
   sb.Append(vbTab)
   sb.Append("Name")
   sb.Append(vbCrLf)
   sb.Append("=========================================")
   sb.Append(vbCrLf)

   ' Duyệt qua các dòng trong DataSet và nối chúng bằng StringBuilder
   Dim dr As DataRow
   For Each dr In ds.Tables(0).Rows
     sb.Append(dr("ProductID"))
     sb.Append(vbTab)
     sb.Append(dr("ProductName"))
```

```
        sb.Append(vbCrLf)
    Next
    ' Chuyển nội dung trong StringBuider sang Text để hiển thị trong RichTextBox
    rtbProducts.Text = sb.ToString
    lblTitle.Text = "Inherited Form with RichTextBox"
  End Sub
End Class
```

18. Kiểm tra tính hợp lệ của các TextBox (Validating TextBox)

Chương trình này biểu diễn cách tạo một tập các TextBox với các mục đích cụ thể chỉ để nhập các kiểu dữ liệu đặc biệt. Ví dụ các TextBox để nhập các địa chỉ email, các địa chỉ IP, số điện thoại,.... Nếu người sử dụng nhập nội dung vào TextBox nhưng không đúng với định dạng dữ liệu thì nội dung sẽ chuyển sang màu đỏ.

Mỗi TextBox đều thừa kế một class cơ sở RegExTextBox. Class này kiểm tra tính hợp lệ của nội dung theo một biểu thức.



Các chức năng chính:

- Thừa kế được sử dụng để thay đổi cách hoạt động của một control đã có (TextBox).
- Khi người sử dụng nhấn vào nút "Validate", chương trình duyệt qua các control để tìm tất cả control RegExTextBox và ép kiểu nó về dạng cơ sở để gọi phương thức IsValid.

Project: **Validating TextBoxes**

RegExTextBox.vb

```
Option Strict On
Imports System.Text.RegularExpressions

Public Class RegExTextBox
  Inherits System.Windows.Forms.TextBox

  ' Chuỗi đại diện cho RegEx sẽ được sử dụng để kiểm tra nội dung
  ' bên trong TextBox.
  Protected validationPattern As String
  ' Chuỗi thông báo khi nội dung không đúng theo mẫu.
  Protected mErrorMessage As String
  ' Đối tượng RegEx được sử dụng để thực hiện kiểm tra
  Protected mValidationExpression As Regex
  ' Màu mặc nhiên sẽ sử dụng khi nội dung không đúng
  Protected mErrorColor As Color = Color.Red

  ' Cho phép người lập trình quy định chuỗi thông báo lỗi vào lúc thiết
  ' kế hoặc lúc đang chạy.
  Public Property ErrorMessage() As String
   Get
     Return mErrorMessage
   End Get
   Set(ByVal Value As String)
     mErrorMessage = Value
   End Set
  End Property

' Nếu nội dung không phù hợp với RegEx, nó sẽ chuyển sang màu này
  Public Property ErrorColor() As Color
   Get
     Return mErrorColor
   End Get
   Set(ByVal Value As Color)
     mErrorColor = Value
   End Set
  End Property
```

```
    ' Cho phép người lập trình xác định được nội dung trong TextBox có
    ' đúng hay không.
    Public ReadOnly Property IsValid() As Boolean
      Get
        If Not mValidationExpression Is Nothing Then
          Return mValidationExpression.IsMatch(Me.Text)
        Else
          Return True
        End If
      End Get
    End Property

    ' Cho phép chỉ định regular expression (là một chuỗi) sẽ được sử
    ' dụng để kiểm tra tính hợp lệ của văn bản bên trong TextBox.
    Public Property ValidationExpression() As String
      Get
        Return validationPattern
      End Get
      Set(ByVal Value As String)
        mValidationExpression = New Regex(Value)
        validationPattern = Value
      End Set
    End Property

    ' Nếu dữ liệu không phù hợp với RegEx, thì đổi màu của văn bản sang ErrorColor
    Protected Overrides Sub OnValidated(ByVal e As System.EventArgs)
      If Not Me.IsValid Then
        Me.ForeColor = mErrorColor
      Else
        Me.ForeColor = Me.DefaultForeColor
      End If

      ' Bất cứ khi nào thừa kế một control, và override một trong các thủ
      ' tục On..., cần phải gọi phương thức On... của lớp cơ sở.
      MyBase.OnValidated(e)
    End Sub
End Class
```

EMailTextBox.vb

```
Option Strict On
Public Class EMailTextBox
    ' Các textbox thừa kế RegExTextBox để kiểm tra nội dung bên trong
    ' nó có thỏa một regular expression.
    Inherits RegExTextBox

    Public Sub New()
     MyBase.New()

     InitializeComponent()

     ' Quy định giá trị cho ValidationExpression để control này kiểm tra
      ' tính hợp lệ của nội dung bên trong TextBox phải là địa chỉ email.
     Me.ValidationExpression = "^([a-zA-Z0-9_\-])([a-zA-Z0-9_\-\.]*)@(\[((25[0-5]|2[0-
    4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])\.){3}|((([a-zA-Z0-9\-]+)\.)+))([a-zA-Z]{2,}|(25[0-
    5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])\])$"
     Me.ErrorMessage = "The e-mail address must be in the form of
    abc@microsoft.com"

    End Sub
' Code phát sinh được bỏ qua
End Class
```

IPAddressTextBox.vb

```
Option Strict On
Public Class IPAddressTextBox
    Inherits RegExTextBox
    Public Sub New()
     MyBase.New()
     InitializeComponent()
     ' Quy định giá trị mặc nhiên cho ValidationExpression để control
      ' này sẽ kiểm tra nội dung bên trong nó là một địa chỉ IP.
     Me.ValidationExpression = "^((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-
    9])\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])$"
     Me.ErrorMessage = "The IP address must be in the form of 111.111.111.111"
    End Sub
' Code phát sinh được bỏ qua
End Class
```

PhoneTextBox.vb

```
Option Strict On
Public Class PhoneTextBox
  Inherits RegExTextBox
  Public Sub New()
   MyBase.New()
   InitializeComponent()
    ' Quy định giá trị mặc nhiên cho ValidationExpression để control này
    ' kiểm tra nội dung bên trong là một số điện thoại
      Me.ValidationExpression = "^((\(\d{3}\) ?)|(\d{3}-))?\d{3}-\d{4}$"
   Me.ErrorMessage = "The phone number must be in the form of (555) 555-1212 or
  555-555-1212."
  End Sub
' code phát sinh được bỏ qua
End Class
```

SsnTextBox.vb

```
Option Strict On

Public Class SsnTextBox
  Inherits RegExTextBox
  Public Sub New()
   MyBase.New()
   InitializeComponent()

    ' Quy định giá trị mặc nhiên cho ValidationExpression
   Me.ValidationExpression = "^\d{3}-\d{2}-\d{4}$"
   Me.ErrorMessage = "The social security number must be in the form of 555-55-
  5555"
  End Sub
' Code phát sinh được bỏ qua
End Class
```

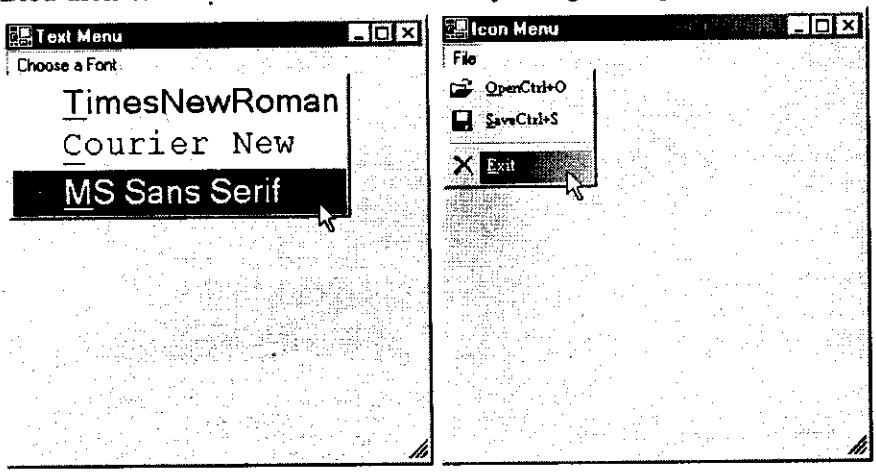
frmMain.vb

```
Option Strict On
Public Class frmMain
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
 ' Sử dụng để duyệt qua tất cả các control trên form
 Dim genericControl As Control

  ' Lưu thông báo sẽ hiển thị cho người sử dụng biết các control chứa
  ' nội dung không hợp lệ.
 Dim validationMessage As String

 ' Duyệt qua tất cả các control có trên form
 For Each genericControl In Controls
   ' Nếu control hiện hành là một RegExTextBox, hoặc thừa kế RegExTextBox
   If TypeOf genericControl Is RegExTextBox Then
     ' Ép kiểu nó thành kiểu RegExTextBox. Kiểu này cho phép sử
      ' dụng thuộc tính IsValid.
     Dim regExControl As RegExTextBox = CType(genericControl, RegExTextBox)

     ' Nếu nội dung bên trong control không hợp lệ, thì cho control
      ' này vào danh sách các control không hợp lệ.
     If Not regExControl.IsValid Then
       validationMessage &= regExControl.Name & ":" & _
        regExControl.ErrorMessage & vbCrLf
     End If
   End If
 Next

 ' Có control nào chứa nội dung không hợp lệ không?
 If validationMessage <> "" Then
   ' Xuất thông báo ra hộp văn bản.
   txtInvalidControls.Text = "The following controls have invalid values : " & vbCrLf &
validationMessage
 Else
   ' Nếu không có, thông báo mọi thứ hợp lệ
   txtInvalidControls.Text = "All controls contain valid input"
 End If

 End Sub
End Class
```

19. Tự vẽ Menu

Biểu diễn cách tạo các menu item theo ý riêng của người lập trình.



Các chức năng chính:

- Cách tạo menu chứa văn bản theo kiểu riêng.
- Cách tạo menu có màu nền biến thiên và biểu tượng bên cạnh.

Project: **Tu ve Menu**

IconMenuItem.vb

```
Imports System
Imports System.ComponentModel
Imports System.Drawing
Imports System.Drawing.Drawing2D
Imports System.Drawing.Text
Imports System.Reflection
Imports System.Resources
Imports System.Windows.Forms

Public Class IconMenuItem : Inherits MenuItem
    Private m_Icon As Icon
    Private m_Font As Font

    ' Mặc nhiên các giá trị này là giá trị của hệ thống. Điều này cho phép màu thích hợp
    ' được hiển thị khi người sử dụng thay đổi thông số màu sắc hiển thị.
    ' Các giá trị này có thể override bằng cách gọi bộ khởi tạo tương ứng cho class này
    Private m_Gradient_Color1 As Color = SystemColors.Highlight
    Private m_Gradient_Color2 As Color = SystemColors.Control
```

```
Public Sub New()
    MyClass.New("", Nothing, Nothing, System.Windows.Forms.Shortcut.None)
End Sub

Public Sub New(ByVal text As String, ByVal icon As Icon, ByVal onClick As
EventHandler, ByVal shortcut As Shortcut)
    MyBase.New(text, onClick, shortcut)
     ' Thuộc tính Owner Draw cho phép thay đổi menuitem bằng cách xử lý
     ' OnMeasureItem và OnDrawItem
    OwnerDraw = True
    m_Font = New Font("Times New Roman", 8)
    m_Icon = icon
End Sub

 ' Bộ khởi tạo này cho phép quy định màu nền biến thiên trên menuitem
 Public Sub New(ByVal text As String, ByVal GradientColor1 As
System.Drawing.Color, ByVal GradientColor2 As System.Drawing.Color, ByVal icon
As Icon, ByVal onClick As EventHandler, ByVal shortcut As Shortcut)
    MyBase.New(text, onClick, shortcut)
     ' Thuộc tính quan trọng để có thể tự vẽ menu
    OwnerDraw = True
    m_Font = New Font("Times New Roman", 8)
    m_Gradient_Color1 = GradientColor1
    m_Gradient_Color2 = GradientColor2
    m_Icon = icon
End Sub

Private Function GetRealText() As String
    Dim s As String = Text
     ' Nối thêm phần phím tắt nếu có
    If ShowShortcut And Shortcut <> Shortcut.None Then
        ' Để nhận được chuỗi chỉ giá trị của Shortcut, ép kiểu nó thành một giá
        ' trị Keys và sử dụng class KeysConverter
       Dim k As Keys = CType(Shortcut, Keys)
       s = s & Convert.ToChar(9) & _
                TypeDescriptor.GetConverter(GetType(Keys)).ConvertToString(k)
    End If
    Return s
End Function
```

```
Protected Overrides Sub OnDrawItem(ByVal e As DrawItemEventArgs)
    ' OnDrawItem thực hiện công việc thực sự vẽ menu sau khi hoàn tất các thông số
    MyBase.OnDrawItem(e)
    Dim br As Brush

    If Not m_Icon Is Nothing Then
      e.Graphics.DrawIcon(m_Icon, e.Bounds.Left + 2, e.Bounds.Top + 2)
    End If

    Dim rcBk As Rectangle = e.Bounds
    rcBk.X += 22

    ' Vẽ màu nền cho menuitem với một màu biến thiên,
    If CBool(e.State And DrawItemState.Selected) Then
    br = New LinearGradientBrush(rcBk, m_Gradient_Color1, m_Gradient_Color2, 0)
    Else
      br = SystemBrushes.Control
    End If

    ' Vẽ hình chữ nhật chính
    e.Graphics.FillRectangle(br, rcBk)

    ' Chừa chổ trống cho phím tắt
    Dim sf As StringFormat = New StringFormat()
    sf.HotkeyPrefix = HotkeyPrefix.Show

    ' Vẽ văn bản nội dung của menu
    br = New SolidBrush(e.ForeColor)
    e.Graphics.DrawString(GetRealText(), m_Font, br, _
                  e.Bounds.Left + 25, e.Bounds.Top + 2, sf)
End Sub

Protected Overrides Sub OnMeasureItem(ByVal e As MeasureItemEventArgs)
    ' Thủ tục sự kiện MeasureItem cùng với sự kiện OnDrawItem là hai sự kiện
    ' chính cần thiết cần quản lý để tạo các menu riêng.
    ' Đo chuỗi tạo thành menuitem và sử dụng nó để quy định kích thước
    ' của menuitem sẽ được vẽ.
```

```
    Dim sf As New StringFormat()
    sf.HotkeyPrefix = HotkeyPrefix.Show
    MyBase.OnMeasureItem(e)
    e.ItemHeight = 22
    e.ItemWidth = CInt(e.Graphics.MeasureString(GetRealText(), m_Font, 10000,
sf).Width) + 10

  End Sub
End Class
```

frmIconModMenu.vb

```
Public Class frmIconModMenu
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

  Private Sub frmIconModMenu_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    ' Các menuitem cơ bản với các biểu tượng bên cạnh

    mnuMain.MenuItems.Add("&File")
    Dim miFile As MenuItem = mnuMain.MenuItems(0)
    Dim handlerFile As EventHandler = New EventHandler( _
                                        AddressOf MenuItemFileClick)

    ' Tạo các MenuItem với các biểu tượng và màu nền chuẩn
    miFile.MenuItems.Add(New IconMenuItem("&Open", New Icon("..\open.ico"), _
      handlerFile, Shortcut.CtrlO))
    miFile.MenuItems.Add(New IconMenuItem("&Save", New Icon("..\save.ico"), _
      handlerFile, Shortcut.CtrlS))
    miFile.MenuItems.Add(New MenuItem("-"))
    miFile.MenuItems.Add(New IconMenuItem("&Exit", New Icon("..\exit.ico"), _
      handlerFile, Shortcut.None))

    ' Phần này tạo các menuitem với các màu nền tùy ý

  'mnuMain.MenuItems.Add("&File")
   'Dim miEdit As MenuItem = mnuMain.MenuItems(0)
```

```
    'Dim handlerFile As EventHandler = New EventHandler(AddressOf _
    '    MenuItemFileClick)
  'miEdit.MenuItems.Add(New IconMenuItem("&Open", Color.LightBlue, Color.Yellow, _
  '    New Icon("..\open.ico"), handlerFile, Shortcut.CtrlO))
  'miEdit.MenuItems.Add(New IconMenuItem("&Save", Color.LightBlue, Color.Yellow, _
  '    New Icon("..\save.ico"), handlerFile, Shortcut.CtrlS))
  'miEdit.MenuItems.Add(New MenuItem("-"))
  'miEdit.MenuItems.Add(New IconMenuItem("&Exit", Color.LightBlue, Color.Yellow, _
  '    New Icon("..\exit.ico"), handlerFile, Shortcut.None))
  End Sub

  Private Sub MenuItemFileClick(ByVal sender As Object, ByVal e As EventArgs)
    Select Case CType(sender, MenuItem).Index
      Case 0    ' Open
        MessageBox.Show("You selected open")
      Case 1    ' Save
        MessageBox.Show("You selected save")
      Case 3    ' Exit
        Close()
    End Select
  End Sub
End Class
```

frmTextModMenu.vb

```
Public Class frmTextModMenu
  Inherits System.Windows.Forms.Form
  Const pointSize As Integer = 18

  Private miFaceName As MenuItem
' Code phát sinh form được bỏ qua

  ' Ví dụ này vẽ một menu chứa ba menuitem với văn bản tùy biến. Các menuitem
  ' được vẽ với kích thước phù hợp với văn bản có trên menuitem.

  Private Sub frmTextModMenu_Load(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles MyBase.Load

    ' Tạo menu chính và thêm các menuitem vào menu
    mnuMain.MenuItems.Add("&Choose a Font")
```

```
    Dim menuItemsText() As String = {"&TimesNewRoman", "&Courier New", "&MS
Sans Serif"}
    Dim menuItems(menuItemsText.Length - 1) As MenuItem
    Dim menuItemCount As Integer = CInt(menuItems.Length)
    Dim evOnClick As New EventHandler(AddressOf MenuFaceNameOnClick)
    Dim evOnMeasure As New MeasureItemEventHandler(AddressOf
MenuFaceNameOnMeasureItem)
    Dim evDrawItem As New DrawItemEventHandler(AddressOf
MenuFaceNameOnDrawItem)

    Dim i As Integer = 0
    For i = 0 To menuItemCount - 1 Step 1
      menuItems(i) = New MenuItem(menuItemsText(i))
       ' Cho phép các sự kiện OnMeasureItem và OnDrawnItem
      menuItems(i).OwnerDraw = True
       ' Thêm các thủ tục xử lý các menuitem
      AddHandler menuItems(i).Click, evOnClick
      AddHandler menuItems(i).MeasureItem, evOnMeasure
      AddHandler menuItems(i).DrawItem, evDrawItem
    Next

    mnuMain.MenuItems(0).MenuItems.AddRange(menuItems)
  End Sub

  Private Sub MenuFaceNameOnClick(ByVal sender As Object, ByVal e As
EventArgs)
     ' Chỉ đơn giản cho người sử dụng biết đã chọn menuitem nào.
    miFaceName = CType(sender, MenuItem)
    miFaceName.Checked = True
    MessageBox.Show("Menu Clicked: " & miFaceName.Text.Substring(1))
  End Sub

  Private Sub MenuFaceNameOnMeasureItem(ByVal sender As Object, ByVal miea
As MeasureItemEventArgs)
     ' Sự kiện MeasureItem và OnDrawItem là hai sự kiện chính cần quản lý để
     ' tạo các menu riêng.
    Dim mi As MenuItem = CType(sender, MenuItem)
    Dim f As New Font(mi.Text.Substring(1), pointSize)
     ' Đo chuỗi để xác định kích kích thước của menuitem được vẽ
    Dim siF As SizeF = miea.Graphics.MeasureString(mi.Text, f)
```

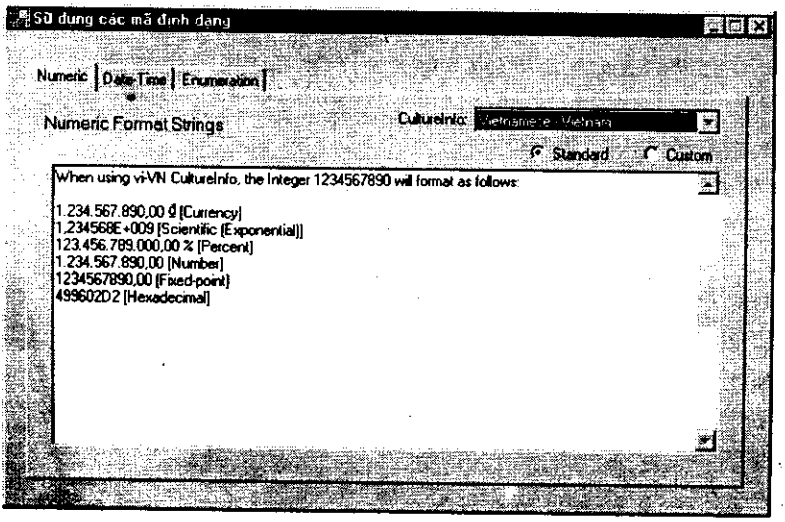
```
        miea.ItemWidth = CInt(Math.Ceiling(siF.Width))
        miea.ItemHeight = CInt(Math.Ceiling(siF.Height))
    End Sub

    Private Sub MenuFaceNameOnDrawItem(ByVal sender As Object, ByVal diea As
    DrawItemEventArgs)

        ' Sau khi đã thiết lập kích thước trong sự kiện OnMeasureItem, ta có thể
        ' tiến hành vẽ menu.
        ' Một đối tượng graphics được truyền vào từ sự kiện OnDrawItem để sử
        ' dụng vẽ menuitem.
        Dim mi As MenuItem = CType(sender, MenuItem)
        Dim grfx As Graphics = diea.Graphics
        Dim br As Brush
        Dim f As New Font(mi.Text.Substring(1), pointSize)

        ' Hiển thị phím tắt
        Dim strFrmt As New StringFormat()
        strFrmt.HotkeyPrefix = Drawing.Text.HotkeyPrefix.Show

        Dim rectCheck As Rectangle = diea.Bounds
        Dim d As Double = SystemInformation.MenuCheckSize.Width *
    rectCheck.Height / SystemInformation.MenuCheckSize.Width
        rectCheck.Width = CInt(d)

        Dim recText As Rectangle = diea.Bounds
        ' Chừa chỗ cho dấu check
        recText.X += rectCheck.Width

        diea.DrawBackground()
        ' Làm nổi bật menuitem khi người sử dụng di chuyển chuột qua nó
        If ((diea.State And DrawItemState.Selected) <> 0) Then
            br = SystemBrushes.HighlightText
        Else
            br = SystemBrushes.FromSystemColor(SystemColors.MenuText)
        End If
        ' vẽ chuỗi chừa chỗ cho phím tắt
        grfx.DrawString(mi.Text, f, br, recText.Left, recText.Top, strFrmt)
    End Sub
End Class
```

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form

'Code phát sinh form được bỏ qua

    Private Sub btnTextModMenu_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnTextModMenu.Click
     Dim frmText As New frmTextModMenu()
     frmText.ShowDialog()
    End Sub

    Private Sub btnIconModMenu_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnIconModMenu.Click
     Dim frmIcon As New frmIconModMenu()
     frmIcon.ShowDialog()
    End Sub
End Class
```

20. Sử dụng các mã định dạng để định dạng dữ liệu trong các chuỗi.

Chương trình này cho thấy cách hiển thị các giá trị số, ngày tháng và bảng liệt kê như là các chuỗi bằng cách sử dụng các code định dạng chuẩn và các code định dạng mở rộng.



Các chức năng chính:

- Hiển thị các giá trị số được định dạng như là các chuỗi sử dụng định dạng chuẩn và định dạng tùy biến. Cách thay đổi định dạng dựa trên CultureInfo được chọn.
- Hiển thị các giá trị ngày-giờ được định dạng như là các chuỗi sử dụng định dạng chuẩn và định dạng tùy biến. Cách thay đổi định dạng dựa trên CultureInfo được chọn.
- Hiển thị các giá trị liệt kê được định dạng như là các chuỗi sử dụng mã định dạng chuẩn.

Project: **Su dung cac ma dinh dang**

frmMain.vb

```
Option Strict On
Imports System.Threading
Imports System.Globalization
Imports System.Text

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    ' Class Culture là một kiểu tùy biến được sử dụng để kết dữ liệu vào một
    ' ComboBox. Chú ý việc sử dụng các thuộc tính public thay cho các biến
    ' đơn thuần. Ta có thể sử dụng cấu trúc sau:
    '
    '  Public Class Culture
    '      Public ID As String
    '      Public Description As String
    '
    '      Sub New(ByVal strID As String, ByVal strDesc As String)
    '          ID = strID
    '          Description = strDesc
    '      End Sub
    '  End Class
    '
```

```
' Tuy nhiên, cấu trúc này phát sinh một ngoại lệ khi chạy InvalidArgumentException
' cho biết là nó không thể kết vào DisplayMember mới. Thuộc tính không phải
' là ReadOnly.

Public Class Culture
 Private _ID As String
 Private _desc As String

 Sub New(ByVal strDesc As String, ByVal strID As String)
   _ID = strID
   _desc = strDesc
 End Sub
 Public ReadOnly Property ID() As String
   Get
     Return _ID
   End Get
 End Property

 Public ReadOnly Property Description() As String
   Get
     Return _desc
   End Get
 End Property
End Class

Private crlf As String = ControlChars.CrLf
Private FormHasLoaded As Boolean = False
Private strCultureValue As String
Private tab As String = ControlChars.Tab

' Gọi phương thức hiển thị các ví dụ định dạng DateTime dựa trên một
' CultureInfo do người sử dụng chọn.
Private Sub cboCultureInfoDateTime_SelectedValueChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
cboCultureInfoDateTime.SelectedIndexChanged
 ' Thủ tục này chỉ làm việc chỉ khi Form đã được load
 If FormHasLoaded Then
   LoadDateTimeFormats()
 End If
End Sub
```

```
' Gọi phương thức hiển thị các ví dụ định dạng Numeric dựa trên một
' CultureInfo được chọn.
Private Sub cboCultureInfoNumeric_SelectedValueChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
cboCultureInfoNumeric.SelectedIndexChanged
 If FormHasLoaded Then
   LoadNumericFormats()
 End If
End Sub

' Load các control ComboBox từ một ArrayList và gọi các phương thức hiển thị các
' ví dụ định dạng.
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

' Kết các ComboBox vào một ArrayList chứa các đối tượng.
Dim arlCultureInfo As New ArrayList()
With arlCultureInfo
   .Add(New Culture("Vietnamese - Vietnam", "vi-VN"))
   .Add(New Culture("English - United States", "en-US"))
   .Add(New Culture("English - United Kingdom", "en-GB"))
   .Add(New Culture("English - New Zealand", "en-NZ"))
   .Add(New Culture("German - Germany", "de-DE"))
   .Add(New Culture("Spanish - Spain", "es-ES"))
   .Add(New Culture("French - France", "fr-FR"))
   .Add(New Culture("Portuguese - Brazil", "pt-BR"))
   .Add(New Culture("Malay - Malaysia", "ms-MY"))
   .Add(New Culture("Afrikaans - South Africa", "af-ZA"))
End With

cboCultureInfoDateTime.DataSource = arlCultureInfo
cboCultureInfoDateTime.DisplayMember = "Description"
cboCultureInfoDateTime.ValueMember = "ID"

cboCultureInfoNumeric.DataSource = arlCultureInfo
cboCultureInfoNumeric.DisplayMember = "Description"
cboCultureInfoNumeric.ValueMember = "ID"

LoadEnumFormats()
```

```
    LoadDateTimeFormats()
    LoadNumericFormats()

    FormHasLoaded = True
  End Sub

  ' Gọi phương thức hiển thị các ví dụ định dạng dựa trên người sử dụng chọn
  ' định dạng chuẩn ("standard") hay tùy biến ("custom").
  Private Sub RadioButtons_CheckedChanged(ByVal sender As Object, ByVal e As
  System.EventArgs) Handles optStandardNumeric.CheckedChanged,
  optCustomNumeric.CheckedChanged, optCustomDateTime.CheckedChanged,
  optStandardDateTime.CheckedChanged

    If FormHasLoaded Then
      Dim opt As RadioButton = CType(sender, RadioButton)
      Select Case opt.Name
        Case "optStandardNumeric", "optCustomNumeric"
          LoadNumericFormats()
        Case "optStandardDateTime", "optCustomDateTime"
          LoadDateTimeFormats()
        Case Else
          LoadEnumFormats()
      End Select
    End If
  End Sub

  ' Hiển thị các ví dụ định dạng DataTime
  Private Sub LoadDateTimeFormats()
    Dim dtmNow As DateTime = Now
    Dim sb As New StringBuilder()
    strCultureValue = cboCultureInfoDateTime.SelectedValue.ToString
    Thread.CurrentThread.CurrentCulture = New CultureInfo(strCultureValue)

    sb.Append("When using " & strCultureValue & " CultureInfo, today's date and time
  will format as follows:")
    sb.Append(crlf)
    sb.Append(crlf)
    If optStandardDateTime.Checked Then
      With sb
        .Append(dtmNow.ToString("d"))
```

```
        .Append(" [Short date pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("D"))
        .Append(" [Long date pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("t"))
        .Append(" [Short time pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("T"))
        .Append(" [Long time pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("F"))
        .Append(" [Full date/time pattern (long)]")
        .Append(crlf)
        .Append(dtmNow.ToString("f"))
        .Append(" [Full date/time pattern (short)]")
        .Append(crlf)
        .Append(dtmNow.ToString("G"))
        .Append(" [General date/time pattern (long)]")
        .Append(crlf)
        .Append(dtmNow.ToString("g"))
        .Append(" [General date/time pattern (short)]")
        .Append(crlf)
        .Append(dtmNow.ToString("M"))
        .Append(" [Month day pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("R"))
        .Append(" [RFC1123 pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("s"))
        .Append(" [Sortable date/time pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("u"))
        .Append(" [Universable sortable date/time pattern]")
        .Append(crlf)
        .Append(dtmNow.ToString("y"))
        .Append(" [Year month pattern]")
        .Append(crlf)
        .Append(crlf)
    End With
```

```
ElseIf optCustomDateTime.Checked Then
  With sb
    .Append(dtmNow.ToString("d, M"))
    .Append(" [d, M]")
    .Append(crlf)
    .Append(dtmNow.ToString("d MMMM"))
    .Append(" [d MMMM]")
    .Append(crlf)
    .Append(dtmNow.ToString("dddd MMMM yy gg"))
    .Append(" [dddd MMMM yy gg]")
    .Append(crlf)
    .Append(dtmNow.ToString("h , m: s"))
    .Append(" [h , m: s]")
    .Append(crlf)
    .Append(dtmNow.ToString("hh,mm:ss"))
    .Append(" [hh,mm:ss]")
    .Append(crlf)
    .Append(dtmNow.ToString("HH-mm-ss-tt"))
    .Append(" [HH-mm-ss-tt]")
    .Append(crlf)
    .Append(dtmNow.ToString("hh:mm, G\MT z"))
    .Append(" [hh:mm, G\MT z]")
    .Append(crlf)
    .Append(dtmNow.ToString("hh:mm, G\MT zzz"))
    .Append(" [hh:mm, G\MT zzz]")
  End With
End If

txtDateTime.Text = sb.ToString
End Sub

' Hiển thị các ví dụ định dạng Enum
Private Sub LoadEnumFormats()
Dim day As DayOfWeek = DayOfWeek.Friday
Dim sb As New StringBuilder()
Thread.CurrentThread.CurrentCulture = New CultureInfo("en-US")
With sb
  .Append("When using any CultureInfo, the system enumeration
DayOfWeek.Friday will format as follows:")
```

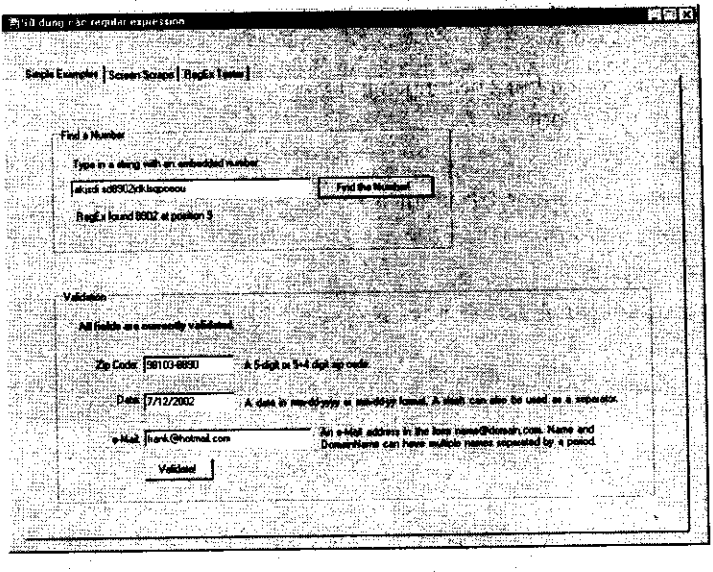
```
        .Append(crlf)
        .Append(crlf)
        .Append(day.ToString("G"))
        .Append(" [G or g]")
        .Append(crlf)
        .Append(day.ToString("F"))
        .Append(" [F or f]")
        .Append(crlf)
        .Append(day.ToString("D"))
        .Append(" [D or d]")
        .Append(crlf)
        .Append(day.ToString("X"))
        .Append(" [X or x]")
    End With

    txtEnum.Text = sb.ToString
End Sub

' Hiển thị các ví dụ định dạng Numeric
Private Sub LoadNumericFormats()
    Dim intNumber As Int32 = 1234567890
    Dim sb As New StringBuilder()
    strCultureValue = cboCultureInfoNumeric.SelectedValue.ToString
    Thread.CurrentThread.CurrentCulture = New CultureInfo(strCultureValue)
    sb.Append("When using " & strCultureValue & " CultureInfo, the Integer
1234567890 will format as follows:")
    sb.Append(crlf)
    sb.Append(crlf)

    If optStandardNumeric.Checked Then
        With sb
            .Append(intNumber.ToString("C"))
            .Append(" [Currency]")
            .Append(crlf)
            .Append(intNumber.ToString("E"))
            .Append(" [Scientific (Exponential)]")
            .Append(crlf)
            .Append(intNumber.ToString("P"))
            .Append(" [Percent]")
            .Append(crlf)
```

```
        .Append(intNumber.ToString("N"))
        .Append(" [Number]")
        .Append(crlf)
        .Append(intNumber.ToString("F"))
        .Append(" [Fixed-point]")
        .Append(crlf)
        .Append(intNumber.ToString("X"))
        .Append(" [Hexadecimal]")
        .Append(crlf)
        .Append(crlf)
      End With

    ElseIf optCustomNumeric.Checked Then
      With sb
        .Append(intNumber.ToString("#####"))
        .Append(" [#####]")
        .Append(crlf)
        .Append(intNumber.ToString("00000"))
        .Append(" [00000]")
        .Append(crlf)
        .Append(intNumber.ToString("(###) ### - ####"))
        .Append(" [(###) ### - ####]")
        .Append(crlf)
        .Append(intNumber.ToString("#.##"))
        .Append(" [#.##]")
        .Append(crlf)
        .Append(intNumber.ToString("00.00"))
        .Append(" [00.00]")
        .Append(crlf)
        .Append(intNumber.ToString("#,#"))
        .Append(" [#,#]")
        .Append(crlf)
        .Append(intNumber.ToString("#,,"))
        .Append(" [#,,]")
        .Append(crlf)
        .Append(intNumber.ToString("#.##"))
        .Append(" [#.##]")
        .Append(crlf)
        .Append(intNumber.ToString("#,,,"))
        .Append(" [#,,,]")
```

```
        .Append(crlf)
        .Append(intNumber.ToString("#,##0,,"))
        .Append(" [#,##0,,]")
        .Append(crlf)
        .Append(intNumber.ToString("#0.##%"))
        .Append(" [#0.##%]")
        .Append(crlf)
        .Append(intNumber.ToString("0.###E+000"))
        .Append(" [0.###E+000]")
        .Append(crlf)
        .Append(intNumber.ToString("##;(##)"))
        .Append(" [##;(##)]")
        .Append(crlf)
      End With
    End If
    txtNumeric.Text = sb.ToString
  End Sub
End Class
```

21. Sử dụng Regular Expression

Chương trình này cho biết cách sử dụng các Regular Expression. Nếu chưa quen với regular expression, đây là một cách chuẩn và đã được xây dựng tốt để phân tích các file văn bản cũng như tìm và có thể thay thế các chuỗi con theo các mẫu.



Các chức năng chính:

- Tab "Simple Samples" là một giới thiệu về sử dụng các regular expression. Ta sẽ biết cách tìm một số xuất hiện đầu tiên trong chuỗi, cũng như các ví dụ phức tạp hơn.
- Tab "Screen Scrape" cho thấy cách sử dụng regular expression cao cấp hơn. Có thể nhập vào một địa chỉ Web và xem được các liên kết hoặc các hình ảnh có trong trang đó (cần kết nối vào Internet). Các thuộc tính của HTML tag được hiển thị trong một ListView. Có thể nhấp đúp chuột vào một hình ảnh để xem nó, hoặc nhấp đúp vào một liên kết để xem nội dung trang đó với Internet Explorer.
- Tab "RegEx Tester" cung cấp một cách thuận tiện để thử nghiệm với các regular expression. Nó cũng biểu diễn cách tìm kiếm, thay thế, và tách chuỗi.

Project: **Su dung Regular Expressions**

ImageAttributes.vb

```
Option Strict On

Public Class ImageAttributes
    Public Src As String
    Public Alt As String
    Public Height As String
    Public Width As String

    Sub New(ByVal strSrc As String, ByVal strAlt As String, ByVal strHeight As String,
    ByVal strWidth As String)
        Src = strSrc
        Width = strWidth
        Height = strHeight
        Alt = strAlt
    End Sub
End Class
```

frmImageViewer.vb

```
Option Strict On
Public Class frmImageViewer
    Inherits System.Windows.Forms.Form

    ' Code phát sinh form được bỏ qua
```

```
    ' Phương thức này overload phương thức Show mặc nhiên để có thể truyền vào
    ' một tham số để hiển thị hình ảnh dựa trên một Stream được trả về từ
    ' GetHttpStream() trong frmMain.
    Public Overloads Sub Show(ByVal strImgSrc As String)
        Dim frmMain As New frmMain()
        With picImage
            Try
                ' Gán thuộc tính Image của PictureBox bằng Image được phát sinh bởi
                ' Stream trả lời của GetHttpStream
                .Image = Image.FromStream(frmMain.GetHttpStream(strImgSrc))
            Catch exp As Exception
                MsgBox(exp.ToString, MsgBoxStyle.Critical, Me.Text)
            End Try

            .SizeMode = PictureBoxSizeMode.CenterImage
        End With

        lblImageFilename.Text = strImgSrc
        Me.Show()
        Application.DoEvents()
    End Sub

    ' Nút đóng form
    Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnClose.Click
        Me.Hide()
    End Sub
End Class
```

frmMain.vb

```
Option Strict On
Imports System.Text.RegularExpressions
Imports System.IO
Imports System.Net

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua
```

```
Private htImages As Hashtable
Protected frmImageViewer As frmImageViewer
Private frmStatus As frmStatus
Public strDomain As String
Public strUrlEntered As String
Private wreqScrape As HttpWebRequest
Private wresScrape As HttpWebResponse

' Xử lý nút Find trên tab "RegEx Tester". Thủ tục này tìm các phần phù hợp trong
' văn bản nguồn sau đó hiển thị nhóm và giá trị tìm được trong TextBox kết quả.
Private Sub btnFind_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFind.Click
 Dim re As Regex
 Dim reOpt As RegexOptions = GetRegexOptions()

 Try
   re = New Regex(txtRegEx.Text, reOpt)
 Catch exp As Exception
   MsgBox("An error was encountered when attempting to parse the " & _
     "source text. This is often caused by an invalid regex pattern." & _
     "  Check your expression and try again. Exp.Msg=" & exp.Message, _
     MsgBoxStyle.Critical, Me.Text)
   Exit Sub
 End Try

 ' Lấy tập hợp các kết quả phù hợp với mẫu
 Dim mc As MatchCollection = re.Matches(txtSource.Text)

 ' Hiển thị số phần tìm thấy và xóa các kết quả đã có
 lblResultCount.Text = mc.Count.ToString & " matches"
 txtResults.Clear()

 Dim m As Match
 ' Duyệt qua các kết quả tìm được, và hiển thị các giá trị nhóm và/hoặc
  ' giá trị tìm được.

 For Each m In mc
   AppendResults("'" & m.Value & "' at index " & m.Index)
   If m.Groups.Count > 1 And chkShowGroups.Checked Then
     Dim g As Group, i As Integer
```

```
        ' Bỏ qua group 0, đây là toàn bộ đối tượng Match
        For i = 1 To m.Groups.Count - 1
          ' Lấy một tham chiếu đến group tương ứng
          g = m.Groups(i)
          AppendResults(String.Format("  group({0}) = '{1}'", i, g.Value))
          If chkShowCaptures.Checked = True Then
            ' Lấy tập hợp tìm thấy của group này
            Dim cc As CaptureCollection = g.Captures
            ' Hiển thị thông tin trên mỗi kết quả tìm thấy
            Dim c As Capture
            For Each c In cc
              AppendResults(String.Format("    " & _
                "capture '{0}' at index {1}", c.Value, c.Index))
            Next
          End If
        Next
      End If
    Next
  End Sub

  ' Xử lý sự kiện click cho nút "Find the Number". Nếu tìm số đầu tiên trong một
  ' chuỗi ký tự thì hiển thị số đó và vị trí bắt đầu của nó ở trong chuỗi.
  Private Sub btnFindNumber_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnFindNumber.Click

    ' Tạo một instance của RegEx và truyền vào một bộ mẫu, bộ này để tìm một
    ' hoặc nhiều con số.
    Dim re As New Regex("\d+")
    ' Gọi Match, truyền vào văn bản nguồn
    Dim m As Match = re.Match(txtFindNumber.Text)
    ' Nếu tìm thấy một kết quả, hiển thị kết quả. Nếu không, hiển thị một thông báo
    If m.Success Then
      lblResults.Text = String.Format("RegEx found " & _
        m.Value & " at position " & m.Index.ToString)
    Else
      lblResults.Text = "You didn't enter a string containing a number!"
    End If

  End Sub
```

```
' Nếu tìm thấy, thay thế văn bản trong văn bản nguồn bằng văn bản thay thế
Private Sub btnReplace_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReplace.Click
 Try
   txtResults.Text = Regex.Replace(txtSource.Text, txtRegEx.Text, _
     txtReplace.Text, GetRegexOptions())
 Catch exp As Exception
   MsgBox(exp.ToString, MsgBoxStyle.Critical, Me.Text)
 End Try
End Sub

' Xử lý nút "Scrape"
Private Sub btnScrape_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnScrape.Click
 Dim strHttpSource As String

 If UrlIsValid() Then
   ' Lấy chuỗi chứa địa chỉ của trang Web.
   Try
     strHttpSource = ConvertStreamToString(GetHttpStream(txtURL.Text))
   Catch exp As Exception
     ' Hiển thị thông báo lỗi
     MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
     Exit Sub
   End Try

   Dim strRegExPattern As String

   ' Gán địa chỉ Web vào một biến
   strUrlEntered = Trim(txtURL.Text)
   ' Lấy tên Domain để sử dụng cho nhiều mục đích khác.
   Dim astrDomain() As String = Regex.Split(strUrlEntered, "/")
   ' Phần tử đầu tiên (0) là "http:" và phần tử thứ ba thực sự là Domain name
   strDomain = astrDomain(0) & "//" & astrDomain(2)
   If optLinks.Checked Then
     ShowStatusIndicators("Connecting to Web page to screen scrape " & _
       "the links. Please stand by...")

     lvwLinks.Items.Clear()
```

```
        ' Biểu thức regular expression ở đây được sử dụng để phân tích liên kết HTML.
        ' Ví dụ: <a href="http://www.ntbooksvn.com">NT Books</a> được giải thích
        ' như sau:
        '   <a          Bắt đầu một liên kết HTML
        '   \s+         Một hoặc nhiều ký tự khoảng trắng
        '   href        Tiếp tục với văn bản chính xác trong liên kết HTML
        '   \s*         Không có hoặc nhiều ký tự khoảng trắng
        '   =           Tiếp tục với văn bản chính xác trong liên kết HTML
        '   \s*         Không có hoặc nhiều ký tự khoảng trắng
        '   ""?         Dấu ngoặc kép
        '   (           Bắt đầu nhóm định nghĩa một chuỗi con: URL thực
        '   [^"" >]+    Một hoặc nhiều ký tự đều phù hợp ngoại trừ các ký tự trong
        '                 dấu ngoặc vuông
        '   )           Kết thúc nhóm định nghĩa đầu tiên
        '   ""?         Dấu ngoặc kép
        '   >           Văn bản chính xác trong liên kết HTML
        '   (.+)        Nhóm chứa bất kỳ ký tự nào: Văn bản hiển thị
        '   </a>        Kết thúc liên kết HTML
        '
        strRegExPattern = "<a\s+href\s*=\s*""?([^"" >]+)""?>(.+)</a>"
    Else
        ShowStatusIndicators("Connecting to Web page to screen scrape the " & _
            "images. Please stand by...")

        htImages = New Hashtable()
        lvwImages.Items.Clear()

    ' Biểu thức regular expression để lấy ra các hình ảnh cũng tương tự như
    ' mẫu để lấy các liên kết HTML. Tuy nhiên, trong trường hợp này phức tạp hơn
    ' bởi vì chúng ta lấy bốn thông tin thuộc tính khác mà có thể nằm không đúng thứ tự.
    '
    ' Để giảm các trường hợp, ta giả sử một số quy tắt về các tag <img> được xử lý
    ' trên trang Web:
        '
        '   1. Thuộc tính src luôn luôn có. Đây là thuộc tính bắt buộc.
        '   2. Thuộc tính src đứng trước width và height. Nếu không, width và
        '       height không được lấy thông tin.
        '   3. Thuộc tính alt đứng sau width và height. Nếu không, bỏ qua thông
```

```
        '      tin của alt.
        '  4. Width và height có thể theo sau src theo bất kỳ thứ tự nào

        ' Điều này bảo đảm là tất cả các hình ảnh trong trang Web được lấy thông tin.
        ' Tuy nhiên có thể một số thuộc tính không lấy được thông tin.
        '
        ' Một số cụm ký tự chính sử dụng trong mẫu này là:
        '
        '  [^>]+      Tất cả các ký tự trừ >.
        '  (?:        Bắt đầu một nhóm không theo dõi. Cụm này dùng với )? để
        '             tạo một group có thể có hoặc không.
        '  |          Sử dụng với width và height như là một biểu thức Or. Chú ý là cặp
        '             đầu tiên width đứng trước, cặp thứ hai có thứ tự ngược lại.
        strRegExPattern = "<img[^>]+(src)\s*=\s*""?([^ ""]+)""?" & _
        "(?:[^>]+(width|height)\s*=\s*""?([^ ""]+)""?\s+(height|width)\s*=\s*""?" & _
        "([^ ""]+)""?)?(?:[^>]+(alt)\s*=\s*""?([^""]+)""?)?"
    End If

    Dim re As New Regex(strRegExPattern, RegexOptions.IgnoreCase)
    Dim m As Match = re.Match(strHttpSource)

    ' Xử lý source code HTML. Bởi vì source là một chuỗi dài, thay vì sử dụng
    ' phương thức Matches, sử dụng Match() để trả về mỗi lần một phần tìm thấy.
    ' Thuộc tính Success quyết định có còn tìm thấy phần nào khác hay không.
    ' NextMatch() thực hiện việc lặp lại.
    While m.Success
        If optImages.Checked Then
            Dim strWidth, strHeight As String
            ' Bởi vì mẫu tìm kiếm có khả năng có hoặc không có các thuộc tính width
            ' và height, xác định thuộc tính nào được liệt kê trước và gán giá trị
            ' group thích hợp.
            If m.Groups(3).Value.ToLower = "width" Then
                strWidth = m.Groups(4).Value
                strHeight = m.Groups(6).Value
            Else          ' Thuộc tính height xuất hiện trước
                strWidth = m.Groups(6).Value
                strHeight = m.Groups(4).Value
            End If
```

```
        ' Gọi AddImage để thêm một instance của đối tượng ImageAttributes vào
        ' một HashTable. Xem thủ tục AddImage để rõ hơn.
        AddImage(New ImageAttributes(m.Groups(2).Value, _
          m.Groups(8).Value, strHeight, strWidth))
      Else
        ' Tạo một đối tượng ListViewItem và quy định tựa đề của cột đầu tiên ("src")
        Dim lvi As New ListViewItem()
        lvi.Text = m.Groups(1).Value
        lvwLinks.Items.Add(lvi)
      End If

      ' Tiếp tục vòng lặp tìm phần phù hợp kế tiếp
      m = m.NextMatch()
    End While

    HideStatusIndicators()
    ' Hiển thị các control tương ứng với các kết quả
    If optImages.Checked Then
      If Not IsNothing(htImages) Then
        FillImagesListView()
      Else
        MsgBox("No images were found whose attributes matched the " & _
          "regular expression.", MsgBoxStyle.Information, Me.Text)
      End If
    Else
      If lvwLinks.Items.Count = 0 Then
        MsgBox("No links were found whose Url matched the regular " & _
        "expression.", MsgBoxStyle.Information, Me.Text)
      End If
    End If
  End If
End Sub

' Thủ tục xử lý nút "Split". Thủ tục này tách (chia) văn bản nguồn vào một mảng
' chuỗi thay vì một MatchCollection (như trong thủ tục xử lý "Find").
' RegEx.Split tương tự như String.Split ngoại trừ là nó định nghĩa giới hạn bằng
' cách sử dụng một regular expression thay vì một ký tự.
Private Sub btnSplit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSplit.Click
```

```
' Xóa các kết quả đã có và split văn bản nguồn
txtResults.Clear()

' Ở đây ta gọi phương thức Shared Split, không cần một instanace của RegEx.
' Một mẫu tốt để thử với split là "\s*,\s*" tạo một mảng chứa các ký tự tách nhau bằng
' dấu phẩy. Trong trường hợp này, RegEx.Split(txtSource.Text, "\s*,\s*") sẽ cho
' kết quả giống với đoạn code sau:
'     txtRegEx.Text.Split(New Char() {CChar(",")})
'     Microsoft.VisualBasic.Split(txtRegEx.Text, ",")
'
' Ta có thể thay đổi regular expression để bỏ qua các phần tử trống trong
' mảng chuỗi kết quả: \s*[,]+\s*
Dim astrResults() As String
Try
  astrResults = Regex.Split(txtSource.Text, txtRegEx.Text, _
    GetRegexOptions())
Catch exp As Exception
  MsgBox("An error was encountered when attempting to parse the " & _
    "source text. This is often caused by an invalid regex pattern." & _
    " Check your expression and try again. Exp.Msg=" & exp.Message, _
    MsgBoxStyle.Critical, Me.Text)
  Exit Sub
End Try

' Để tách một mảng chuỗi thì ít nhất hai phần tử phải được tạo
If astrResults.Length > 1 Then
  AppendResults("The source text was split into " & _
    astrResults.Length & " items.")
  Dim s As String
  For Each s In astrResults
    AppendResults(s)
  Next
Else
  AppendResults("The source text could not be split. " & _
    "Check your regular expression pattern and try again.")
End If

End Sub
```

```
' Xử lý nút "Validate!" trên tab "Simple Samples".
Private Sub btnValidate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnValidate.Click
Dim IsValid As Boolean = True

' Khi thực hiện kiểm tra tính hợp lệ, hai phần tử mẫu được sử dụng thường xuyên:
'
'  ^ và $    Chuỗi nguồn có thể chỉ chứa những gì ở giữa hai dấu này,
'            ví dụ: một ngày tháng.
'            Bọc mẫu kiểm tra vào các ký tự này để bảo đảm tính chính xác.
'            Ví dụ, nếu không bọc bằng các ký tự này, có thể truyền vào ứng dụng
'            một ngày như sau: "kjdi12-25-2000xpL"
'
'  \s*       Tại vị trí đầu và vị trí cuối của một chuỗi, cho biết là khoảng trắng
'             có thể chấp nhận.
'
' Mẫu sau kiểm tra dữ liệu nhập vào có phải là một zip code đúng theo định dạng
' ddddd hoặc ddddd-dddd, với d là một ký số 0-9. Các phần tử mẫu chính sẽ
' sử dụng là:
'
'  \d        Bất kỳ chữ số nào (0-9).
'  |         Ký hiệu (tương ứng với Or) cho biết là Zip code có thể là 5 ký số
'            hoặc là 5 ký số theo sau là một dấu gạch ngang và 4 ký số.
If Not Regex.IsMatch(txtZip.Text, "^\s*(\d{5}|(\d{5}-\d{4}))\s*$") Then
  txtZip.ForeColor = Color.Red
  IsValid = False
Else
  txtZip.ForeColor = Color.Black
End If

' Mẫu sau kiểm tra chuỗi nhập vào có phải là một ngày có dạng mm-dd-yy hoặc
' mm-dd-yyyy. Các phần tử mẫu chính:
'
'  \d{1,2}   Các số ngày và tháng chỉ có thể có 1 hoặc hai chữ số. Sử dụng
'             (\d{4}|\d{2}) có nghĩa là năm chỉ có thể có 2 hoặc 4 chữ số.
'  (/|-)     Có thể dùng dấu sổ hoặc dấu gạch ngang làm ký tự phân cách.
'  \1        Dấu phân cách giữa ngày và năm phải giống với dấu phân cách giữa
'            tháng và ngày. Số 1 liên hệ tới nhóm đầu tiên, được định
'            nghĩa bởi các dấu ngoặc đơn, ví dụ (/|-).
```

```
If Not Regex.IsMatch(txtDate.Text, _
  "^\s*\d{1,2}(/|-)\d{1,2}\1(\d{4}|\d{2})\s*$") Then

  txtDate.ForeColor = Color.Red
  IsValid = False
Else
  txtDate.ForeColor = Color.Black
End If

' Mẫu sau kiểm tra chuỗi nhập vào có phải là một địa chỉ email đúng theo
' dạng "name@domain.com". Thực sự có thể không phải là ".com" có thể là các
' ký tự theo sau dấu chấm cuối cùng. Tên (name) có thể có nhiều từ phân tách
' bởi một hoặc nhiều dấu chấm. Tên Domain cũng có thể có nhiều từ phân cách
' bởi các dấu chấm.
'
'  ([\w-]+\.)*?   Bất ký tự nào (a-z, A-Z, 0-9, gạch dưới) hoặc gạch ngang, theo sau
'                 là một dấu chấm.
'  [\w-]          Ở cả hai phía của ký tự @  để bảo đảm địa chỉ
'                 có dạng name@domainname
'   +\.           Bảo đảm có ít nhất một dấu chấm trong domain name
'
If Not Regex.IsMatch(txtEmail.Text, _
  "^([\w-]+\.)*?[\w-]+@[\w-]+\.([\w-]+\.)*?[\w]+$") Then

  txtEmail.ForeColor = Color.Red
  IsValid = False
Else
  txtEmail.ForeColor = Color.Black
End If

If IsValid Then
  lblValid.Visible = True
Else
  lblValid.Visible = False
End If

End Sub
```

```
' Thủ tục này xử lý sự kiện DoubleClick trên Images ListView.
' Nhấp đúp chuột vào một hình ảnh sẽ mở form frmImageViewer, yêu cầu hình ảnh
' là một Stream từ Web server, và hiển thị nó trong một PictureBox.
Private Sub lvImages_DoubleClick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lvwImages.DoubleClick

  ' Đặt tất cả code trong thủ tục kiểm tra hợp lệ để bảo đảm người sử dụng không
   ' thay đổi địa chỉ thành một địa chỉ sai trước khi nhấp đúp chuột
  If UrlIsValid() Then
    Dim strImgSrc As String = lvwImages.SelectedItems(0).Text

    ShowStatusIndicators("Connecting to Web page to retrieve the " & _
      "selected image. Please stand by...")

    If IsNothing(frmImageViewer) Then
      frmImageViewer = New frmImageViewer()
    End If

    ' Nếu đường dẫn image được chọn không phải là một địa chỉ đầy đủ, làm cho
     ' nó đúng và hiển thị hình ảnh trên form ImageViewer.
    If Microsoft.VisualBasic.Left(strImgSrc, 7) <> "http://" Then
      strImgSrc = MakeRelativeUrlAbsolute(strImgSrc)
    End If

    frmImageViewer.Show(strImgSrc)

    HideStatusIndicators()
  End If

End Sub

' Xử lý sự kiện DoubleClick cho "Links" ListView. Nhấp đúp vào một liên kết sẽ
' mở trình duyệt Internet Explorer và hiển thị trang được yêu cầu.
Private Sub lvLinks_DoubleClick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lvwLinks.DoubleClick

  If UrlIsValid() Then

    Dim strClickedUrl As String = lvwLinks.SelectedItems(0).Text
```

```
    ShowStatusIndicators("Starting Internet Explorer and connecting to " & _
      "the selected Web page. Please stand by...")

    ' Xây dựng lại đường dẫn nếu chưa đầy đủ, và mở nó trên IE
    If Microsoft.VisualBasic.Left(strClickedUrl, 7) <> "http://" Then
      strClickedUrl = MakeRelativeUrlAbsolute(strClickedUrl)
    End If

    Process.Start("IExplore.exe", strClickedUrl)

    HideStatusIndicators()

  End If
End Sub

' Xử lý sự kiện CheckedChanged cho các control RadioButton trên tab "Screen
' Scrape". Các control ListView được hoán chuyển dựa trên option nào được chọn.
Private Sub RadioButtons_CheckedChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles optImages.CheckedChanged,
optLinks.CheckedChanged
  If optLinks.Checked Then
    lvwLinks.Visible = True
    lvwImages.Visible = False
  Else
    lvwImages.Visible = True
    lvwLinks.Visible = False
  End If
End Sub

' Thủ tục này thêm một đối tượng ImageAttributes vào HashTable nếu nó chưa được
' cho vào. Thuộc tính "src" được sử dụng như là khóa để tìm kiếm. Nếu khóa Src đã
' tồn tại, đối tượng không được thêm vào. Điều này tránh được việc các hình ảnh
' xuất hiện nhiều lần trong danh sách.
Protected Sub AddImage(ByVal imgAttr As ImageAttributes)
  If Not htImages.Contains(imgAttr.Src) Then
    htImages.Add(imgAttr.Src, imgAttr)
  End If

End Sub
```

```
' Thủ tục này hỗ trợ cho việc nối các kết quả
Sub AppendResults(ByVal msg As String)
 txtResults.AppendText(msg & ControlChars.CrLf)
End Sub

' Hàm này đọc một Stream trả về bởi đối tượng HttpWebResponse và chuyển
' nó sang một chuỗi để RegEx xử lý.
Function ConvertStreamToString(ByVal stmSource As Stream) As String
 Dim sr As StreamReader

 If Not IsNothing(stmSource) Then
   Try
     sr = New StreamReader(stmSource)
     ' Đọc và trả về toàn bộ nội dung của stream
     Return sr.ReadToEnd
   Catch exp As Exception
     ' Không hiển thị MsgBox. Chỉ chuyển thông báo lỗi cho GetHttpStream()
     Throw New Exception()
   Finally
     ' Giải phóng cả Stream và StreamReader
     wresScrape.Close()
     sr.Close()
   End Try
 End If

End Function

' Thủ tục này duyệt qua một HashTable và thêm một ListViewItem chứa
' ListViewItem.SubItems cho mỗi đối tượng ImageAttributes trong HashTable.
Protected Sub FillImagesListView()
 Dim strSrc As String

 For Each strSrc In htImages.Keys
   ' Tạo một đối tượng ListViewItem và cho nội dung vào cột đầu tiên
   Dim lvi As New ListViewItem()
   lvi.Text = strSrc

   ' Đưa nội dung vào các cột còn lại và thêm đối tượng ListViewItem vào ListView
   Dim imgAttr As ImageAttributes = CType(htImages(strSrc), ImageAttributes)
```

```
        With lvi.SubItems
            .Add(imgAttr.Alt)
            .Add(imgAttr.Height)
            .Add(imgAttr.Width)
        End With

        ' Cho ListViewItem vào tập hợp các Item của ListView
        lvwImages.Items.Add(lvi)
    Next
End Sub

' Hàm này sử dụng các class của .NET được phát sinh từ System.Net.WebRequest
' để lấy Stream trả lời HTTP và là nguồn cho RegEx phân tích hoặc là hình ảnh để
' hiển thị khi được gọi bởi frmImageViewer.Show().
Function GetHttpStream(ByVal url As String) As Stream
    ' Tạo một yêu cầu sử dụng WebRequestFactory.
    wreqScrape = CType(WebRequest.Create(url), HttpWebRequest)
    With wreqScrape
        .UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.1)"
        .Method = "GET"
        .Timeout = 10000
    End With

    Try
        ' Trả về Stream trả lời
        wresScrape = CType(wreqScrape.GetResponse(), HttpWebResponse)
        Return wresScrape.GetResponseStream()
    Catch exp As Exception
        ' Khi một lỗi có thể do sai Url hoặc chưa kết nối Internet, tạo một thông
        ' báo lỗi trả về cho hàm gọi.
        Throw New Exception("There was an error retrieving the Web page " & _
            "you requested. Please check the Url and your connection to " & _
            "the Internet, and try again.")
    End Try
End Function

' Thủ tục hỗ trợ cho việc lấy các option cho regular expression
Function GetRegexOptions() As RegexOptions
    ' Sử dụng bất kỳ chọn lựa nào của RegExOptions cũng tương tự với đặt toàn bộ
```

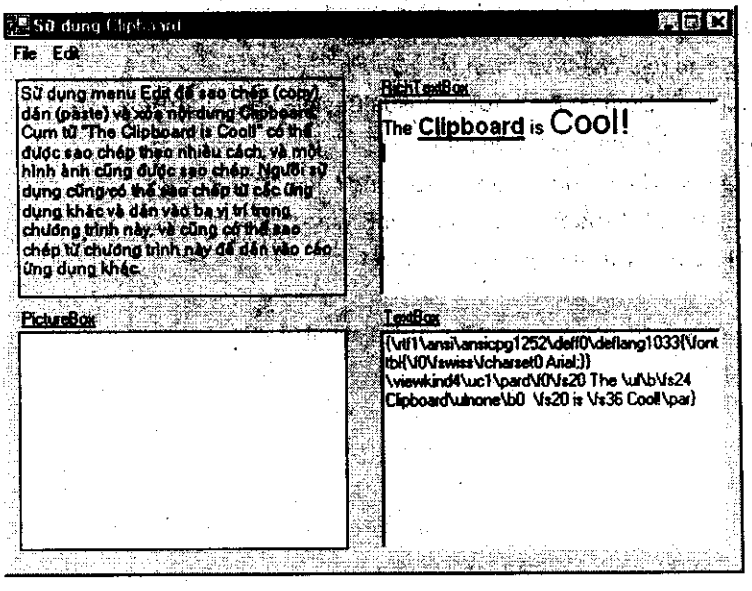
```
    ' regular expression vào trong dấu ngoặc đơn (ví dụ: làm cho toàn bộ mẫu trong
    ' một nhóm riêng) và sau đó mở đầu nó bên trong dấu ngoặc đơn mở với ký
    ' tự option. Nếu muốn điều khiển tốt hơn trên một option, tạo một nhóm bên trong
    ' mẫu và dùng cùng cú pháp trên. Trong trường hợp không muốn sử dụng các
    ' RegExOptions, tắt nó bằng cách phủ định nó, ví dụ: để tắt tính năng phân biệt chữ
    ' hoa và chữ thường, gõ ?-i: bên trong dấu ngoại đơn mở của một nhóm.
    '
    ' Các option mặc nhiên được tắt
    '
    ' IgnoreCase bật/tắt tính năng phân biệt chữ hoa chữ thường cho toàn bộ mẫu. Sử
    ' dụng option này tương đương với gõ (?i:mẫu gốc) trong regular expression
    If chkIgnoreCase.Checked Then GetRegexOptions = RegexOptions.IgnoreCase

    ' Singleline thay đổi cách cư xử của ký tự chấm (.) để nó thỏa bất kỳ ký tự nào (thay
    ' cho cách mặc nhiên là thỏa bất kỳ ký tự ngoại trừ ký tự dòng mới, \r hoặc \n). Sử
    ' dụng option này tương đương với dùng (?s:mẫu gốc) trong cú pháp regular
    ' expression.
    '
    ' Chú ý là nhiều RegExOptions có thể sử dụng khi được phân tách bằng toán tử Or.
    ' Ngoài ra, có thể cho phép nhiều option bằng cách sử dụng cú pháp của regular
    ' expression bằng cách đặt các option với nhau sau ký tự ?. ví dụ, để bật tính năng
    ' Singline và ignore case, cú pháp như sau: (?si:mẫu gốc).
    If chkSingleLine.Checked Then
      GetRegexOptions = GetRegexOptions Or RegexOptions.Singleline
    End If

    ' Multiline thay đổi cách cư xử của các ký tự ^ và $ để chúng thỏa từ đầu đến cuối
    ' của một dòng thay vì từ đầu đến cuối của một chuỗi. Cú pháp tương ứng trong
    ' regular expression là: (?m:mẫu gốc).
    If chkMultiline.Checked Then
      GetRegexOptions = GetRegexOptions Or RegexOptions.Multiline
    End If
  End Function

  ' Thủ tục này tắt form thông báo trạng thái được hiển thị bởi ShowStatusIndicators
  Private Sub HideStatusIndicators()
    frmStatus.Hide()
    Me.Cursor = Cursors.Default
  End Sub
```

```
' Hàm này nhận một địa chỉ tương đối và chuyển nó thành địa chỉ tuyệt đối. Đây là
' hàm hỗ trợ cho sự kiện DoubleClick trên tab "Screen Scrape". Các thuộc tính "href"
' và "url" có thể chứa một đường dẫn tuyệt đối, tương đối với gốc, hoặc một đường
' dẫn tương đối. Nếu đường dẫn được chọn chưa là đường dẫn tuyệt đối, hàm này
' nối vào phía trước nó tên domain và một dấu sổ nếu cần.
Public Function MakeRelativeUrlAbsolute(ByVal strRelativeUrl As String) As String

 ' Nếu là đường dẫn tương đối với gốc (bắt đầu bằng "/"), nó cần thêm domian name
 ' đứng trước. Nếu là một địa chỉ tương đối nó cần thêm địa chỉ do người sử dụng
 ' nhập vào.

 ' Nếu là đường dẫn tương đối với gốc
 If Microsoft.VisualBasic.Left(strRelativeUrl, 1) = "/" Then
   ' đặt tên domain đứng trước
   Return strDomain & strRelativeUrl
 ElseIf Microsoft.VisualBasic.Left(strRelativeUrl, 3) = "../" Then
   ' Bỏ các dấu chấm và thêm tên domain
   Return strDomain & Microsoft.VisualBasic.Replace(strRelativeUrl, "../", "/")
 Else    ' Là đường dẫn tương đối
   ' kiểm tra địa chỉ nhập vào có kết thúc bằng "/", nếu không có thì thêm vào
   If Microsoft.VisualBasic.Right(strUrlEntered, 1) = "/" Then
     Return strUrlEntered & strRelativeUrl
   Else
     Return strUrlEntered & "/" & strRelativeUrl
   End If
 End If
End Function

' Thủ tục này hiển thị một form cho biết công việc đang được thực hiện và
' chuyển con trỏ chuột sang WaitCursor.
Private Sub ShowStatusIndicators(ByVal strMsg As String)

 frmStatus = New frmStatus()
 frmStatus.Show(strMsg)
 Me.Cursor = Cursors.WaitCursor

End Sub
```

```
' Hàm này kiểm tra địa chỉ nhập vào có bắt đầu bằng http:// không
Function UrlIsValid() As Boolean
 If Microsoft.VisualBasic.Left(Trim(txtURL.Text), 7) <> "http://" Then
   MsgBox("The Url must begin with http://.")
   Return False
 End If

 Return True
End Function

End Class
```

22. Sử dụng Clipboard

Ví dụ này chỉ cách đặt vào và lấy các item ra khỏi Clipboard với nhiều định dạng khác nhau. Nó biểu diễn các phương thức và thuộc tính chính được sử dụng khi làm việc với Clipboard.



Các chức năng chính:

- Chuỗi "The Clipboard is Cool!" được cung cấp ở nhiều định dạng khác nhau (là các biến của class). Chuỗi này có thể được copy vào Clipboard ở một định dạng, hoặc được đưa vào Clipboard với nhiều định dạng cùng lúc, tất cả đều ở trong menu Edit.
- Một hình ảnh có thể được load và đưa vào Clipboard từ menu Edit.

- Menu Paste As được phát sinh động từ các định dạng khác nhau có trong Clipboard.
- Các thành phần có thể được copy từ các ứng dụng Windows khác và dán vào ba vị trí trong chương trình này, và ngược lại.

Chạy ứng dụng và thử dùng các khả năng trong menu Edit. Ví dụ chọn **Edit -> Copy Text As -> RTF**. Sau đó chọn **Edit -> Paste As -> Rich Text Format**. Sau đó ta có thể thấy được sự khác nhau khi văn bản được dán vào RichTextBox và TextBox.

Sau đó làm tương tự với HTML. Chú ý là RichTextBox không hỗ trợ định dạng HTML. Thử dán nội dung vào một trình soạn thảo HTML (Visual Studio .NET HTML document) để thấy kết quả.

Thử dùng tính năng **Edit -> Copy As -> All Formats**, tất cả các phiên bản định dạng của văn bản được đặt vào Clipboard. Sau đó có thể dán với định dạng Text, RTF, HTML và XML.

Project: **Su dung Clipboard**

frmMain.vb

```
Option Strict On

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Các chuỗi sau chứa nhiều kiểu định dạng của văn bản " The Clipboard is Cool!"
    Dim strText As String = "The Clipboard is Cool!"
    Dim strHTML As String = "<P>The <B><FONT size='4'>" & _
      "<U>Clipboard</U></FONT></B>" & _
      "is <FONT size='5'>Cool!</FONT></P>"
    Dim strRTF As String = "{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{" & _
      "\f0\fswiss\fcharset0 Arial;}}\viewkind4\uc1\pard\f0\fs20 The \ul\b\fs24 " & _
      "Clipboard\ulnone\b0  \fs20 is \fs36 Cool!\par}"
    Dim strXML As String = "<?xml version='1.0'?><Message>The Clipboard " & _
      "is Cool!</Message>"

    ' myImage lưu một hình ảnh trong bộ nhớ
    Dim myImage As New System.Drawing.Bitmap("..\twodogs.jpg")

' Code phát sinh form được bỏ qua
```

```
' Sao chép hình ảnh vào Clipboard với định dạng bitmap
Private Sub mnuCopyImageAsBitmap_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuCopyImageAsBitmap.Click
    Dim myDataObject As New DataObject()
    myDataObject.SetData(DataFormats.Bitmap, True, myImage)
    Clipboard.SetDataObject(myDataObject, True)
End Sub

' Chép " The Clipboard is Cool!" vào Clipboard với tất cả định dạng có thể. Điều này
' chỉ có thể thực hiện được bằng cách sử dụng một instance DataObject. Phương
' thức SetData được gọi cho tất cả các định dạng, cùng với tên định dạng của chúng.
Private Sub mnuCopyTextAsAllFormats_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuCopyTextAsAllFormats.Click
    Dim myDataObject As New DataObject()

    ' Đặt định dạng văn bản và unicode vào DataObject
    myDataObject.SetData(DataFormats.Text, strText)
    myDataObject.SetData(DataFormats.UnicodeText, strText)

    ' Đặt định dạng HTML vào DataObject
    myDataObject.SetData(DataFormats.Html, strHTML)

    ' Đặt định dạng RTF vào DataObject
    myDataObject.SetData(DataFormats.Rtf, strRTF)
    ' Đặt định dạng XML vào DataObject
    myDataObject.SetData("MyInternalXmlFormat", strXML)
    ' Đặt myDataObject (chứa tất cả định dạng) vào Clipboard
    Clipboard.SetDataObject(myDataObject, True)

End Sub

' Chép chuỗi "The Clipboard is Cool!" vào Clipboard với định dạng HTML.
' Chuỗi HTML được cho vào DataObject trước để chỉ định định dạng cho nó.
Private Sub mnuCopyTextAsHTML_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuCopyTextAsHTML.Click
    Dim myDataObject As New DataObject()
    myDataObject.SetData(DataFormats.Html, strHTML)
    Clipboard.SetDataObject(myDataObject, True)
End Sub
```

```
' Chép chuỗi "The Clipboard is Cool!" vào Clipboard với định dạng RTF.
Private Sub mnuCopyTextAsRTF_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuCopyTextAsRTF.Click
    Dim myDataObject As New DataObject()
    myDataObject.SetData(DataFormats.Rtf, strRTF)
    Clipboard.SetDataObject(myDataObject, True)
End Sub

' Chép chuỗi "The Clipboard is Cool!" vào Clipboard với định dạng text.
' Chuỗi văn bản được chép trực tiếp vào Clipboard
Private Sub mnuCopyTextAsText_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuCopyTextAsText.Click
    Clipboard.SetDataObject(strText, True)
End Sub

' Chép chuỗi "The Clipboard is Cool!" vào Clipboard với định dạng XML.
' Bởi vì XML không được Clipboard hỗ trợ, đây là định dạng riêng của ứng dụng.
' Và bởi vì định dạng này ít có ý nghĩa cho các ứng dụng bên ngoài, tham số thứ
' hai được gán là False, bắt buộc Clipboard không cho phép các ứng dụng
' khác sử dụng dữ liệu này.
Private Sub mnuCopyTextAsMXL_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuCopyTextAsMXL.Click

    Dim myDataObject As New DataObject()
    myDataObject.SetData("MyInternalXmlFormat", strXML)
    Clipboard.SetDataObject(myDataObject, False)
End Sub

' MnuEdit_Popup bảo đảm là khi menu Edit được nhấn chuột thì menu Paste As sẽ
' có các chức năng tương ứng cho các định dạng dữ liệu có trong Clipboard
Private Sub mnuEdit_Popup(ByVal sender As Object, ByVal e As
System.EventArgs) Handles mnuEdit.Popup

    Dim myPasteAsMenu As MenuItem ' Sử dụng để xây dựng menu Paste As
    Dim myTypes() As MenuItem ' Sử dụng để hiển thị các định dạng khác nhau
    Dim strArray() As String ' Mảng chứa các định dạng có hỗ trợ
    Dim i As Integer
```

```
    ' Nếu Clipboard có dữ liệu, nhập vào mảng danh sách tất cả định dạng có hỗ trợ
    If Not (Clipboard.GetDataObject() Is Nothing) Then
        ' GetFormats() trả về một mảng chuỗi chứa các định dạng có hỗ trợ
        strArray = Clipboard.GetDataObject().GetFormats()
        ReDim myTypes(strArray.Length - 1)
        ' Tạo các MenuItem mới sử dụng chung phương
        ' thức PasteAsMenuEventHandler để xử lý
        For i = 0 To strArray.Length - 1
            myTypes(i) = New MenuItem(strArray(i), New _
                System.EventHandler(AddressOf PasteAsMenuEventHandler))
        Next i
    End If

    ' Bảo đảm là các thủ tục của .NET Framework hiển thị menu
    Me.mnuEdit.OwnerDraw = False
    ' Tạo menu Paste As với các menu con là các định dạng có sẵn
    myPasteAsMenu = New MenuItem("&Paste As", myTypes)
    ' Nếu nó đã được thêm vào rồi, xóa nó
    If mnuEdit.MenuItems.Count = 3 Then
        mnuEdit.MenuItems.RemoveAt(2)
    End If
    ' Thêm menu Paste As vào menu Edit
    Me.mnuEdit.MenuItems.Add(myPasteAsMenu)
End Sub

' PasteAsMenuEventHandler xử lý tất cả các lệnh Paste, không quan tâm đến
' định dạng nào đã được chọn.
Private Sub PasteAsMenuEventHandler(ByVal sender As Object, ByVal e As
System.EventArgs)
    Dim strType As String ' chứa giá trị định dạng
    Dim obj As Object ' chứa dữ liệu sẽ được dán

    ' Xóa nội dung tất cả các đích đến
    Me.rtbPaste.Clear()
    Me.txtPaste.Clear()
    Me.picturePaste.Image = Nothing

    ' Lấy định dạng do người sử dụng chọn như là một chuỗi
    strType = CType(sender, MenuItem).Text
```

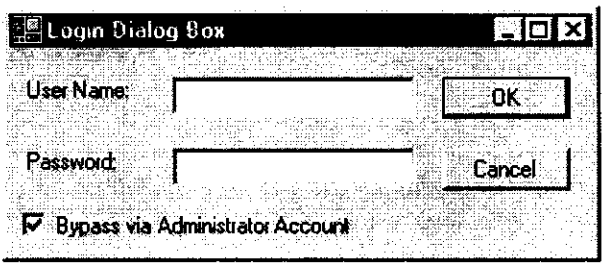
```
        ' Bảo đảm là Clipboard có hỗ trợ định dạng được chọn
        If Clipboard.GetDataObject().GetDataPresent(strType) Then
            ' Gán dữ liệu trong clipboard vào obj với định dạng đã yêu cầu
            obj = Clipboard.GetDataObject().GetData(strType)
            If Not obj Is Nothing Then

                ' Dán vào RichTextBox bằng phương thức Paste
                Me.rtbPaste.Paste(DataFormats.GetFormat(strType))

                ' Dán văn bản miêu tả vào TextBox
                If obj.GetType().ToString() = "System.String" Then
                    Me.txtPaste.AppendText(CType(obj, String))
                Else
                    Me.txtPaste.AppendText(obj.GetType.ToString())
                End If

                ' Thực hiện dán vào PictureBox. Nếu không được, PictureBox không hỗ trợ
                ' định dạng này, do đó xóa nội dung ảnh
                Try
                    picturePaste.Image = CType(obj, Image)
                Catch ex As Exception
                    picturePaste.Image = Nothing
                End Try
            End If
        End If

    End Sub
End Class
```

Phần II - Security

23. Tạo hộp thoại Login

Project này biểu diễn cách tạo một hộp thoại login để người sử dụng đăng nhập vào ứng dụng và giới hạn các chức năng trong ứng dụng. Người sử dụng trước tiên phải đăng nhập, chỉ được 3 lần thử nếu không thành công, có thể dùng phương cách đăng nhập riêng của ứng dụng hoặc dùng Windows Authentication (account đăng nhập vào Windows). Sau khi đăng nhập, quyền

hạn của người sử dụng được kiểm tra. Nếu là thành viên của "Manager" hoặc Windows Administrator thì form chính hiển thị danh sách người sử dụng chứa trong một file XML, nếu không phải thì form hiển thị là form trống.



Các chức năng chính:

- Một đối tượng Windows Principal được sử dụng để xác định người sử dụng có đăng nhập vào Windows với quyền Administrator hay không.
- Đối tượng Generic Principal được sử dụng để xác định người sử dụng có phải là thành viên của "Manager".
- Một file XML được load vào một DataSet và được dùng để kiểm tra thông tin người sử dụng.
- Nếu đăng nhập với quyền Administrator hoặc Manager (tên Account: Bob) thì danh sách người sử dụng và mật mã hiển thị lên. Nếu đăng nhập với account John thì không hiển thị gì cả.

File **Users.xml** được lưu trong thư mục bin và được sử dụng để lưu thông tin account của các user. Vì để đơn giản, file XML này không được mã hóa, việc mã hóa tránh được người sử dụng xem trực tiếp nội dung của file xml.

Trong Users.xml có 2 account:

Name	Password	Role
Bob	pass	Manager
John	pass	Sales

users.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<users>
  <user>
   <name>Bob</name>
   <password>pass</password>
   <role>Manager</role>
  </user>
  <user>
   <name>John</name>
```

```
    <password>pass</password>
    <role>Sales</role>
  </user>
</users>
```

Users.vb

```
Option Strict On
Imports System.Security.Principal
Imports System.Threading
Imports System.IO

Public Class Users
  Function IsLogin(ByVal strName As String, ByVal strPassword As String) As
  Boolean
    ' Thủ tục kiểm tra người sử dụng có tồn tại trong file XML hay không
    Dim dsUsers As New DataSet()
    Dim drRows() As DataRow

    Try
      ' Đọc file XML vào một DataSet và lọc trên tên (name) và mật mã (password)
      ' để được một tập hợp DataRows. Phương thức này không phân biệt chữ
      ' hoa và chữ thường.
      dsUsers.ReadXml("..\Users.xml")
      drRows = dsUsers.Tables(0).Select("name = '" & _
          strName & "' and password = '" & strPassword & "'")

' Phải viết code kiểm tra khi thêm user vào danh sách để bảo đảm không trùng tên
      ' Nếu có một dòng trong tập hợp thì một record được tìm thấy
      If drRows.Length > 0 Then
        Return True
      Else
        Return False
      End If
    Catch e As FileNotFoundException
      MsgBox("Users.Xml file not found.", MsgBoxStyle.Critical, _
                                  "Unable to Authenticate user.")
      End
    End Try
  End Function
```

```
Function GetLogin(ByVal strName As String, ByVal strPassword As String) As
GenericPrincipal
    ' Hàm trả về một Generic Principal tương ứng cho login account

    Dim dsUsers As New DataSet()
    Dim drRows() As DataRow

    Try
        ' Đọc XML vào DataSet và lọc lấy một tập hợp DataRows
        dsUsers.ReadXml("..\Users.xml")
        drRows = dsUsers.Tables(0).Select("name = '" & _
            strName & "' and password = '" & strPassword & "'")
    Catch e As FileNotFoundException
        MsgBox("Users.Xml file not found.", MsgBoxStyle.Critical, "Shutting Down...")
        End
    End Try

    ' Tạo Generic Identity tương ứng cho User
    Dim GenIdentity As New GenericIdentity(strName)
    ' Định nghĩa quyền hạn cho user là một mảng
    Dim Roles() As String = {CStr(drRows(0).Item("Role")), ""}
    Dim GenPrincipal As New GenericPrincipal(GenIdentity, Roles)

    Return GenPrincipal
End Function

Function IsAdministrator() As Boolean
    ' Hàm kiểm tra Windows Login có phải là Administrator

    ' single role-based validation
    ' Dim WinPrincipal As New WindowsPrincipal(WindowsIdentity.GetCurrent())

    ' repeated role-based validation
    AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal)
    Dim WinPrincipal As WindowsPrincipal = CType(Thread.CurrentPrincipal,
WindowsPrincipal)

    ' Kiểm tra người sử dụng có là Administrator
    If WinPrincipal.IsInRole(WindowsBuiltInRole.Administrator) Then
```

```
        Return True
      Else
        Return False
      End If
    End Function
End Class
```

frmLogin.vb

```
Imports System.Threading
Imports System.Security.Principal

Public Class frmLogin
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
  Dim intLoginAttempts As Integer

  Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnCancel.Click
    End
  End Sub

  Private Sub btnOK_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnOK.Click
    ' Tạo instance của class Users
    Dim objUser As New Users()
    Dim GenPrincipal As GenericPrincipal
    Dim strName As String = txtUserName.Text
    Dim strPassword As String = txtPassword.Text

    ' Kiểm tra Windows Administrator. Administrator có thể bỏ qua hệ thống
    ' an toàn của ứng dụng
    If chkAdministratorAccount.Checked Then
      If objUser.IsAdministrator Then
        ' Hiển thị Users Name
        MsgBox(Thread.CurrentPrincipal.Identity.Name & _
        " has logged in successfully!", MsgBoxStyle.Information, "Login Successful")

        ' Hiển thị form Main
        Dim Main As New frmMain()
```

```
            Main.ShowDialog()

            ' Đóng form Login
            Me.Close()
        Else
            ' Tăng số lần thực hiện login
            intLoginAttempts += 1
            MsgBox("User not an Administrator.  Please provide a User Name and
Password.", MsgBoxStyle.Exclamation, Me.Text)
        End If
    Else
        ' Kiểm tra user login có tồn tại
        If objUser.IsLogin(strName, strPassword) Then
            GenPrincipal = objUser.GetLogin(strName, strPassword)
            Thread.CurrentPrincipal = GenPrincipal

            MsgBox(Thread.CurrentPrincipal.Identity.Name & " _
            has logged in successfully!", MsgBoxStyle.Information, "Login Successful")

            Dim Main As New frmMain()
            Main.ShowDialog()

            Me.Close()
        Else
            intLoginAttempts += 1

            ' Sau 3 lần đăng nhập không thành công thì thoát chương trình
            If intLoginAttempts >= 3 Then
                MsgBox("Too many failed login attempts", _
                                MsgBoxStyle.Exclamation, Me.Text)
                End
            Else
                MsgBox("User Name not found.  Please try again", _
                                MsgBoxStyle.Exclamation, Me.Text)
            End If
        End If
    End If

  End Sub
End Class
```

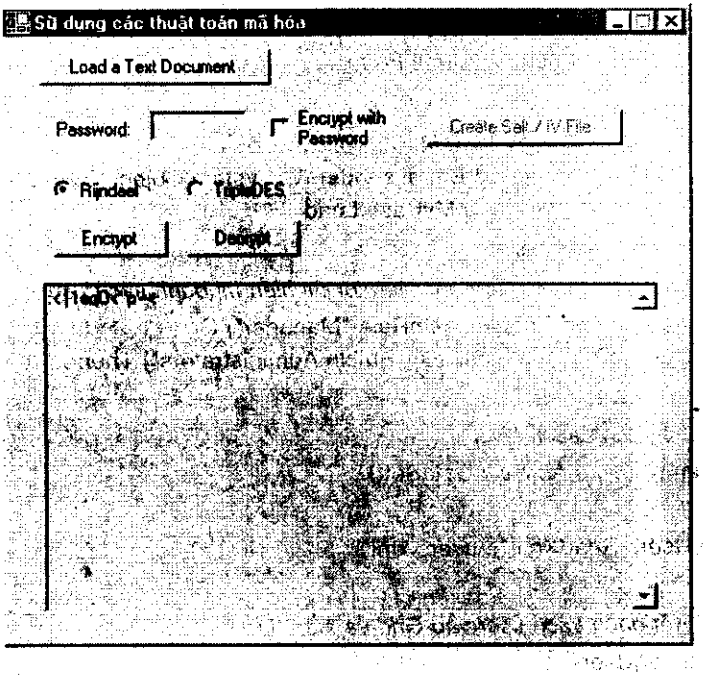
frmMain.vb

```
Option Strict On
Imports System.Threading

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        ' Kiểm tra quyền hạn
        ' Nếu là Manager hoặc Administrator thì hiển thị danh sách user
        If Thread.CurrentPrincipal.IsInRole("Manager") Or
    Thread.CurrentPrincipal.IsInRole("BuiltIn\Administrators") Then

            ' Lấy danh sách User
            Dim dsUsers As New DataSet()

            dsUsers.ReadXml("..\users.xml")

            ' Kết danh sách user vào Grid và hiển thị
            With dgUsers
                .CaptionText = "User Accounts"
                .DataSource = dsUsers.Tables(0)
            End With

        Else
            MsgBox("You must be a member of the Manager or Administrator's roles to " & _
                "view username and password information", MsgBoxStyle.Exclamation, _
                "Insufficient Permissions")
        End If
    End Sub
End Class
```

24. Mã hóa và giải mã dữ liệu

Mã hóa Symmetric sử dụng một khóa bí mật, được chia sẻ để mã hóa và giải mã dữ liệu. Mã hóa dùng khóa bí mật còn được gọi là mã hóa cân đối bởi vì dùng cùng một khóa để mã hóa và giải mã, làm cho việc bảo vệ khóa là

quan trọng. Mã hóa private-key nhanh hơn là mã hóa public-key làm cho nó lý tưởng với việc mã hóa trên các stream dữ liệu lớn. Tuy nhiên, trong thực tế, mã hóa public-key được sử dụng kết hợp với mã hóa private-key để chuyển private key cho đối tác khác.



Các chức năng chính:

- Chương trình này chỉ cách sử dụng Rijndael hoặc TripleDES để mã hóa và giải mã các file văn bản. Khi đánh dấu chọn "Encrypt with password", khóa không được phát sinh tự động mà được phát sinh từ một mật mã (password) đã được xử lý ("salted"). Hơn thế nữa, salt và initialization vector (IV) được lưu trong một file .dat. Với cách này ta thấy được cách để gửi một tài liệu cho ai đó được an toàn: tài liệu được mã hóa có thể gửi đi trên đường không an toàn, file .dat và password được truyền đi an toàn với phương pháp mã hóa assymetric (hoặc public key).

Thay vì tạo một file khóa .dat ta có thể ghi salt và IV vào đầu tài liệu đã được mã hóa như là văn bản chưa mã hóa. Khi giải mã, đọc salt và IV từ tài liệu đã mã hóa như là đọc trên file .dat. Với cách này, chỉ cần tài liệu được mã hóa và password.

Project: **Ma hoa va giai ma du lieu**

SampleCrypto.vb

```
Option Strict On

Imports System.IO
Imports System.Security.Cryptography
Imports System.Threading

Class SampleCrypto

    ' Thủ tục này tạo mới một đối tượng symmetric algorithm với kiểu được chọn
    Public Sub New(ByVal strCryptoName As String)
        crpSym = SymmetricAlgorithm.Create(strCryptoName)
        ReDimByteArrays()
    End Sub

    Private abytIV() As Byte
    Private abytKey() As Byte
    Private abytSalt() As Byte
    Private crpSym As SymmetricAlgorithm
    Private strPassword As String = ""
    Private strSaltIVFile As String = ""
    Private strSourceFile As String = ""

    Public Property Password() As String
        Get
            Return strPassword
        End Get
        Set(ByVal Value As String)
            strPassword = Value
        End Set
    End Property

    Public Property SaltIVFile() As String
        Get
            Return strSaltIVFile
        End Get
        Set(ByVal Value As String)
            If File.Exists(Value) Then
                strSaltIVFile = Value
            Else
```

```
            Throw New FileNotFoundException("The SaltIV .dat file for the " & _
              "selected crypto type was not found. Before encrypting or " & _
              "decrypting you must create this file.")
          End If
        End Set
      End Property

      Public Property SourceFileName() As String
        Get
          Return strSourceFile
        End Get
        Set(ByVal Value As String)
          If File.Exists(Value) Then
            strSourceFile = Value
          Else
            Throw New FileNotFoundException(Value & " does not exist.")
          End If
        End Set
      End Property

      ' Thủ tục này tạo một file .dạt chứa salt và IV
      Public Function CreateSaltIVFile(ByVal strSaveToPath As String) As Boolean

        ' Định lại kích thước các mảng cho phù hợp
        ReDimByteArrays()

        ' Tạo một đối tượng FileStream để ghi salt và IV vào một file
        Dim fsKey As New FileStream(strSaveToPath, FileMode.OpenOrCreate, _
          FileAccess.Write)

        ' Phát sinh một giá trị salt ngẫu nhiên. Các byte ngẫu nhiên này được nối
        ' vào password trước khi phát sinh khóa (key). Tương tự cho IV.
        Dim rng As RandomNumberGenerator = RandomNumberGenerator.Create()
        rng.GetBytes(abytSalt)

        Dim pdb As New PasswordDeriveBytes(strPassword, abytSalt)
        ' Lấy cùng số byte của abytKey trong ReDimByteArrays()
        abytKey = pdb.GetBytes(abytKey.Length)
```

```
        ' Phát sinh một IV ngẫu nhiên mới
        crpSym.GenerateIV()
        abytIV = crpSym.IV

        Try
            fsKey.Write(abytSalt, 0, abytSalt.Length)
            fsKey.Write(abytIV, 0, abytIV.Length)
            strSaltIVFile = strSaveToPath
            Return True
        Catch exp As Exception
            Throw New Exception(exp.Message)
        Finally
            fsKey.Close()
        End Try
    End Function

    ' Thủ tục này giải mã một file
    Public Sub DecryptFile()

        ' Nếu không sử dụng password thì sử dụng các giá trị mặc nhiên của
        ' đối tượng symmetric algorithm
        If strPassword <> "" Then
            OpenSaltIVFileAndSetKeyIV()
        End If

        ' Tạo một FileStream để đọc file đã mã hóa
        Dim fsCipherText As New FileStream(strSourceFile, FileMode.Open,
FileAccess.Read)
        ' Tạo một FileStream để ghi vào file tạm thời
        Dim fsPlainText As New FileStream("temp.dat", FileMode.Create,
FileAccess.Write)

        ' Đọc file đã mã hóa và giải mã
        Dim csDecrypted As New CryptoStream(fsCipherText, _
                                            crpSym.CreateDecryptor(), _
                                            CryptoStreamMode.Read)
        ' Tạo một StreamWriter để ghi vào file tạm
        Dim swWriter As New StreamWriter(fsPlainText)
        ' Đọc stream đã giải mã vào StreamReader
        Dim srReader As New StreamReader(csDecrypted)
```

```
    Try
        ' Ghi stream đã giải mã
        swWriter.Write(srReader.ReadToEnd)
    Catch expCrypto As CryptographicException
        Throw New CryptographicException()
    Finally
        ' Đóng và giải phóng tài nguyên
        swWriter.Close()
        csDecrypted.Close()
    End Try

    SwapFiles(True)
End Sub

' Thủ tục này mã hóa một file
Public Sub EncryptFile()

    ' Nếu không sử dụng password thì sử dụng các giá trị mặc nhiên của
    ' đối tượng symmetric algorithm
    If strPassword <> "" Then
        OpenSaltIVFileAndSetKeyIV()
    End If

    ' Tạo một FileStream để đọc từ file nguồn
    Dim fsInput As New FileStream(strSourceFile, FileMode.Open, FileAccess.Read)

    ' Tạo một mảng byte từ đối tượng FileStream
    Dim abytInput(CInt(fsInput.Length - 1)) As Byte
    fsInput.Read(abytInput, 0, CInt(fsInput.Length))
    fsInput.Close()

    ' Tạo một FileStream để ghi ra file tạm
    Dim fsCipherText As New FileStream("temp.dat", FileMode.Create,
FileAccess.Write)
    fsCipherText.SetLength(0)

    Dim csEncrypted As New CryptoStream(fsCipherText, _
                                        crpSym.CreateEncryptor(), _
                                        CryptoStreamMode.Write)
```

```
        ' Truyền vào mảng byte của file nguồn và ghi ra các byte đã mã hóa
        ' vào file temp.dat. Dòng logic như sau:
        ' abytInput ----> Encryption ----> fsCipherText.
        csEncrypted.Write(abytInput, 0, abytInput.Length)

        csEncrypted.FlushFinalBlock()

        csEncrypted.Close()

        SwapFiles(False)
    End Sub

    ' Thủ tục này mở file .dat, đọc vào salt và IV, sau đó quy định khóa và IV của
    ' đối tượng crypto
    Private Sub OpenSaltIVFileAndSetKeyIV()

        ReDimByteArrays()

        ' Tạo một FileStream đọc nội dung của file .dat chứa salt và IV
        Dim fsKey As New FileStream(strSaltIVFile, FileMode.Open, FileAccess.Read)
        fsKey.Read(abytSalt, 0, abytSalt.Length)
        fsKey.Read(abytIV, 0, abytIV.Length)
        fsKey.Close()

        ' Phát sinh key từ salted password
        Dim pdb As New PasswordDeriveBytes(strPassword, abytSalt)
        abytKey = pdb.GetBytes(abytKey.Length)

        ' Nếu class crypto hiện hành là TripleDES, kiểm tra key được sử dụng không
        ' phải là một trong các key yếu (Weak Keys) (ví dụ: key đã bị tấn công)
        If crpSym.GetType Is GetType(TripleDESCryptoServiceProvider) Then
            ' Để sử dụng phương thức IsWeakKey cần ép kiểu biến SymmetricAlgorithm
            ' sang class cơ sở TripleDES hoặc TripleDESCryptoServiceProvider
            Dim tdes As TripleDES = CType(crpSym, TripleDES)
            If tdes.IsWeakKey(abytKey) Then
                Throw New Exception("The current key is listed as a Weak Key. " & _
                    "You should generate a different key before proceeding further.")
            End If
        End If
```

```
    ' Gán các mảng byte vào các thuộc tính Key và IV của đối tượng crypto
    crpSym.Key = abytKey
    crpSym.IV = abytIV
End Sub

' Thủ tục này định lại kích thước các mảng byte cho có độ dài đúng với
' đối tượng crypto
Private Sub ReDimByteArrays()
    ' Phục vụ cho mục đích xem thông tin, ghi ra các kích thước key hợp lệ
    Debug.WriteLine(crpSym.GetType.ToString & " legal key sizes in bits:")
    Dim myKeySizes As KeySizes
    For Each myKeySizes In crpSym.LegalKeySizes
        With myKeySizes
            Debug.WriteLine("Max=" & .MaxSize & " bits " & _
                "(" & (.MaxSize / 8) & " bytes)")
            Debug.WriteLine("Min=" & .MinSize & " bits " & _
                "(" & (.MinSize / 8) & " bytes)")
            Debug.WriteLine("Skip=" & .SkipSize & " bits " & _
                "(" & (.SkipSize / 8) & " bytes)")
        End With
    Next

    If crpSym.GetType Is GetType(System.Security.Cryptography.RijndaelManaged)
Then
        ' Nhớ rằng kích thước của mảng lúc nào cũng lớn hơn 1 so với chỉ số cao nhất
        ' của mảng, chỉ số khai báo mảng. Do đó ReDim lại kích thước phải nhỏ
        ' hơn 1 so với kích thước key hợp lệ.
        ReDim abytKey(31)
        ' Chiều dài của salt bằng ½ chiều dài của key.
        ReDim abytSalt(15)
        ' Không có thuộc tính LegalIVSizes. Do đó để lấy được mảng byte IV hợp
        ' lệ có thể làm như sau:
        '     crpSym.GenerateIV()
        '     abytIV = crpSym.IV
        '     Debug.WriteLine("Valid abytIV.Length=" & abytIV.Length.ToString)
        ReDim abytIV(15)
    Else
        ReDim abytKey(23)
```

```
        ReDim abytSalt(11)
        ReDim abytIV(7)
      End If
    End Sub

    ' Thủ tục này chép nội dung của file tạm vào file nguồn trong quá trình
    ' mã hóa và giải mã
    Private Sub SwapFiles(ByVal UseFileAccessWait As Boolean)
      If UseFileAccessWait Then
        WaitForExclusiveAccess(strSourceFile)
      End If

      ' Thay thế nội dung file nguồn bằng nội dung file tạm và xóa file tạm
      File.Copy("temp.dat", strSourceFile, True)
      File.Delete("temp.dat")
    End Sub

    ' Thủ tục này dùng để tăng tốc truy xuất đến một file mà vừa mới được đọc.
    ' Nó được sử dụng sau khi giải mã.
    Private Sub WaitForExclusiveAccess(ByVal fullPath As String)
      While (True)
        Try
          Dim fs As FileStream
          fs = New FileStream(fullPath, FileMode.Append, _
            FileAccess.Write, FileShare.None)
          fs.Close()
          Exit Sub
        Catch e As Exception
          Thread.Sleep(300)
        End Try
      End While
    End Sub

End Class
```

frmMain.vb

```
Option Strict On

Imports System.IO
```

```
Imports System.Text.Encoding
Imports System.Text.RegularExpressions
Imports System.Security.Cryptography

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private crpSample As SampleCrypto
    Private FormHasLoaded As Boolean = False
    Private strCurrentKeyFile As String
    Private strSourcePath As String
    Private strRijndaelSaltIVFile As String
    Private strTripleDESSaltIVFile As String

    ' Xử lý nút "Create Salt / IV Key"
    Private Sub btnCreateKey_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCreateKey.Click
        Try
            If PasswordIsValid() Then
                crpSample.Password = txtPassword.Text
            Else
                Exit Sub
            End If

            If crpSample.CreateSaltIVFile(strCurrentKeyFile) Then
                MsgBox("Salt and IV successfully generated and saved to a .dat " & _
                       vbCrLf & "file in the Visual Studio .NET Solution root folder.", _
                       MsgBoxStyle.Information, Me.Text)
            End If
        Catch exp As Exception
            MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
        End Try
    End Sub

    ' Xử lý cho hai nút "Encrypt" và "Decrypt"
    Private Sub EncryptDecrypt_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnEncrypt.Click, btnDecrypt.Click
        Dim btn As Button = CType(sender, Button)
```

```
    Try
        If chkAdvanced.Checked Then
            If IsValid() Then
                With crpSample
                    .SaltIVFile = strCurrentKeyFile
                    .Password = txtPassword.Text
                End With
            Else
                Exit Sub
            End If
        End If

        crpSample.SourceFileName = strSourcePath

        If btn.Name = "btnEncrypt" Then
            crpSample.EncryptFile()
        Else
            crpSample.DecryptFile()
        End If

        txtCrypto.Text = ReadFileAsString(strSourcePath)

    Catch expCrypto As CryptographicException
        MsgBox("The file could not be decrypted. Make sure you entered " & _
            "the correct password. " & vbCrLf & _
            "This error can also be caused by changing " & _
            "crypto type between encryption and decryption.", _
            MsgBoxStyle.Critical, Me.Text)
    Catch exp As Exception
        MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
    End Try
End Sub

' Xử lý nút "Load"
Private Sub btnLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLoad.Click
    With odlgSourceFile
        .InitialDirectory = "C:\"
        ' File nguồn có thể có bất kỳ kiểu nào. Bộ lọc Filter giới hạn
```

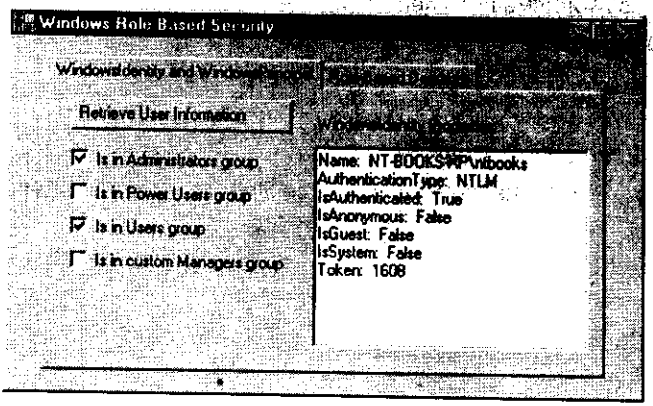
```
        ' Text Format chỉ với mục đích demo
        .Filter = "Text Format (*.txt)|*.txt"
        .FilterIndex = 1

        If .ShowDialog() = DialogResult.OK Then
          Try
            txtCrypto.Text = _
              ReadFileAsString(.FileName)
            strSourcePath = .FileName
          Catch exp As ArgumentException
            MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
          End Try
        End If
      End With
    End Sub

    ' Xử lý sự kiện CheckedChanged cho checkbox chkAdvanced
    Private Sub chkAdvanced_CheckedChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles chkAdvanced.CheckedChanged
      If chkAdvanced.Checked Then
        lblPassword.Enabled = True
        txtPassword.Enabled = True
        btnCreateKey.Enabled = True
      Else
        lblPassword.Enabled = False
        txtPassword.Enabled = False
        btnCreateKey.Enabled = False
      End If
    End Sub

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
      ' Quy định kiểu crypto mặc nhiên
      crpSample = New SampleCrypto("Rijndael")

      ' Quy định đường dẫn để lưu các file khóa là thư mục gốc của solution bằng
      ' cách loại bỏ "bin" trong tên thư mục hiện hành.
      Dim strCurrentDirectory As String = _
        Microsoft.VisualBasic.Left(Environment.CurrentDirectory, _
        Len(Environment.CurrentDirectory) - 3)
```

```
    strRijndaelSaltIVFile = strCurrentDirectory & "RijndaelSaltIV.dat"
    strTripleDESSaltIVFile = strCurrentDirectory & "TripleDESSaltIV.dat"

    ' Quy định file khóa mặc nhiên cho kiểu crypto mặc nhiên
    strCurrentKeyFile = strRijndaelSaltIVFile

    btnEncrypt.Select()

    FormHasLoaded = True
  End Sub

  ' Xử lý sự kiện CheckedChanged cho các radiobutton
  Private Sub RadioButtons_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles optRijndaelAdv.CheckedChanged,
optTripleDESAdv.CheckedChanged
    If FormHasLoaded Then
      If optRijndaelAdv.Checked Then
        crpSample = New SampleCrypto("Rijndael")
        strCurrentKeyFile = strRijndaelSaltIVFile
      Else
        crpSample = New SampleCrypto("TripleDES")
        strCurrentKeyFile = strTripleDESSaltIVFile
      End If
    End If
  End Sub

  ' Kiểm tra dữ liệu nhập trên form
  Private Function IsValid() As Boolean
    If Not PasswordIsValid() Then
      Return False
    End If

    If strSourcePath = "" Then
      MsgBox("You must first load a source file!", MsgBoxStyle.Exclamation,
Me.Text)
      Return False
    End If
    Return True
  End Function
```

```
    ' Kiểm tra password
    Private Function PasswordIsValid() As Boolean
        If Not Regex.IsMatch(txtPassword.Text, "^\s*(\w){8}\s*$") Then
            MsgBox("You must enter an 8-digit password consisting of numbers " & _
                "and/or letters.", MsgBoxStyle.Exclamation, Me.Text)
            Return False
        End If

        Return True
    End Function

    ' Đọc nội dung của một file và chuyển nó sang một chuỗi
    Shared Function ReadFileAsString(ByVal path As String) As String
        Dim fs As New FileStream(path, FileMode.Open, FileAccess.Read,
    FileShare.Read)
        Dim abyt(CInt(fs.Length - 1)) As Byte
        fs.Read(abyt, 0, abyt.Length)
        fs.Close()
        Return UTF8.GetString(abyt)
    End Function
End Class
```

25. Windows Role-Based Security

Chương trình này chỉ cách lấy thông tin login và group của các Windows User. Điều này cho phép ứng dụng cấp phép sử dụng các tính năng của ứng dụng dựa trên định danh (identity) của người sử dụng. Ngoài ra, nó còn chỉ cách tạo role-based security thông qua code.



Các chức năng chính:

- Form chính có một Tab control chứa hai trang. Trang đầu tiên lấy thông tin của người sử dụng đăng nhập Windows để tạo một đối tượng WindowsPrincipal. Các thuộc tính của WindowsIdentity được hiển thị. Trong code sử dụng phương thức IsInRole để đánh dấu các control CheckBox dựa trên nhóm của thành viên.
- Tab thứ hai chứa các Button thực hiện code mở một form đã được bảo vệ dựa trên các yêu cầu an toàn. Nếu user là thành viên của Windows group, form được bảo vệ sẽ mở ra. Nếu security exception xảy ra tương ứng với user không là thành viên của group, một thông báo sẽ hiển thị.

Khi nhấn nút " Retrieve User Information", chương trình kiểm tra user hiện đang sử dụng có là thành viên của bốn nhóm:

- Administrators
- Power Users
- Users
- Managers

Group Managers là một nhóm do người sử dụng tạo. Nếu user không là thành viên của nhóm được yêu cầu, checkbox bên cạnh tên nhóm sẽ không được đánh dấu. Ngoài ra nếu group Managers không tồn tại, checkbox cũng không được đánh dấu.

Nếu có sự thay đổi thành viên nhóm trong Windows để kiểm tra các tính năng của chương trình (ví dụ như tạo nhóm Managers và thêm user vào nó), thì cần phải log off và log on lại để các thay đổi có tác dụng.

Project: **Windows Role-Based Sercurity**

frmMain.vb

```
Option Strict On
Imports System.Security.Principal
Imports System.Security.Permissions

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

    Private idWindows As WindowsIdentity
    Private prinWindows As WindowsPrincipal
```

```
' Lấy tên máy tính
Private machName As String = System.Environment.MachineName

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ' Quy định PrincipalPolicy là WindowsPrincipal để ánh xạ các group của
    ' hệ điều hành vào các role.
    ' Dòng lệnh này cần có cho các yêu cầu (Demand) role-based security
    AppDomain.CurrentDomain.SetPrincipalPolicy(PrincipalPolicy.WindowsPrincipal)
End Sub

Private Sub btnAdminRun_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdminRun.Click
    ' Xóa nội dung TextBox
    Me.txtDisplay.Text = String.Empty

    ' Đối tượng PrincipalPermission cho phép kiểm tra security của người sử
    ' dụng hiện hành bằng cách truyền vào tên user và tên group (hoặc role).
    ' Nếu truyền vào Nothing, tất cả thành viên của một role cụ thể được xem xét,
    ' không riêng user nào.
    ' Chú ý là không thể sử dụng WindowsBuiltInRole ở đây – ta phải truyền vào
    ' một chuỗi sử dụng từ khóa BUILTIN và tên Windows group.
    Dim op As New PrincipalPermission(Nothing, "BUILTIN\Administrators")
    Try
        op.Demand()
        Dim frm As New frmProtected()
        frm.Show()
        frm.txtProtected.Text = String.Format( _
        "Demand succeeded.{0}User is a member of the Administrators group.", _
        ControlChars.CrLf)

    Catch ex As System.Security.SecurityException
        ' Khối Catch xử lý các exception nếu ai đó không là thành viên của
        ' nhóm Administrators nhưng cố thực hiện code.
        txtDisplay.Text = String.Format( _
        "Security Exception:{0}{1}{2}Not a member of the Administrators group.", _
        ControlChars.CrLf, ex.Message, ControlChars.CrLf)
    End Try
End Sub
```

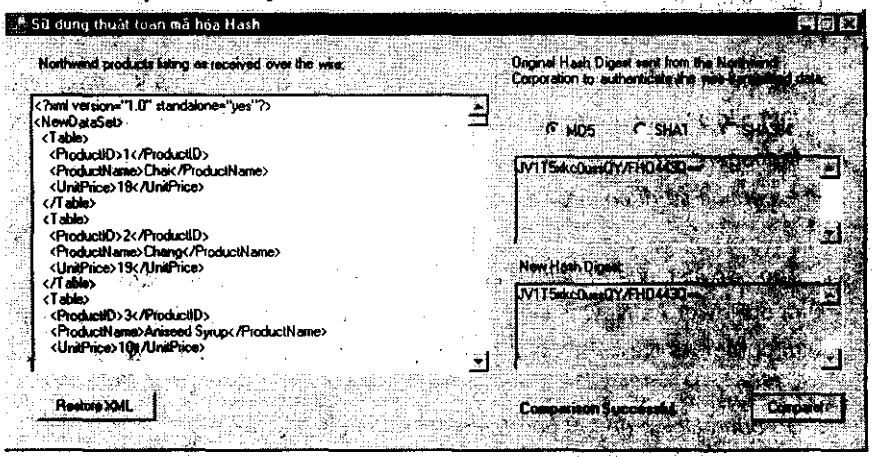
```
Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLogin.Click
    ' Tạo một đối tượng WindowsIdentity cho user hiện hành sử dụng
    ' phương thức GetCurrent
    idWindows = WindowsIdentity.GetCurrent()

    ' Tạo đối tượng WindowsPrincipal dựa trên đối tượng WindowsIdentity.
    ' Đối tượng WindowsPrincipal chứa Group của user hiện hành.
    prinWindows = New WindowsPrincipal(idWindows)

    ' Đánh dấu các CheckBox dựa trên user có thuộc một role cụ thể nào.
    ckAdministrator.Checked = _
     prinWindows.IsInRole(WindowsBuiltInRole.Administrator)

    ckPowerUsers.Checked = _
     prinWindows.IsInRole(WindowsBuiltInRole.PowerUser)

    ckUsers.Checked = _
     prinWindows.IsInRole(WindowsBuiltInRole.User)

    ' Tên máy tính cần có để làm việc với các tên group tự tạo
    ' Nếu group Managers không tồn tại, CheckBox không được đánh dấu
    ' và không xảy ra exception nào
    ckManagers.Checked = _
     prinWindows.IsInRole(machName & "\Managers")

    ' Hiển thị các thuộc tính của WindowsIdentity
    txtLogin.AppendText(String.Format("Name: {0}{1}", _
     idWindows.Name, ControlChars.CrLf))
    txtLogin.AppendText(String.Format("AuthenticationType: {0}{1}", _
     idWindows.AuthenticationType, ControlChars.CrLf))
    txtLogin.AppendText(String.Format("IsAuthenticated: {0}{1}", _
     idWindows.IsAuthenticated, ControlChars.CrLf))
    txtLogin.AppendText(String.Format("IsAnonymous: {0}{1}", _
     idWindows.IsAnonymous, ControlChars.CrLf))
    txtLogin.AppendText(String.Format("IsGuest: {0}{1}", _
     idWindows.IsGuest, ControlChars.CrLf))
    txtLogin.AppendText(String.Format("IsSystem: {0}{1}", _
     idWindows.IsSystem, ControlChars.CrLf))
```

```
    txtLogin.AppendText(String.Format("Token:  {0}{1}", _
     idWindows.Token, ControlChars.CrLf))
    lblProperties.Visible = True
End Sub

Private Sub btnManagerRun_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnManagerRun.Click
    txtDisplay.Text = String.Empty
     ' MachineName là bắt buộc khi sử dụng các group tự tạo. Nếu user không
     ' phải là thành viên của group, hoặc group không tồn tại, một security exception
     ' sẽ được thrown và Demand sẽ dừng.
    Dim op As New PrincipalPermission(Nothing, machName & "\Managers")

    Try
      op.Demand()
      Dim frm As New frmProtected()
      frm.Show()
      frm.txtProtected.Text = String.Format( _
      "Demand succeeded.{0}User is a member of the Managers group.", _
      ControlChars.CrLf)
    Catch ex As System.Security.SecurityException
      txtDisplay.Text = String.Format( _
      "Security Exception:{0}{1}{2}Not a member of the Managers group.", _
      ControlChars.CrLf, ex.Message, ControlChars.CrLf)
    End Try
End Sub

Private Sub btnPowerRun_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPowerRun.Click
    txtDisplay.Text = String.Empty

     ' group Power Users là một group xây dựng sẵn của Windows
    Dim op As New PrincipalPermission(Nothing, "BUILTIN\PowerUsers")
    Try
      op.Demand()
      Dim frm As New frmProtected()
      frm.Show()
      frm.txtProtected.Text = String.Format( _
      "Demand succeeded.{0}User is a member of the Power Users group.", _
      ControlChars.CrLf)
```

```
    Catch ex As System.Security.SecurityException
      txtDisplay.Text = String.Format( _
      "Security Exception:{0}{1}{2}Not a member of the Power Users group.", _
      ControlChars.CrLf, ex.Message, ControlChars.CrLf)
    End Try
  End Sub

  Private Sub btnUnion_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUnion.Click
    txtDisplay.Text = String.Empty

    ' Có thể tạo một Demand để kiểm tra nhiều group bằng cách sử dụng phương
    ' thức Union. Ví dụ này đòi hỏi user phải là thành viên của Power Users hoặc
    ' Managers. Nếu user không là thành viên của cả hai group, một security
    ' exception sẽ được thrown.

    ' Tạo các instance PrincipalPermission cho PowerUsers và Managers
    Dim opPower As New PrincipalPermission(Nothing, "BUILTIN\PowerUsers")
    Dim opMgr As New PrincipalPermission(Nothing, machName & "\Managers")

    ' Sử dụng Union để kết hợp Managers và Power Users
    Try
      opPower.Union(opMgr).Demand()
      Dim frm As New frmProtected()
      frm.Show()
      frm.txtProtected.Text = String.Format( _
      "Demand succeeded.{0}User is a member of the Power Users or the
Managers group.", _
      ControlChars.CrLf)

    Catch ex As System.Security.SecurityException
      txtDisplay.Text = String.Format( _
      "Security Exception:{0}{1}{2}Not a member of Managers or Power Users.", _
      ControlChars.CrLf, ex.Message, ControlChars.CrLf)
    End Try
  End Sub

End Class
```

26. Sử dụng thuật toán mã hóa Hash (băm)

Các giá trị Hash (Hash Value hoặc Hash Digest) là các số duy nhất và cực kỳ cô đọng đại diện cho một phần dữ liệu. Thay đổi dù chỉ một ký tự trong file nguồn có thể làm cho giá trị hash thay đổi hoàn toàn. Hash được sử dụng cho việc bảo đảm tính toàn vẹn của dữ liệu. Ví dụ, bằng cách so sánh giá trị Hash phát sinh trước khi gửi message với giá trị Hash phát sinh sau khi nhận message, người nhận có thể xác định được messege có bị hư (lỗi mất dữ liệu, hoặc sai) hay không. Chương trình này biểu diễn cách sử dụng các chức năng hash với mục đích này.



Các chức năng chính:

Ngữ cảnh: Giả sử công ty Northwind đã gửi một danh sách các sản phẩm và giá cho chúng ta. Dữ liệu được truyền trên một kết nối an toàn, nhưng cả hai bên đều muốn bảo đảm là danh sách không bị lỗi trong quá trình truyền.

- Khi chạy ứng dụng này, một tài liệu XML được tạo và chứa tất cả các sản phẩm lấy từ database SQL Server Northwind và chuyển DataSet vào tài liệu XML. Giả lập này thay cho người gửi tạo tài liệu nguồn. Nội dung của tài liệu này sau đó được đọc vào một StreamReader và chuyển thành một chuỗi để hiển thị trong TextBox. Giả lập này thay cho việc nhận tài liệu được truyền.
- Ngoài ra, khi Form được load, một MD5 hash digest được tạo để sử dụng làm "khóa" gốc để ta có thể kiểm tra tài liệu XML được truyền. Nhấn vào nút Compare! để so sánh hash digest của "tài liệu nhận được" và hash digest gửi từ Northwind. Thay đổi nội dung của TextBox và nhấn Compare! để thấy kết quả.
- Chương trình này sử dụng ba kiểu hash khác nhau: MD5, SHA1, và SHA384.

Project: **Cryptographic Hash Algorithms**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.Security.Cryptography
Imports System.IO
Imports System.Text

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Protected Const COMPARE_INSTRUCTIONS As String = _
        "Click Compare! to generate a hash digest based on the product listing " & _
        "to the left. The original and new hash digests will then be compared " & _
        "to authenticate the displayed product listing."
    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL or MSDE."
    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Private DidPreviouslyConnect As Boolean = False
    Private FormHasLoaded As Boolean = False
    Private hash As Byte()
    Private strConn As String = SQL_CONNECTION_STRING
    Private strOriginalXML As String
```

```
' Xử lý nút "Compare!". Nó so sánh từng byte, hash digest gốc với
' hash digest phát sinh từ TextBox bên trái.
Private Sub btnCompare_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCompare.Click
    ' Tạo một đối tượng Encoding để có thể sử dụng phương thức GetBytes
    ' để tạo các mảng byte
    Dim uEncode As New UnicodeEncoding()

    ' Tạo một mảng byte từ file XML gốc được gửi từ Northwind
    Dim bytHashOriginal As Byte() = uEncode.GetBytes(txtHashOriginal.Text)

    ' Phát sinh một hash digest từ nội dung của TextBox. Nội dung này giả lập cho
    ' XML nhận được từ Northwind thông qua đường truyền. Ta cần so sánh giá trị
    ' hash mới với giá trị hash gốc để bảo đảm XML không bị lỗi trong
    ' quá trình truyền.
    Dim strHashForCompare As String = GenerateHashDigest(txtXML.Text)
    ' Từ hash digest mới, tạo một mảng byte để so sánh với mảng byte chứa
    ' hash digest gốc.
    Dim bytHashForCompare As Byte() = uEncode.GetBytes(strHashForCompare)
    ' Hiển thị hash digest mới trong TextBox
    txtHashForCompare.Text = strHashForCompare

    ' Duyệt qua tất cả các byte trong các giá trị hash
    Dim i As Integer
    For i = 0 To bytHashOriginal.Length - 1
        ' So sánh từng byte, nếu có bất kỳ giá trị không bằng nhau thì hiển
        ' thị một thông báo và thoát vòng lặp.
        If bytHashOriginal(i) <> bytHashForCompare(i) Then
            lblResults.Text = "Data has been corrupted!"
            Exit For
        Else

            ' Tất cả các byte bằng nhau do đó XML được truyền thành công.
            lblResults.Text = "Comparison Successful."
        End If
    Next i

End Sub
```

```
' Xử lý nút "Restore XML". Giúp nhanh chóng có được XML gốc không bị lỗi.
Private Sub btnRestore_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRestore.Click
    txtXML.Text = strOriginalXML
End Sub

' Thủ tục này gọi một thủ tục khác để tạo tài liệu XML gốc, đọc nội dung XML, hiển
' thị nó trong TextBox lớn, sau đó gọi một hàm phát sinh MD5 hash digest để
' hiển thị trong TextBox ở phía trên.
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    ' Tạo tài liệu XML giả lập cho danh sách sản phẩm gốc lấy từ Northwind
    CreateOriginalProductsList()

    ' Mở tài liệu XML và chuyển thành một chuỗi. Giả lập cho danh sách
    ' sản phẩm nhận được
    Dim sr As New StreamReader("..\products.xml")
    strOriginalXML = sr.ReadToEnd
    sr.Close()

    ' Hiển thị XML nhận được và hash digest được sử dụng để kiểm tra XML đã nhận.
    ' Hash digest này phát sinh từ nội dung gốc, không phải từ XML hiển thị
    ' trong TextBox
    txtXML.Text = strOriginalXML
    txtHashOriginal.Text = GenerateHashDigest(strOriginalXML)
    txtHashForCompare.Text = COMPARE_INSTRUCTIONS

    ' Gọi Select() để chuyển tiêu điểm (focus) đến nút "Compare!" tránh việc
    ' XML trong TextBox được tự động chọn.
    btnCompare.Select()
    ' Giá trị này sử dụng để tránh code trong thủ tục sự kiện CheckedChanged
    ' chạy trước khi form được load.
    FormHasLoaded = True

End Sub
```

```
    ' Thủ tục này xử lý CheckedChanged cho ba RadioButton. Thủ tục này phát sinh
    ' một hash digest mới dựa trên kiểu được chọn và khởi tạo lại giao diện (UI).
    ' Code được đặt trong kiểm tra FormHasLoaded để bảo đảm nó không chạy
    ' trước khi form được load
    Private Sub RadioButtons_CheckedChanged(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles optSHA1.CheckedChanged,
    optMD5.CheckedChanged, optSHA384.CheckedChanged

        If FormHasLoaded Then
            txtHashOriginal.Text = GenerateHashDigest(strOriginalXML)
            txtHashForCompare.Text = COMPARE_INSTRUCTIONS
            lblResults.Text = ""
            btnCompare.Focus()
        End If
    End Sub

    ' Thủ tục này tạo tài liệu XML sử dụng cho so sánh hash
    Sub CreateOriginalProductsList()
        ' Hiển thị thông báo cho biết đang thực hiện kết nối
        Dim frmStatusMessage As New frmStatus()
        If Not DidPreviouslyConnect Then
            frmStatusMessage.Show("Connecting to SQL Server")
        End If

        ' Thực hiện kết nối
        Dim IsConnecting As Boolean = True
        While IsConnecting
            Try
                Dim scnnNW As New SqlConnection(strConn)

                Dim strSQL As String = _
                    "SELECT ProductID, ProductName, UnitPrice " & _
                    "FROM Products"

                ' Đối tượng SqlCommand sử dụng để thực hiện các lệnh SQL
                Dim scmd As New SqlCommand(strSQL, scnnNW)

                ' SqlDataAdapter sử dụng đối tượng SqlCommand để đưa dữ liệu vào DataSet
                Dim sda As New SqlDataAdapter(scmd)
```

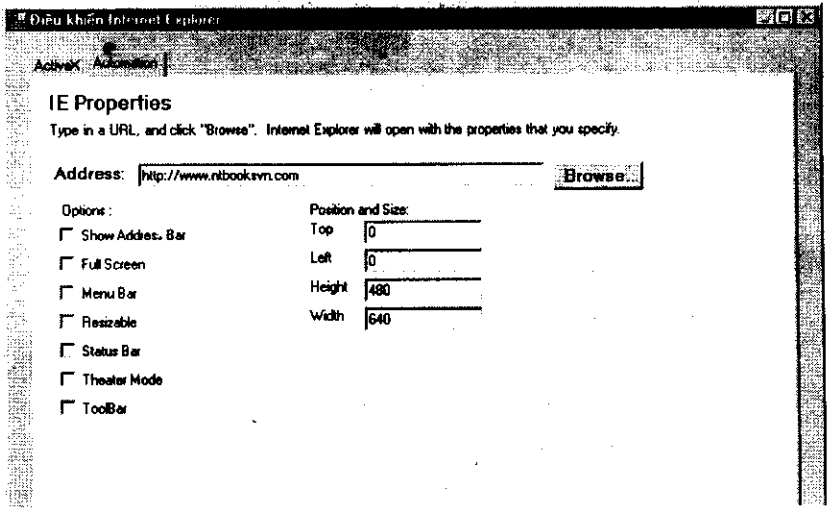
```
            ' Tạo một DataSet mới và đưa dữ liệu vào DataTable đầu tiên của nó
            Dim dsProducts As New DataSet()
            sda.Fill(dsProducts)

            ' Chuyển DataSet sang XML
            Try
                dsProducts.WriteXml("..\products.xml")
            Catch exp As Exception
                MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
            End Try

            ' Dữ liệu đã lấy xong, thoát vòng lặp và đóng form trạng thái
            IsConnecting = False
            DidPreviouslyConnect = True
            frmStatusMessage.Close()

        Catch expSql As SqlException
            MsgBox(expSql.Message, MsgBoxStyle.Critical, Me.Text)
            Exit Sub

        Catch exp As Exception
            If strConn = SQL_CONNECTION_STRING Then
                ' Không thể kết nối vào SQL Server. Chuyển qua MSDE
                strConn = MSDE_CONNECTION_STRING
                frmStatusMessage.Show("Connecting to MSDE")
            Else
                ' Không thể kết nối vào SQL Server hoặc MSDE
                frmStatusMessage.Close()
                MsgBox(CONNECTION_ERROR_MSG, MsgBoxStyle.Critical, Me.Text)
                End
            End If
        End Try

    End While

End Sub
```

```
    ' Hàm này thực hiện công việc mã hóa, phát sinh các kiểu hash digest
    Function GenerateHashDigest(ByVal strSource As String) As String
        Dim uEncode As New UnicodeEncoding()
        Dim bytProducts() As Byte = uEncode.GetBytes(strSource)
        If optMD5.Checked Then
            Dim md5 As New MD5CryptoServiceProvider()
            hash = md5.ComputeHash(bytProducts)
        ElseIf optSHA1.Checked Then
            Dim sha1 As New SHA1CryptoServiceProvider()
            hash = sha1.ComputeHash(bytProducts)
        Else
            Dim sha384 As New SHA384Managed()
            hash = sha384.ComputeHash(bytProducts)
        End If

        ' Base64 là một phương thức mã  hóa dữ liệu nhị phân sang dạng văn bản ASCII
        Return Convert.ToBase64String(hash)
    End Function
End Class
```

Phần III - Kết hợp nhiều phần

27. Điều khiển Internet Explorer

Chương trình này biểu diễn cách điều khiển Internet Explorer.



Các chức năng chính:

- Chương trình này chỉ cách sử dụng Internet Explorer làm một ActiveX Control.
- Cách điều khiển Internet Explorer và quy định nhiều thuộc tính như kích thước, các thanh công cụ,...

Các control chính:

Loại	Thuộc tính	Giá trị
AxWebBrowser	(Name)	ieForHosting

Project: **Điều khiển IE**

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Protected htmlDocActiveX, htmlDocAutomation As mshtml.IHTMLDocument2
    Protected WithEvents ieForAutomation As SHDocVw.InternetExplorer
    Protected ieElement As mshtml.IHTMLElement
    Protected frmStatus As frmStatus

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        ' ieForHosting là một ActiveX control. Đối tượng AxSHDocVw.WebBrowser
        ' được sử dụng khi muốn chạy IE như là một control nhúng vào ứng dụng.
        ieForHosting.Navigate("www.ntbooksvn.com")
        EnableNavIfHistory()
    End Sub

    Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowse.Click
        ieForHosting.Navigate(tbAddress.Text)
    End Sub

    Private Sub btnBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBack.Click
        htmlDocActiveX.parentWindow.history.go(-1)
        tbAddress.Text = htmlDocActiveX.url
    End Sub
```

```
Private Sub btnForward_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    htmlDocActiveX.parentWindow.history.go(+1)
    tbAddress.Text = htmlDocActiveX.url
End Sub

Private Sub btnBrowse2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowse2.Click
    ' Mở Internet Explorer, và cấu hình giao diện dựa trên các option được chọn
    If ieForAutomation Is Nothing Then
        ieForAutomation = New SHDocVw.InternetExplorer()
    End If

    ieForAutomation.AddressBar = chkAddressBar.Checked
    ieForAutomation.FullScreen = chkFullScreen.Checked
    ieForAutomation.MenuBar = chkMenuBar.Checked
    ieForAutomation.Resizable = chkResizable.Checked
    ieForAutomation.StatusBar = chkStatusBar.Checked
    ieForAutomation.TheaterMode = chkTheaterMode.Checked
    ieForAutomation.ToolBar = Convert.ToInt32(chkToolBar.Checked)

    If IsNumeric(txtHeight.Text) Then
        ieForAutomation.Height = Int32.Parse(txtHeight.Text)
    End If

    If IsNumeric(txtWidth.Text) Then
        ieForAutomation.Width = Int32.Parse(txtWidth.Text)
    End If

    If IsNumeric(txtTop.Text) Then
        ieForAutomation.Top = Int32.Parse(txtTop.Text)
    End If

    If IsNumeric(txtLeft.Text) Then
        ieForAutomation.Left = Int32.Parse(txtLeft.Text)
    End If

    Me.Cursor = Cursors.WaitCursor
```

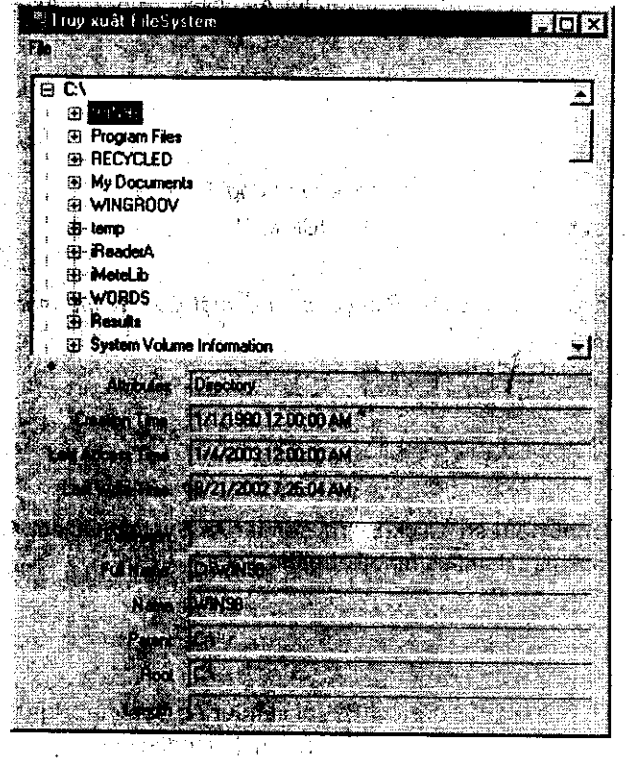
```
    frmStatus = New frmStatus()
    frmStatus.Show("Connecting to Web page and processing HTML. Please stand
by...")

    ieForAutomation.Visible = True
    ieForAutomation.Navigate(tbAddress2.Text)
  End Sub

  Private Sub chkAddressBar_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkAddressBar.CheckedChanged
      ' Thanh Address chỉ hiển thị khi cho phép Toolbar
    If chkAddressBar.Checked = True AndAlso _
      chkToolBar.Checked = False Then

      If MessageBox.Show("You must also enable the toolbar for the " & _
        "address bar to be visible.  Do you want to enable the " & _
        "tool bar?", _
        "Enable ToolBar?", _
        MessageBoxButtons.YesNo, _
        MessageBoxIcon.Question) = DialogResult.Yes Then

        chkToolBar.Checked = True

      End If
    End If
  End Sub

  ' Xử lý khi trang Web đã load xong
  Private Sub ieForAutomation_DocumentComplete(ByVal pDisp As Object, ByRef
URL As Object) Handles ieForAutomation.DocumentComplete

  Me.Invoke(New HideStatusDelegate(AddressOf Me.HideStatus))
  End Sub

  Private Sub ieForAutomation_OnQuit() Handles ieForAutomation.OnQuit

    ieForAutomation = Nothing
  End Sub
```

```
    ' Xử lý khi trang Web đã load xong. Nếu không thực hiện gán htmlDocActiveX ở
    ' vị trí khác trong chương trình thì sẽ nhận được một tham chiếu null nếu tài liệu
    ' chưa load xong..
    Private Sub ieForHosting_DocumentComplete(ByVal sender As System.Object,
    ByVal e As AxSHDocVw.DWebBrowserEvents2_DocumentCompleteEvent)
    Handles ieForHosting.DocumentComplete
        htmlDocActiveX = CType(ieForHosting.Document, mshtml.IHTMLDocument2)
        Me.Invoke(New OnCompleteDelegate(AddressOf Me.OnComplete), _
            New Object() {htmlDocActiveX})
    End Sub

    Protected Sub EnableNavIfHistory()
        ' Nếu chưa có History, không cho sử dụng các nút Forward và Back
        If htmlDocActiveX Is Nothing OrElse
    htmlDocActiveX.parentWindow.history.length = 0 Then
            btnBack.Enabled = False
            btnForward.Enabled = False
        Else
            btnBack.Enabled = True
            btnForward.Enabled = True
        End If
    End Sub

    ' Ẩn form trạng thái khi load xong trang web
    Delegate Sub HideStatusDelegate()
    Protected Sub HideStatus()
        frmStatus.Hide()
        Me.Cursor = Cursors.Default
    End Sub

    ' Xử lý khi ActiveX control hoàn tất việc load tài liệu
    Delegate Sub OnCompleteDelegate(ByVal htmlDoc As mshtml.IHTMLDocument2)
    Public Sub OnComplete(ByVal htmlDoc As mshtml.IHTMLDocument2)
        tbAddress.Text = htmlDoc.url
        EnableNavIfHistory()
    End Sub
End Class
```

Phần IV - File

28. Truy xuất File System

Biểu diễn sử dụng các class DirectoryInfo/Directory và FileInfo/File.



Các chức năng chính:

Chương trình này cho phép tìm hiểu sâu hơn một control TreeView chứa thông tin thư mục và file system, chọn các thư mục hoặc các file và xem thông tin về các đối tượng đã chọn. Cụ thể là cho thấy một số phương thức chia sẻ (shared) của các class DirectoryInfo và FileInfo, cùng với các phương thức của instance của các class này. Ngoài ra, chương trình còn sử dụng tính thừa kế class FileSystemInfo của hai đối tượng DirectoryInfo và FileInfo, sử dụng cùng một thủ tục để hiển thị thông tin cho cả hai loại đối tượng.

Project: **File System**

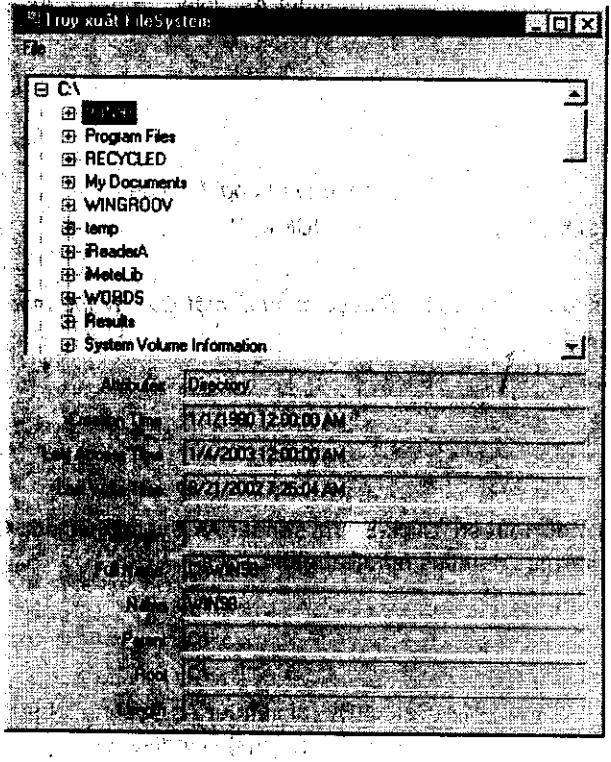
frmMain.vb

- Option Strict On
- Imports System.IO

Phần IV - File

28. Truy xuất File System

Biểu diễn sử dụng các class DirectoryInfo/Directory và FileInfo/File.



Các chức năng chính:

Chương trình này cho phép tìm hiểu sâu hơn một control TreeView chứa thông tin thư mục và file system, chọn các thư mục hoặc các file và xem thông tin về các đối tượng đã chọn. Cụ thể là cho thấy một số phương thức chia sẻ (shared) của các class DirectoryInfo và FileInfo, cùng với các phương thức của instance của các class này. Ngoài ra, chương trình còn sử dụng tính thừa kế class FileSystemInfo của hai đối tượng DirectoryInfo và FileInfo, sử dụng cùng một thủ tục để hiển thị thông tin cho cả hai loại đối tượng.

Project: **File System**

frmMain.vb

```
Option Strict On
Imports System.IO
```

```
Public Class frmMain
    Inherits System.Windows.Forms.Form

'Code phát sinh form được bỏ qua

    Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        LoadTreeView()
    End Sub

    ' Sử dụng văn bản này cho các node giả trong TreeView.
    Private Const DUMMY As String = "DUMMY"

    ' Dùng để theo dõi một phần tử được chọn là một file hay thư mục
    Private Enum ItemType
        Directory = 1
        File = 2
    End Enum

    Private Sub mnuRefresh_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuRefresh.Click
        LoadTreeView()
    End Sub

    Private Sub tvwRoot_AfterSelect(ByVal sender As System.Object, ByVal e As
    System.Windows.Forms.TreeViewEventArgs) Handles tvwRoot.AfterSelect

        Try
            With e.Node
                Select Case .Tag
                    Case ItemType.File
                        ' Đối tượng FileInfo không hỗ trợ các thuộc tính Parent và Root
                        lblParent.Text = String.Empty
                        lblRoot.Text = String.Empty

                        ' Disable/Enable các label.
                        lblParentLabel.Enabled = False
                        lblRootLabel.Enabled = False
                        lblLengthLabel.Enabled = True
```

```
            ' Lấy một đối tượng FileInfo tương ứng với file được chọn,
            ' sau đó hiển thị các thuộc tính của nó
            Dim fi As New FileInfo(.FullPath)
            lblLength.Text = fi.Length.ToString
            DisplayFSIProperties(fi)

        Case ItemType.Directory
            ' Đối tượng DirectoryInfo không có thuộc tính Length
            lblLength.Text = String.Empty

            ' Disable/Enable các label.
            lblLengthLabel.Enabled = False
            lblParentLabel.Enabled = True
            lblRootLabel.Enabled = True

            ' Lấy đối tượng DirectoryInfo  tương ứng với thư mục được chọn,
            ' sau đó hiển thị các thuộc tính của nó.
            Dim di As New DirectoryInfo(.FullPath)
            lblParent.Text = di.Parent.Name
            lblRoot.Text = di.Root.Name
            DisplayFSIProperties(di)
        Case Else
            ' Xóa các label
            ClearProperties()
      End Select
    End With
  Catch exp As Exception
    MessageBox.Show(exp.Message, Me.Text)
  End Try
End Sub

Private Sub tvwRoot_BeforeExpand(ByVal sender As Object, ByVal e As
System.Windows.Forms.TreeViewCancelEventArgs) Handles
tvwRoot.BeforeExpand

  ' Trước khi mở một node, loại bỏ các node con đã có. Hàm này giả lập cách
  ' làm việc của Windows Explorer: nếu có các file hoặc thư mục trong đĩa thì
  ' TreeView hiển thị chúng. Nếu không, chỉ đơn giản làm biến mất dấu + khi
  ' thực hiện mở node.
```

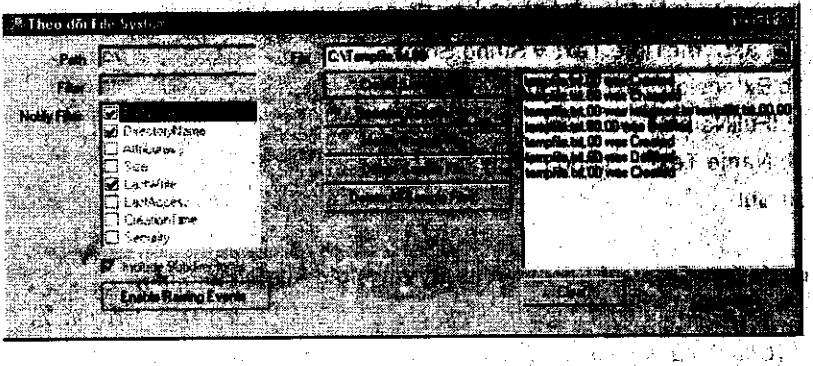
```
    Try
      e.Node.Nodes.Clear()
       ' Lấy các tên thư mục và các tên file
      AddFolders(e.Node)
      AddFiles(e.Node)

    Catch exp As IOException
       ' Có thể đĩa chưa sẵn sàng. Chỉ bỏ qua lỗi.
    Catch exp As Exception
       ' Nếu lỗi khác xảy ra, hiển thị thông báo
      MessageBox.Show(exp.ToString, Me.Text)
    End Try
  End Sub

  Private Sub AddFiles(ByVal nod As TreeNode)
    Dim strPath As String = nod.FullPath
    Dim strFile As String

    With nod
      For Each strFile In Directory.GetFiles(strPath)
        With nod.Nodes.Add(Path.GetFileName(strFile))
          .Tag = ItemType.File
        End With
      Next
    End With
  End Sub

  Private Sub AddFolders(ByVal nod As TreeNode)
    Dim strPath As String = nod.FullPath
    Dim strDir As String

    With nod
      For Each strDir In Directory.GetDirectories(strPath)
        ' Path.GetFileName chỉ trả về phần tên của đường
        ' dẫn đầy đủ được trả về từ phương thức GetDirectories
        With nod.Nodes.Add(Path.GetFileName(strDir))
          ' Lưu lại phần tử này là một thư mục. Ta cần thông tin này sau này.
          .Tag = ItemType.Directory
```

```
            ' Thêm một node giả để dấu + xuất hiện.
            .Nodes.Add(DUMMY)
        End With
    Next
  End With
End Sub

Private Sub ClearProperties()
  lblParent.Text = String.Empty
  lblRoot.Text = String.Empty
  lblLength.Text = String.Empty
  lblAttributes.Text = String.Empty
  lblCreationTime.Text = String.Empty
  lblLastAccessTime.Text = String.Empty
  lblLastWriteTime.Text = String.Empty
  lblExtension.Text = String.Empty
  lblFullName.Text = String.Empty
  lblName.Text = String.Empty
End Sub

Private Sub DisplayFSIProperties(ByVal fsi As FileSystemInfo)
  ' Hiển thị một số thông tin chung cho cả hai đối tượng DirectoryInfo và FileInfo.
  ' Chú ý, cả hai đối tượng đều thừa kế từ FileSystemInfo
  lblAttributes.Text = fsi.Attributes.ToString
  lblCreationTime.Text = fsi.CreationTime.ToString
  lblLastAccessTime.Text = fsi.LastAccessTime.ToString
  lblLastWriteTime.Text = fsi.LastWriteTime.ToString
  lblExtension.Text = fsi.Extension
  lblFullName.Text = fsi.FullName
  lblName.Text = fsi.Name
End Sub

Private Sub LoadTreeView()
  Dim strDrive As String
  ' Duyệt qua tất cả driver có trên máy tính, tạo mỗi đĩa một node. Ngoài ra,
  ' thêm một node giả vào các node đã tồn tại để xuất hiện dấu +
  ' Directory.GetLogicalDrives trả về một mảng chuỗi, mỗi phần tử tương
  ' ứng cho một đĩa trên máy tính.
  tvwRoot.Nodes.Clear()
```

```
        For Each strDrive In Directory.GetLogicalDrives()
            With tvwRoot.Nodes.Add(strDrive)
                .Nodes.Add(DUMMY)
            End With
        Next
    End Sub
End Class
```

29. Theo dõi File System

.NET Framework làm cho việc theo dõi các thay đổi trên File System dễ dàng hơn, class FileSystemWatcher có thể phát sinh các sự kiện khi ta tạo, xóa, thay đổi, hoặc đổi tên các file trong các thư mục mà chương trình đang theo dõi.



Các chức năng chính:

Chương trình biểu diễn cách sử dụng một số thuộc tính của FileSystemWatcher:

Path Đường dẫn sẽ được theo dõi.

Filter Các kiểu file sẽ được theo dõi. Mặc nhiên là *.*.

NotifyFilter Các kiểu thay đổi sẽ được theo dõi. Thuộc tính này có thể là 0 hoặc là sư kết hợp nhiều giá trị của System.IO.NotifyFilters. Mặc nhiên, FileSystemWatcher theo dõi các thay đổi FileName, DirectoryName, và LastWrite.

IncludeSubdirectories Cho biết có theo dõi các thư mục con của đường dẫn được chọn.

EnableRaisingEvents Nếu là True, đối tượng theo dõi các thay đổi trên file system và phát sinh các sự kiện khi có sự thay đổi. Trong chương trình này, thuộc tính này được điều khiển bằng một CheckBox được hiển thị như là mọt nút bật/tắt (thuộc tính Appearance).

Nhấn nút **Enable Raising Events** để bật tính năng theo dõi cho đối tượng ***FileSystemWatcher***. *Các thủ tục sự kiện trong form sẽ lần lượt ghi các thông* báo vào ListBox khi thay đổi các file trong đường dẫn được chọn. Để cho việc thử nghiệm dễ dàng hơn, có thể sử dụng các nút **Create/Rename/Modify/Delete Sample File** trên form. Các nút này tạo, thay đổi tên file, thay đổi nội dung, và xóa các file tạm thời trong thư mục được chọn. Dùng nút **Delete All Sample Files** để xóa tất cả các file tạo ra trong quá trình thử.

Người sử dụng cũng có thể dùng Windows Explorer hoặc các ứng dụng khác để tạo, thay đổi tên, thay đổi nội dung, và xóa các file. Chương trình này vẫn hiển thị các thông báo tương ứng, bởi vì **FileSystemWatcher** theo dõi các thay đổi mà không quan tâm đến chương trình thay đổi.

Các control chính:

Loại	Thuộc tính	Giá trị
FileSystemWatcher	(Name)	fsw
CheckBox	(Name)	chkEvents
	Appearance	Button

Project: Theo dõi **File System**

frmMain.vb

```
Option Strict On
Imports System.IO
Imports System.Enum

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    ' Khi chương trình tạo file tạm, nó sử dụng tên này làm tên gốc
    Private Const FILE_ROOT As String = "Tempfile.txt"

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnClear.Click

        ' Xóa danh sách các sự kiện trên các file
        lstEvents.Items.Clear()
    End Sub
```

```
Private Sub btnCreate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCreate.Click
    Try
        Dim strFile As String = GetUniqueFileName(fsw.Path, FILE_ROOT)

        ' Tạo một file và đóng lại ngay lập tức
        File.Create(strFile).Close()

        ' Thêm file mới vào combobox
        cboSampleFile.Items.Add(strFile)
        cboSampleFile.Text = strFile

        ' Enable và Disable các button
        HandleItems()

    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
    End Try
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
    Dim strName As String = cboSampleFile.Text

    ' Nếu file vẫn còn tồn tại, xóa nó, và cập nhật lại combobox
    If CheckFile(strName) Then
        Try
            File.Delete(strName)
            RemoveFileName(strName)
            HandleItems()

        Catch exp As Exception
            MessageBox.Show(exp.Message, Me.Text)
        End Try
    End If

End Sub
```

```
Private Sub btnModify_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnModify.Click
    Dim strName As String = cboSampleFile.Text

    ' Nếu file vẫn còn tồn tại, thay đổi nội dung của nó
    If CheckFile(strName) Then
      Try
        Dim fs As New FileStream(cboSampleFile.Text, FileMode.Append)
        Dim sw As New StreamWriter(fs)
        sw.WriteLine("This is more text")
        sw.Close()
        fs.Close()

      Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
      End Try
    End If
  End Sub

  Private Sub btnRename_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRename.Click
    Dim strOld As String
    Dim strNew As String

    ' nếu file vẫn còn tồn tại, đổi tên nó
    strOld = cboSampleFile.Text
    If CheckFile(strOld) Then
      strNew = GetUniqueFileName(Path.GetDirectoryName(strOld), _
                                 Path.GetFileName(strOld))
      Try
        File.Move(strOld, strNew)
        cboSampleFile.Items.Remove(strOld)
        cboSampleFile.Items.Add(strNew)
        cboSampleFile.Text = strNew

      Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
      End Try
    End If
  End Sub
```

```
Private Sub cboSampleFile_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboSampleFile.SelectedIndexChanged

    ' Kiểm tra xem file có còn tồn tại. Nếu không còn, loại tên file khỏi danh sách
    CheckFile(cboSampleFile.Text)
    HandleItems()
End Sub

Private Sub chkEvents_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles chkEvents.CheckedChanged
    Dim blnIsRunning As Boolean

    If chkEvents.Checked Then
        Try
            GatherFSWProperties(fsw)

        Catch exp As Exception
            MessageBox.Show(exp.Message, Me.Text)
            chkEvents.Checked = False
        End Try
    End If

    ' Đối tượng FileSystemWatcher có theo dõi các thay đổi hay không?
    blnIsRunning = chkEvents.Checked
    fsw.EnableRaisingEvents = blnIsRunning

    ' Nếu có theo dõi, disable các control chọn thuộc tính
    ' Nếu không, enable các control
    txtPath.Enabled = Not blnIsRunning
    txtFilter.Enabled = Not blnIsRunning
    chkIncludeSubdirectories.Enabled = Not blnIsRunning
    clstNotifyFilter.Enabled = Not blnIsRunning
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ' Cho các tên trong Enum vào CheckedListBox. Có thể kết data source
    ' vào kết quả trả về từ phương thức GetValues.
    ' Phương thức GetValues có trong namespace System.Enum. Nó đòi hỏi
```

```
        ' một kiểu enum, và trả về một mảng chứa giá trị tương ứng với các phần
        ' tử trong enum. Dòng code này thêm danh sách các giá trị NotifyFilter vào
        ' control CheckedListBox
        clstNotifyFilter.DataSource = GetValues(GetType(NotifyFilters))
        ' Hiển thị tất cả các thuộc tính của đối tượng FileSystemWatcher
        DisplayFSWProperties(fsw)
    End Sub

    Private Sub fsw_Error(ByVal sender As Object, ByVal e As
    System.IO.ErrorEventArgs) Handles fsw.Error
        Dim exp As Exception = e.GetException
        AddItem(exp.Message)
    End Sub

    Private Sub fsw_Renamed(ByVal sender As Object, ByVal e As
    System.IO.RenamedEventArgs) Handles fsw.Renamed
        Dim strText As String = String.Format("{0} was renamed to {1}", _
                    e.OldName, e.Name)
        AddItem(strText)
    End Sub

    Private Sub AddItem(ByVal strText As String)
        lstEvents.Items.Add(strText)
    End Sub

    Private Function CheckFile(ByVal strFileName As String) As Boolean
        ' Kiểm tra xem một file có còn tồn tại hay không. Nếu không, thông báo nó
        ' đã bị xóa và cập nhật lại danh sách các file.
        If File.Exists(strFileName) Then
            Return True
        Else
            MessageBox.Show("The selected file has been deleted. It will be removed
    from this list as well.")
            RemoveFileName(strFileName)
            HandleItems()
            Return False
        End If
    End Function
```

```
Private Sub DisplayFSWProperties(ByVal fsw As FileSystemWatcher)
    SetChecks(fsw.NotifyFilter, clstNotifyFilter)
    txtPath.Text = fsw.Path
    txtFilter.Text = fsw.Filter
    chkEvents.Checked = fsw.EnableRaisingEvents
    chkIncludeSubdirectories.Checked = fsw.IncludeSubdirectories
End Sub

Private Sub GatherFSWProperties(ByVal fsw As FileSystemWatcher)

    ' Đoạn code này có thể gây lỗi, nhưng thủ tục gọi nó sẽ bắt lỗi.
    fsw.Path = txtPath.Text
    fsw.Filter = txtFilter.Text
    fsw.IncludeSubdirectories = chkIncludeSubdirectories.Checked
    fsw.NotifyFilter= CType(GetChecks(clstNotifyFilter), NotifyFilters)
End Sub

Private Function GetChecks(ByVal clst As CheckedListBox) As Integer
    ' Duyệt qua tất cả các phần tử trong CheckedListBox, và phát sinh một số
    ' nguyên tương ứng cho tất cả các bit được chọn. Mỗi phần tử trong
    ' CheckedListBox thực sự là một số nguyên (trong một enum), do đó chỉ cần
    ' chuyển các phần tử sang số nguyên, sau đó OR với kết quả.

    Dim intValue As Integer
    Dim obj As Object

    Try
        For Each obj In clst.CheckedItems
            intValue = intValue Or CInt(obj)
        Next
        Return intValue
    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
    End Try
End Function
```

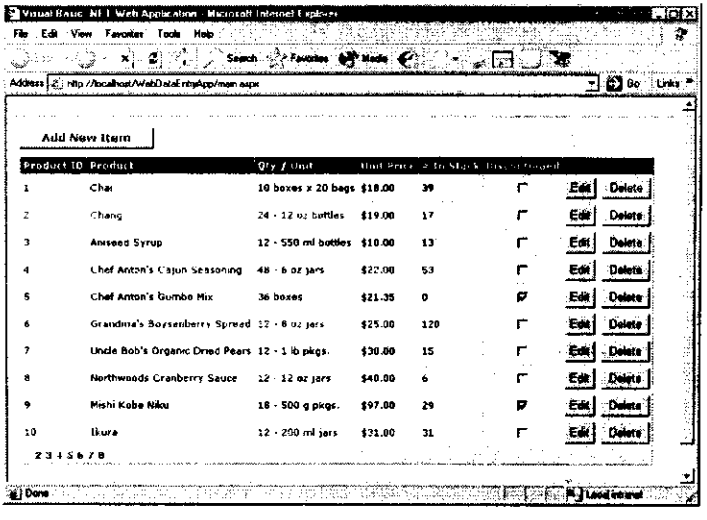
```
Private Function GetUniqueFileName(ByVal Path As String, ByVal Root As String)
As String
    ' Phát sinh một tên file không trùng, sử dụng Root làm tên file gốc, trong một
    ' đường dẫn cụ thể. Ví dụ đường dẫn là C:\ và Root là TempFile.txt, hàm này sẽ
    ' trả về C:\TempFile.txt.01, tăng giá trị cho đến khi tên file chưa tồn tại.
    Dim strTemp As String
    Dim i As Integer

    Path = Path.TrimEnd("\ ".ToCharArray)
    Root = Root.TrimEnd(". ".ToCharArray)

    ' Lặp, tăng biến đếm, thực hiện cho đến khi tên file chưa tồn tại
    Do
      strTemp = String.Format("{0}\{1}.{2:00}", Path, Root, i)
      i += 1
    Loop While File.Exists(strTemp) And i < 100
    Return strTemp
End Function

Private Sub HandleChangedCreatedDeleted(ByVal sender As Object, ByVal e As
System.IO.FileSystemEventArgs) Handles fsw.Changed, fsw.Created, fsw.Deleted
    ' Thủ tục này có thể xử lý các sự kiện Changed, Created, và Deleted bởi vì tất cả
    ' chúng có cùng các tham số.
    Dim strText As String = String.Format("{0} was {1}", e.Name, e.ChangeType)
    AddItem(strText)
End Sub

Private Sub HandleItems()
    ' Enable hoặc Disable các button
    Dim blnEnable As Boolean = (cboSampleFile.Text <> String.Empty)
    btnModify.Enabled = blnEnable
    btnDelete.Enabled = blnEnable
    btnRename.Enabled = blnEnable
End Sub
```

```
    Private Sub RemoveFileName(ByVal strFileName As String)
        ' Loại bỏ một tên file cụ thể ra khỏi combobox, nếu còn các tên file khác,
        ' chọn tên file đầu tiên.
        cboSampleFile.Items.Remove(strFileName)
        If cboSampleFile.Items.Count > 0 Then
            cboSampleFile.Text = cboSampleFile.Items(0).ToString
        End If
    End Sub

    Private Sub SetChecks(ByVal Value As Integer, ByVal clst As CheckedListBox)
        Dim intValue As Integer
        Dim i As Integer
        Dim blnCheck As Boolean

        ' Duyệt qua mỗi phần tử trong CheckedListBox. Với mỗi phần tử,
        ' nếu bit tương ứng có giá trị, đánh dấu phần tử này trong danh sách.
        For i = 0 To clst.Items.Count - 1
            intValue = CInt(clst.Items(i))
            blnCheck = ((intValue And Value) = intValue)
            clst.SetItemChecked(i, blnCheck)
        Next
    End Sub

    Private Sub btnDeleteAll_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnDeleteAll.Click
        ' Xóa tất cả các file trong combobox
        Dim strName As String
        For Each strName In cboSampleFile.Items
            Try
                File.Delete(strName)
            Catch
                ' Nếu không thể xóa file, nghĩa là nó đã được xóa rồi.
            End Try
        Next
        cboSampleFile.Items.Clear()
        HandleItems()
    End Sub

End Class
```

Phần V - Web Development

30. Tạo một ứng dụng Web nhập dữ liệu

Chương trình này biểu diễn cách làm việc với DataGrid trong một ứng dụng nhập liệu trên Web.



Các chức năng chính:

- Hiển thị dữ liệu trong một DataGrid được tùy biến định dạng với các class **BoundColumns**, **TemplateColumns**, và **CSS**.
- Thêm một record mới, thay đổi hoặc xóa record đã tồn tại.
- Application Cache được sử dụng để lưu DataView được kết vào DataGrid. Với cách này, dữ liệu chỉ được download một lần.
- DataGrid vẫn được sắp xếp ngay cả khi có phân trang. Biểu thức sắp xếp được lưu trong một control HTMLInputHidden.
- Các control RequiredFieldValidator và RegularExpressionValidator được sử dụng để bảo đảm dữ liệu hợp lệ.

Chú ý: Chương trình sử dụng database Northwind trong SQL Server.

- Để install chương trình này, chạy file Setup.vbs. Sau đó mở Solution. Nhấn F5 để chạy ứng dụng.
- File setup.vbs giả sử thư mục web mặc nhiên là c:\inetpub\wwwroot. Nếu thư mục web mặc nhiên là thư mục khác thì thay đổi file này để nó chép các file vào thư mục web mặc nhiên.

Các control chính:

Loại	Thuộc tính	Giá trị
WebControls.DataGrid	(Name)	grdProducts
	AllowPaging	True
	AllowSorting	True
	PageSize	10
	DataKeyField	ProductID

Project: **Tao ung dung Web nhap du lieu**

Main.aspx *(download tai http://www.ntbooksvn.com)*

```
<%@Page Language="vb" AutoEventWireup="false" Codebehind="main.aspx.vb"
Inherits="WebDataEntryApp.Main" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>Visual Basic .NET Web Application</title>
    <meta content="Microsoft Visual Studio.NET 7.0" name="GENERATOR">
    <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
    <LINK href="styles.css" rel="stylesheet">
  </HEAD>
  <body>
    <form id="frmMain" method="post" runat="server">
      <hr>
      <TABLE width="100%" border="0">
        <TR>
          <td><asp:button id="btnAddNew" Text="Add New Item"
CausesValidation="False" Runat="server" CssClass="Button"></asp:button></td>
        </TR>
        <TR>
          <TD vAlign="top"><asp:datagrid id="grdProducts" runat="server"
DataKeyField="ProductID" AllowSorting="True" AutoGenerateColumns="False"
AllowPaging="True">
            <SelectedItemStyle CssClass="DataGrid_SelectedItemStyle" />
            <ItemStyle CssClass="DataGrid_ItemStyle" />
            <AlternatingItemStyle CssClass="DataGrid_AlternatingItemStyle" />
            <HeaderStyle CssClass="DataGrid_HeaderStyle" />
            <FooterStyle CssClass="DataGrid_FooterStyle" />
```

```
                <Columns>
                  <asp:BoundColumn DataField="ProductID"
SortExpression="ProductID" HeaderText="Product ID" />
                  <asp:BoundColumn DataField="ProductName"
SortExpression="ProductName" HeaderText="Product" />
                  <asp:BoundColumn DataField="QuantityPerUnit"
SortExpression="QuantityPerUnit" HeaderText="Qty / Unit" />
                  <asp:BoundColumn DataField="UnitPrice"
SortExpression="UnitPrice" HeaderText="Unit Price" DataFormatString="{0:c}" />
                  <asp:BoundColumn DataField="UnitsInStock"
SortExpression="UnitsInStock" HeaderText="# In Stock" />
                  <asp:TemplateColumn SortExpression="Discontinued" ItemStyle-
CssClass="Discontinued" HeaderText="Discontinued">
                    <ItemTemplate>
                      <asp:CheckBox ID="chkDiscontinuedGrid" Runat="server"
Checked='<%# CBool(Container.DataItem("Discontinued")) %>' />
                    </ItemTemplate>
                  </asp:TemplateColumn>
                  <asp:ButtonColumn Text="Edit" ButtonType="PushButton"
CommandName="Select" />
                  <asp:ButtonColumn Text="Delete" ButtonType="PushButton"
CommandName="Select" />
                </Columns>
                <PagerStyle CssClass="DataGrid_PagerStyle"
Mode="NumericPages" />
              </asp:datagrid><br>
              <input id="htmlHiddenSortExpression" type="hidden"
value="ProductID" name="htmlHiddenSortExpression" runat="server">
            </TD>
          </TR>
        </TABLE>
        <span class="Message">
          <%= strMsg %>
        </span><span class="ErrorMessage">
          <%= strErrorMsg %>
        </span>
        <asp:panel id="pnlForm" Runat="server" Visible="False">
          <TABLE id="Table1" width="100%">
            <TR>
              <TD class="FormLabel" width="15%">Product Name:
              </TD>
```

```
            <TD class="FormInput">
              <asp:textbox id="txtProductName" runat="server"
EnableViewState="False" MaxLength="40" Columns="42"></asp:textbox> 
              <asp:RequiredFieldValidator id="rfvProductName" runat="server"
ControlToValidate="txtProductName" Display="Dynamic"
ErrorMessage="Required!"></asp:RequiredFieldValidator></TD>
          </TR>
          <TR>
            <TD class="FormLabel">Qty / Unit:
            </TD>
            <TD class="FormInput">
              <asp:textbox id="txtQtyUnit" runat="server"
EnableViewState="False" MaxLength="20" Columns="22"></asp:textbox> 
              <asp:RequiredFieldValidator id="rfvQtyUnit" runat="server"
ControlToValidate="txtQtyUnit" Display="Dynamic"
ErrorMessage="Required!"></asp:RequiredFieldValidator></TD>
          </TR>
          <TR>
            <TD class="FormLabel">Unit Price:
            </TD>
            <TD class="FormInput">
              <asp:textbox id="txtPrice" runat="server" EnableViewState="False"
MaxLength="8" Columns="10"></asp:textbox> 
              <asp:RequiredFieldValidator id="rfvPrice" runat="server"
ControlToValidate="txtPrice" Display="Dynamic"
ErrorMessage="Required!"></asp:RequiredFieldValidator> 
              <asp:RegularExpressionValidator id="revPrice" runat="server"
ControlToValidate="txtPrice" Display="Dynamic" ErrorMessage="You must enter a
valid price."
ValidationExpression="\d+[.]?[\d]{0,2}"></asp:RegularExpressionValidator></TD>
          </TR>
          <TR>
            <TD class="FormLabel"># In Stock:
            </TD>
            <TD class="FormInput">
              <asp:textbox id="txtInStock" runat="server"
EnableViewState="False" MaxLength="4" Columns="6"></asp:textbox> 
              <asp:RequiredFieldValidator id="rfvInStock" runat="server"
ControlToValidate="txtInStock" Display="Dynamic"
ErrorMessage="Required!"></asp:RequiredFieldValidator> 
```

```
                <asp:RegularExpressionValidator id="revInStock" runat="server"
ControlToValidate="txtInStock" Display="Dynamic" ErrorMessage="You must enter
a number." ValidationExpression="\d+"></asp:RegularExpressionValidator></TD>
            </TR>
            <TR>
              <TD class="FormLabel">Discontinued:
              </TD>
              <TD class="FormCheckBox">
                <asp:checkbox id="chkDiscontinued" runat="server"
CssClass="CheckBox"></asp:checkbox></TD>
            </TR>
            <TR>
              <TD> </TD>
              <TD>
                <asp:Button id="btnSave" Text="Save Changes" Runat="server"
CssClass="Button"></asp:Button></TD>
            </TR>
          </TABLE>
        </asp:panel></form>
    </body>
</HTML>
```

Main.aspx.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class Main
    Inherits System.Web.UI.Page
    Protected WithEvents grdProducts As DataGrid
    Protected WithEvents txtProductName, txtQtyUnit, txtPrice, txtInStock As TextBox
    Protected WithEvents chkDiscontinued As CheckBox
    Protected WithEvents htmlHiddenSortExpression As HtmlInputHidden
    Protected WithEvents pnlForm As Panel
    Protected WithEvents rfvQtyUnit, rfvPrice, rfvInStock, rfvProductName As
RequiredFieldValidator
    Protected WithEvents revPrice, revInStock As RegularExpressionValidator
    Protected WithEvents btnAddNew As System.Web.UI.WebControls.Button
    Protected WithEvents btnSave As System.Web.UI.WebControls.Button
    Protected WithEvents lblTitle As System.Web.UI.WebControls.Label
```

```
' Code phát sinh Web form được bỏ qua

  Private dvProducts As DataView
  Public strMsg As String
  Public strErrorMsg As String

  Protected Const SQL_CONNECTION_STRING As String = "Persist Security
Info=False;User ID=sa;Password=;Initial Catalog=Northwind;Data
Source=localhost"
  ' Xử lý khi nhấn nút "Add New Item"
  Private Sub btnAddNew_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnAddNew.Click
      ' Xóa các giá trị đã có hoặc hiển thị form soạn thảo nếu nó đang được ẩn
    If pnlForm.Visible Then
      txtProductName.Text = ""
      txtQtyUnit.Text = ""
      txtPrice.Text = ""
      txtInStock.Text = ""
      chkDiscontinued.Checked = False
    Else
      pnlForm.Visible = True
    End If

      ' Gán giá trị CommandArgument làm cờ cho phương thức SaveItem.
    btnSave.CommandArgument = "Add"
  End Sub

  ' Xử lý nút "Save Changes"
  Private Sub btnSave_Click(ByVal sender As Object, ByVal e As System.EventArgs
Handles btnSave.Click

   If IsValid Then
       SaveItem()
    End If

  End Sub
```

```
' Xử lý sự kiện ItemCommand của DataGrid.
Private Sub grdProducts_ItemCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventArgs) Handles
grdProducts.ItemCommand
    ' Tất cả command trước tiên phải qua thủ tục ItemCommand.
  If e.Item.ItemType = ListItemType.Pager Or _
    e.Item.ItemType = ListItemType.Header Then Exit Sub

    ' Để biết Button nào đã được nhấn, bắt thuộc tính CommandSource
    ' của DataGridCommandEventArg "e"
  Dim btn As Button = CType(e.CommandSource, Button)
  If btn.Text = "Edit" Then
      ' Các giá trị của các field trong form edit được lấy từ tập hợp Cells của e.Item
    txtProductName.Text = e.Item.Cells(1).Text
    txtQtyUnit.Text = e.Item.Cells(2).Text
      ' Cắt dấu $ ra khỏi giá tiền (price)
    txtPrice.Text = Microsoft.VisualBasic.Right(e.Item.Cells(3).Text, _
      Len(e.Item.Cells(3).Text) - 1)
    txtInStock.Text = e.Item.Cells(4).Text
    chkDiscontinued.Checked = _
      CType(e.Item.Cells(5).FindControl("chkDiscontinuedGrid"), _
        CheckBox).Checked
    pnlForm.Visible = True
  Else ' Xóa product
      ' Sử dụng tập hợp DataKeys để truy xuất đến các giá trị khóa của mỗi record
      ' trong DataGrid. Tập hợp này cho phép lưu trường khóa của control DataGrid
      ' mà không cần hiển thị nó trong control. Dữ liệu của tập hợp này tự động lấy
      ' giá trị từ field được chỉ định bởi thuộc tính DataKeyField.

    DeleteItem(grdProducts.DataKeys(e.Item.ItemIndex).ToString)
  End If

    ' Xóa CommandArgument để phục vụ cho nút "Save". Nó được gán là "Add"
    ' khi nhấn nút "Add New Item" và giá trị này vẫn còn vì ViewState của button
    ' là enable.
  btnSave.CommandArgument = ""
End Sub
```

```
' Xử lý sự kiện SortCommand cho DataGrid. Sự kiện này xảy ra khi người sử dụng
' nhấn chuột vào một trong các tiêu đề của các cột, DataGrid phải có thuộc tính
' AllowSorting = True và một SortExpression được quy định trong BoundColumn
Private Sub grdProducts_SortCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridSortCommandEventArgs) Handles
grdProducts.SortCommand
    ' Lưu lại sort expression hiện hành trong một HTML control, nếu không sẽ
    ' mất kiểu sắp xếp hiện hành khi phân trang.
    htmlHiddenSortExpression.Value = e.SortExpression
    ' Kết lại dữ liệu
    BindProductsGrid()
End Sub

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ' Công việc này chỉ làm khi lần đầu tiên vào trang này, hoặc nhấn CTRL+F5
    If Not Page.IsPostBack Then
        SetupDemo()
        BindProductsGrid()
    End If
End Sub

' Xử lý sự kiện thay đổi trang trên DataGrid. Thuộc tính AllowPaging phải bằng True
Private Sub Products_PageIndexChanged(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles
grdProducts.PageIndexChanged

    grdProducts.CurrentPageIndex = e.NewPageIndex
    BindProductsGrid()
End Sub

' Thủ tục này tạo một bảng (HowToProducts) trên SQL Server phục vụ cho mục
' đích demo và  loại bỏ các DataView đã tồn tại trong application cache.
Private Sub SetupDemo()
    Dim strSQL As String = _
        "USE Northwind" & vbCrLf & _
        "IF EXISTS (" & _
        "SELECT * " & _
        "FROM Northwind.dbo.sysobjects " & _
        "WHERE Name = 'HowToProducts' " & _
```

```
        "AND TYPE = 'u')" & vbCrLf & _
        "BEGIN" & vbCrLf & _
        "DROP TABLE HowToProducts" & vbCrLf & _
        "END" & vbCrLf & _
        "SELECT ProductID, ProductName, QuantityPerUnit, UnitPrice, " & _
        "  UnitsInStock, Discontinued " & _
        "INTO Northwind.dbo.HowToProducts " & _
        "FROM Products " & _
        "ALTER TABLE HowToProducts " & _
        "ADD CONSTRAINT [ProductID] PRIMARY KEY CLUSTERED (ProductID)"

    Dim scnnNW As New SqlConnection(SQL_CONNECTION_STRING)
    Dim scmd As New SqlCommand(strSQL, scnnNW)
    scnnNW.Open()
    scmd.ExecuteNonQuery()
    scnnNW.Close()

    Cache.Remove("dvProducts")
End Sub

' Thủ tục này lấy một DataView và kết nó vào DataGrid Products
Private Sub BindProductsGrid()
    GetDataSource()

    With grdProducts
        .DataSource = dvProducts
        .DataBind()
    End With
End Sub

' Hàm này tạo mới một DataSet
Private Function CreateDataSet() As DataSet

    Dim strSQL As String = _
        "SELECT *  " & _
        "FROM HowToProducts"
    Dim scnnNW As New SqlConnection(SQL_CONNECTION_STRING)

    ' Đối tượng SqlCommand dùng để thực hiện các lệnh SQL
    Dim scmd As New SqlCommand(strSQL, scnnNW)
```

```
    ' SqlDataAdapter sử dụng đối tượng SqlCommand để đưa dữ liệu vào DataSet
    Dim sda As New SqlDataAdapter(scmd)

    ' Tạo và đưa dữ liệu vào DataSet
    Dim ds As New DataSet()
    sda.Fill(ds)

    Return ds
End Function

' Xóa một item khỏi Database
Private Sub DeleteItem(ByVal strProductID As String)
    Dim strSQL As String = _
      "DELETE " & vbCrLf & _
      "FROM HowToProducts " & _
      "WHERE ProductID = " & strProductID

    Dim scnnNW As New SqlConnection(SQL_CONNECTION_STRING)
    Dim scmd As New SqlCommand(strSQL, scnnNW)

    Try
      scnnNW.Open()
      scmd.ExecuteNonQuery()

      Cache.Remove("dvProducts")
      BindProductsGrid()

      strMsg = "Product successfully deleted from the database."
      pnlForm.Visible = False
    Catch exp As Exception
      strErrorMsg = "Database error! Product not deleted from database. " & _
        "Error message: " & exp.Message
    Finally
      scnnNW.Close()
    End Try
End Sub
```

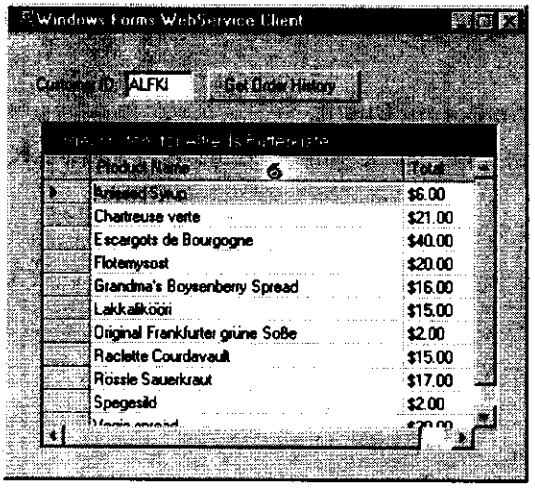
```
' Thủ tục này trả về một DataView được sắp xếp, dataview này có thể lấy từ ứng
' dụng hoặc tạo mới. Điều này tăng hiệu xuất ứng dụng vì không phải kết nối
' vào Database và tạo mới một DataSet mỗi khi chọn trang hoặc sắp xếp.
Private Sub GetDataSource()

    ' Nếu tìm thấy DataView trong cache, sử dụng nó
    If Not IsNothing(Cache("dvProducts")) Then
        ' Tất cả các thành phần trong cache đều có kiểu là Object, do đó
        ' phải ép nó kiểu trước khi sử dụng
        dvProducts = CType(Cache("dvProducts"), DataView)
    Else ' Không tìm thấy DataView trong cache
        ' Tạo mới một DataView
        dvProducts = CreateDataSet().Tables(0).DefaultView

        ' Đưa DataView mới tạo vào Cache
        Cache("dvProducts") = dvProducts
    End If

    ' sắp xếp DataView dựa trên expression lưu trong HtmlInputHidden
    dvProducts.Sort = htmlHiddenSortExpression.Value
End Sub

' Thủ tục này lưu một record vào database, bao gồm cập nhật một record đã
' tồn tại hoặc thêm một record mới. Các chức năng dựa trên giá trị của thuộc
' tính btnSave.CommandArgument
Private Sub SaveItem()
    Dim strSQL As String

    If btnSave.CommandArgument = "Add" Then
        strSQL = _
            "INSERT INTO HowToProducts " & _
            "  (ProductName, QuantityPerUnit, UnitPrice, UnitsInStock, " & _
            "   Discontinued) " & _
            "VALUES " & _
            "  (@ProductName, @QuantityPerUnit, @UnitPrice, @UnitsInStock, " & _
            "   @Discontinued)"

    Else ' Cập nhật một record đã có
```

```
    strSQL = _
      "UPDATE HowToProducts " & _
      "SET ProductName = @ProductName, " & _
      "   QuantityPerUnit = @QuantityPerUnit, " & _
      "   UnitPrice = @UnitPrice, " & _
      "   UnitsInStock = @UnitsInStock, " & _
      "   Discontinued = @Discontinued " & _
      "WHERE ProductID = @ProductID"
  End If

  Dim scnnNW As New SqlConnection(SQL_CONNECTION_STRING)
  Dim scmd As New SqlCommand(strSQL, scnnNW)

   ' Thêm tất cả các tham số SQL cần thiết
  With scmd.Parameters
     ' Tham số ProductID chỉ cần cho cập nhật
    If btnSave.CommandArgument <> "Add" Then
      .Add(New SqlParameter("@ProductID", _
        SqlDbType.Int)).Value = _
        CInt(grdProducts.DataKeys(grdProducts.SelectedIndex).ToString)
    End If

    .Add(New SqlParameter("@ProductName", _
      SqlDbType.NVarChar, 40)).Value = txtProductName.Text
    .Add(New SqlParameter("@QuantityPerUnit", _
      SqlDbType.NVarChar, 20)).Value = txtQtyUnit.Text
    .Add(New SqlParameter("@UnitPrice", _
      SqlDbType.Money)).Value = CDbl(txtPrice.Text)
    .Add(New SqlParameter("@UnitsInStock", _
      SqlDbType.Int)).Value = CInt(txtInStock.Text)
    .Add(New SqlParameter("@Discontinued", _
      SqlDbType.Bit)).Value = chkDiscontinued.Checked
  End With

  Try
    scnnNW.Open()
    scmd.ExecuteNonQuery()
    Cache.Remove("dvProducts")
    BindProductsGrid()
```

```
            strMsg = "Product successfully saved to the database."
            pnlForm.Visible = False
        Catch exp As Exception
            strErrorMsg = "Database error! Product not saved to the " & _
              "database. Error message: " & exp.Message
        Finally
            scnnNW.Close()
        End Try
    End Sub
End Class
```

31. Xây dựng các dịch vụ Web (Web Service)

Các Web service là một công nghệ rất quan trọng để triển khai các ứng dụng từ xa sử dụng các chuẩn HTTP và XML.



Các chức năng chính:

Ví dụ này gồm một Web Service và hai ứng dụng client sử dụng Web service này. Một client ở dạng ứng dụng Web, và client kia là ứng dụng Windows. Web service cung cấp hai phương thức trả về dữ liệu lấy từ database Northwind. **GetTenMostExpensiveProducts()** là một phương thức đơn giản trả về một DataSet, nó được sử dụng bởi ứng dụng Web. **GetCustomerOrderHistory()** biểu diễn cách trả về một kiểu phức tạp – một instance của một class có hai trường public, một trong số chúng là kiểu DataSet. Phương thức này được sử dụng bởi ứng dụng Windows.

Chú ý:

Chương trình có sử dụng database Northwind trong SQL Server.

Cài đặt Web Service và Web Forms Client:

- Nhấp đúp chuột vào SetupWebs.vbs (file này giả định thư mục web mặc nhiên là c:\inetpub\wwwroot).
- Nếu có lỗi khi cài đặt, chép hai thư mục trong thư mục WebSites vào thư mục Web gốc mặc nhiên, cấu hình hai thư mục này là các IIS Application bằng cách: mở trình quản trị IIS, nhấn chuột phải vào tên thư mục, chọn Properties và nhấn vào nút Create trên tab Directory.

Chạy Web Forms Client:

- Trong Soluton Explorer, nhấn chuột phải vào WebServiceClient và chọn Set As StartUp Project.
- Nhấn chuột phải vào Main.aspx và chọn Set As Start Page.
- Nhấn **F5** để build và chạy ứng dụng.

Chạy Windows Forms Client:

- Trong Solution Explorer, nhấn chuột phải vào VB.NET Web Services (Client) và chọn Set as StartUp Project.
- Nhấn **F5** để chạy ứng dụng.
- Các giá trị Customer ID là COMMI, GODOS, và ISLAT.

Chạy trực tiếp Web Service, không cần client:

- Trong Solution Explorer, nhấn chuột phải vào main.asmx và chọn **View in Browser**.
- Nhấn vào liên kết của các phương thức.

Thêm một DataSet vào project cho dữ liệu lấy từ database Northwind:

- Trong Solution Explorer, nhấn chuột phải vào project, chọn Add I Add New Item.
- Trong phần **Templates**, chọn DataSet. Đặt cho nó một tên với phần mở rộng là .xsd và nhấn Open. Thao tác này thêm một XSD schema vào project. Từ file này phát sinh class có kiểu DataSet.
- Mở Server Explorer và mở các node bên dưới SQL Servers để đi đến Northwind Stored Procedures.
- Chọn và kéo một stored procedure vào dao diện đồ họa của file xsd.

- Nhấn chuột phải vào dao diện và đánh dấu Generate DataSet nếu chưa được đánh dấu.
- Nhấn **CTRL + S** để lưu và phát sinh class có kiểu DataSet.
- Để xem class này, trong Solution Explorer nhấn vào nút **Show All Files**. Mở node .xsd để thấy file .vb tương ứng.

Project: **WebServices**

CustOrderHist.vb

```
Option Strict On
Imports System.Xml.Serialization

Public Class CustomerAndOrderHistoryInfo
    Public Orders As dsCustOrderHist
    ' Khai báo sử dụng XmlAttribute để XML trả về bởi Web Service  với field
    ' CompanyName là một thuộc tính thay vì một element (kiểu mặc nhiên). Thuộc
    ' tính AttributeName quy định thay đổi tên của nó từ CompanyName sang Company.
    <XmlAttributeAttribute(AttributeName:="Company")> Public CompanyName As
  String
End Class
```

dsCustOrderHist.xsd

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="dsCustOrderHist"
  targetNamespace="http://tempuri.org/dsCustOrderHist.xsd"
  elementFormDefault="qualified" attributeFormDefault="qualified"
  xmlns="http://tempuri.org/dsCustOrderHist.xsd"
  xmlns:mstns="http://tempuri.org/dsCustOrderHist.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
  microsoft-com:xml-msdata">
  <xs:element name="dsCustOrderHist" msdata:IsDataSet="true">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="CustOrderHist">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ProductName" type="xs:string" />
              <xs:element name="Total" msdata:ReadOnly="true" type="xs:int"
  minOccurs="0" />
```

```
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

main.asmx

```
Option Strict On
Imports System.Web.Services
Imports System.Data.SqlClient

<WebService()> _
Public Class Main
  Inherits System.Web.Services.WebService

  ' Code phát sinh được bỏ qua

  ' Khởi tạo các hằng số để kết nối vào database và hiển thị lỗi kết nối
  Protected Const CONNECTION_ERROR_MSG As String = _
    "To run this sample, you must have SQL " & _
    "or MSDE with the Northwind database installed."

  Protected Const SQL_CONNECTION_STRING As String = _
    "Server=localhost;" & _
    "DataBase=northwind;" & _
    "Integrated Security=false;" & _
    "User ID=sa;"

  Protected strConn As String = SQL_CONNECTION_STRING

  ' Trả về một instance của class CustomerAndOrderHistoryInfo
  <WebMethod> _
  Public Function GetCustomerOrderHistory(ByVal strCustID As String) _
    As CustomerAndOrderHistoryInfo

    ' Kết nối vào SQL Server cục bộ
    Dim IsConnecting As Boolean = True
```

```
While IsConnecting
  Dim scnnNW As New SqlConnection(strConn)
  Dim scmd As New SqlCommand("CustOrderHist", scnnNW)

   ' Tạo một instance của kiểu trả về
  Dim cohi As New CustomerAndOrderHistoryInfo()

  Try
    ' Báo cho đối tượng Command biết rằng văn bản truyền vào khi tạo đối
    ' tượng là một stored procedure không văn là câu truy vấn SQL thông thường.
    scmd.CommandType = CommandType.StoredProcedure

     ' Tạo một đối tượng SqlParameter để truyền vào stored procedure
    scmd.Parameters.Add(New SqlParameter("@CustomerID", _
      SqlDbType.NChar, 5)).Value = strCustID

     ' SqlDataAdapter sử dụng SqlCommand để đưa dữ liệu vào DataSet
    Dim sda As New SqlDataAdapter(scmd)

     ' Tạo một instance của class có kiểu DataSet
    Dim dsOrderHistory As New dsCustOrderHist()

    sda.Fill(dsOrderHistory.CustOrderHist)

     ' Dữ liệu đã được lấy thành công, thoát vòng lặp
    IsConnecting = False

     ' Gán instance có kiểu DataSet (đã được nhập dữ liệu) vào trường Orders
    cohi.Orders = dsOrderHistory

     ' Thay đổi lệnh SQL để lấy tên công ty (company name)
    scmd.CommandText = _
      "SELECT CompanyName " & _
      "FROM Customers " & _
      "WHERE CustomerID = @CustomerID"

     ' Thay đổi kiểu lệnh sang dạng lệnh SQL thông thường
    scmd.CommandType = CommandType.Text
```

```
            ' Mở kết nối và thực hiện câu lệnh SQL mới. Sử dụng ExecuteScalar() để
            ' thực hiện bởi vì ta biết trước chỉ trả về cột đầu tiên của record đầu tiên.
            scnnNW.Open()
            Dim objReturnVal As Object = scmd.ExecuteScalar()
            If Not IsDBNull(objReturnVal) Then
              cohi.CompanyName = objReturnVal.ToString
            End If

          Catch expSql As SqlException
            Throw New Exception(expSql.Message)

          Catch exp As Exception
            Throw New Exception(exp.Message)
          Finally
            ' Đóng kết nối
            scnnNW.Close()
          End Try

          Return cohi
        End While
    End Function

    ' Trả về một DataSet chứa 10 sản phẩm đắt tiền nhất
    <WebMethod()> _
    Public Function GetTenMostExpensiveProducts() As DataSet

        Dim IsConnecting As Boolean = True
        While IsConnecting
          Try
            Dim scnnNW As New SqlConnection(strConn)

            ' Đặt tên Stored Procedure trong dấu ngoặc vuông vì nó chứa khoảng trắng
            Dim scmd As New SqlCommand("[Ten Most Expensive Products]", _
              scnnNW)

            scmd.CommandType = CommandType.StoredProcedure

            Dim sda As New SqlDataAdapter(scmd)
```

```
            ' Tạo và đưa dữ liệu vào DataSet
            Dim dsTenMostExpProds As New DataSet()
            sda.Fill(dsTenMostExpProds)

            IsConnecting = False

            Return dsTenMostExpProds

          Catch expSql As SqlException
            Throw New Exception(expSql.Message)
            Exit Function

          Catch exp As Exception
              ' Không thể kết nối vào SQL Server
            Throw New Exception(exp.Message)
          End Try
        End While
    End Function
End Class
```

Project: **WebServiceClient**

Web References: http://localhost/WebService/main.asmx

Main.aspx

```
<%@Page Language="vb" AutoEventWireup="false" Codebehind="main.aspx.vb"
  Inherits="WebServiceClient.Main" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>Visual Basic .NET WebService Client</title>
    <meta name="GENERATOR" content="Microsoft Visual Studio.NET 7.0">
    <meta name="CODE_LANGUAGE" content="Visual Basic 7.0">
    <meta name="vs_defaultClientScript" content="JavaScript">
    <meta name="vs_targetSchema"
  content="http://schemas.microsoft.com/intellisense/ie5">
  </HEAD>
  <body>
    <form id="frmMain" method="post" runat="server">
      <P>
```

```
        <hr>
      <P></P>
      <P>
        <asp:Button id="btnTenMost" runat="server" Text="Get Ten Most
Expensive Products"></asp:Button><BR>
        <BR>
        <asp:DataGrid id="grdProducts" runat="server" BorderColor="#E7E7FF"
BorderStyle="None" BorderWidth="1px" BackColor="White" CellPadding="3"
GridLines="Horizontal">
          <SelectedItemStyle Font-Bold="True" ForeColor="#F7F7F7"
BackColor="#738A9C"></SelectedItemStyle>
          <AlternatingItemStyle BackColor="#F7F7F7"></AlternatingItemStyle>
          <ItemStyle ForeColor="#4A3C8C" BackColor="#E7E7FF"></ItemStyle>
          <HeaderStyle Font-Bold="True" ForeColor="#F7F7F7"
BackColor="#4A3C8C"></HeaderStyle>
          <FooterStyle ForeColor="#4A3C8C"
BackColor="#B5C7DE"></FooterStyle>
          <PagerStyle HorizontalAlign="Right" ForeColor="#4A3C8C"
BackColor="#E7E7FF" Mode="NumericPages"></PagerStyle>
        </asp:DataGrid></P>
    </form>
  </body>
</HTML>
```

main.aspx.vb

```
Public Class Main
  Inherits System.Web.UI.Page
  Protected WithEvents grdProducts As System.Web.UI.WebControls.DataGrid
  Protected WithEvents btnTenMost As System.Web.UI.WebControls.Button

#Region " Web Form Designer Generated Code "

  'This call is required by the Web Form Designer.
  <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

  End Sub

  Private Sub Page_Init(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Init
    'CODEGEN: This method call is required by the Web Form Designer
    'Do not modify it using the code editor.
```

```
        InitializeComponent()
    End Sub

#End Region

    ' Thủ tục này xử lý sự kiện nhấn chuột trên nút "Get Ten Most Expensive Products".
    ' Tạo một instance của class Web Service. Kết quả của phương thức Web được
    ' gọi sau đó được kết vào DataGrid.
    Private Sub btnTenMost_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnTenMost.Click
        Dim ws As New localhost.Main()
        grdProducts.DataSource = ws.GetTenMostExpensiveProducts
        grdProducts.DataBind()
    End Sub
End Class
```

Project: **VB.NET WebService (Client)**

Web References: http://localhost/WebServices/main.asmx

frmMain.vb

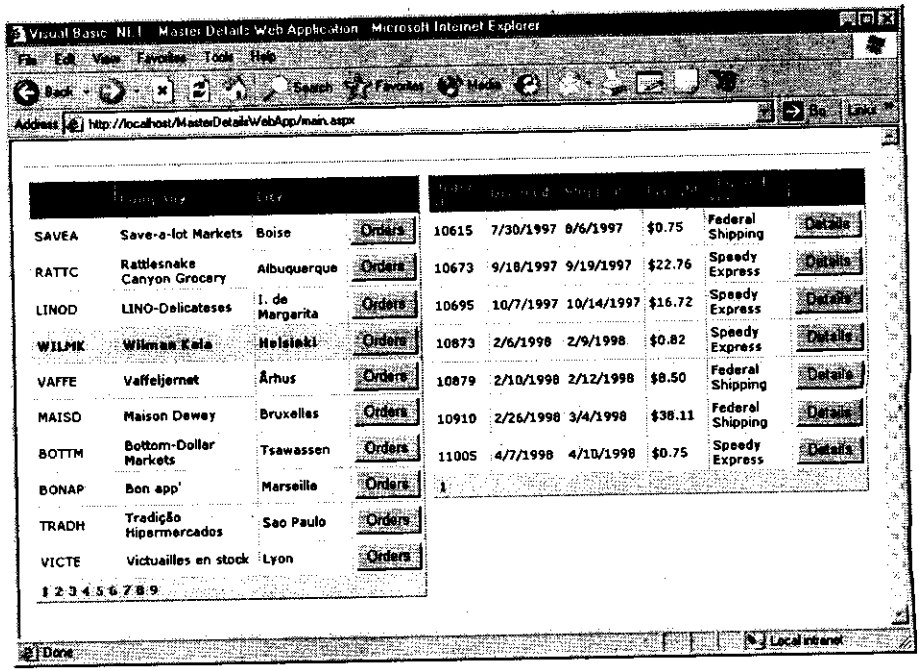
```
Option Strict On
Imports System.Text.RegularExpressions

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    ' Thủ tục này xử lý nút "Get Orders History".
    Private Sub btnGetOrderHistory_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnGetOrderHistory.Click
        If Not CustIDIsValid() Then
            Exit Sub
        End If

        Dim strCustID As String = Trim(txtCustID.Text)
        ' Tạo một instance của class Web service proxy
        Dim ws As New localhost.Main()
```

```
    ' Tạo một biến lưu kiểu trả về
    Dim cohi As localhost.CustomerAndOrderHistoryInfo
    Try
      cohi = ws.GetCustomerOrderHistory(txtCustID.Text)
    Catch exp As Exception
      MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
    End Try

    If cohi.Orders.CustOrderHist.Rows.Count > 0 Then

      With grdOrders
        .CaptionText = "Order History for " & cohi.Company & ":"
        .CaptionFont = New Font("Microsoft Sans Serif", 10)
        .DataSource = cohi.Orders.CustOrderHist
        .TableStyles.Clear()
      End With

      Dim grdTableStyle1 As New DataGridTableStyle()
      With grdTableStyle1
          ' Luôn luôn quy định MappingName, ngay cả DataView chỉ có một bảng.
        .MappingName = cohi.Orders.Tables(0).TableName
      End With

      ' Sử dụng column style để thay cho các cột tự động phát sinh trong DataTable.
      ' Khi có sử dụng các đối tượng column style, các cột muốn hiển thị phải kết
      ' với một đối tượng column style.
      Dim grdColStyle1 As New DataGridTextBoxColumn()
      With grdColStyle1
        .MappingName = "ProductName"
        .HeaderText = "Product Name"
        .Width = 230
      End With

      Dim grdColStyle2 As New DataGridTextBoxColumn()
      With grdColStyle2
        .MappingName = "Total"
        .HeaderText = "Total"
        .Format = "c"
        .Width = 75
      End With
```

```
        grdTableStyle1.GridColumnStyles.AddRange _
          (New DataGridColumnStyle() {grdColStyle1, grdColStyle2})
        grdOrders.TableStyles.Add(grdTableStyle1)

        grdOrders.Visible = True
        lblNoResults.Visible = False
      Else
        lblNoResults.Text = "No data was returned for the Customer ID you " & _
          "entered. Please check the value and try again. Examples are " & _
          "COMMI, GODOS, and ISLAT."
        lblNoResults.Visible = True
        grdOrders.Visible = False
      End If
    End Sub

    ' Kiểm tra tính hợp lệ của CustomerID
    Private Function CustIDIsValid() As Boolean
      ' Sử dụng regular expression để kiểm tra thỏa theo mẫu
      If Not Regex.IsMatch(txtCustID.Text, "^\s*(\D){5}\s*$") Then
        MsgBox("You must enter a five-letter CustomerID. " & _
          "Examples are COMMI, GODOS, and ISLAT.", _
          MsgBoxStyle.Exclamation, Me.Text)
        Return False
      End If
      Return True
    End Function
End Class
```

32. Tạo ứng dụng Web Master-Details

Giao diện master-details là một trong các kiểu giao diện truy xuất dữ liệu phổ biến nhất. Tưởng tượng rằng dữ liệu được tổ chức theo cây phân cấp, giống như control TreeView sử dụng trong Windows Explorer. Với giao diện Master-Details ta có thể khoan sâu vào dữ liệu theo nhiều cấp, cấp trước đó là cấp cha (master) và các cấp con của master là các chi tiết của master (details).



Các chức năng chính:

- Chương trình có ba control DataGrid, mỗi control được kết vào một DataSet riêng của nó sử dụng database Northwind. Master DataGrid chứa cấp cao nhất hiển thị Customers (các khách hàng). Nhấn vào nút "Orders" của một customer để xem cấp kế tiếp của dữ liệu trong DataGrid thứ hai: Tất cả các đơn đặt hàng (order) của khách hàng (customer) đó. Nhấn vào nút "Details" của Order để xem cấp dữ liệu cuối cùng: tất cả các chi tiết của order đó.

Chú ý:

- Chương trình sử dụng database Northwind trong SQL Server.
- Xem cách cài đặt ứng dụng Web ở các ví dụ trước.

Main.aspx

```
<%@Page Language="vb" AutoEventWireup="false" Codebehind="main.aspx.vb"
  Inherits="HowToBuildMasterDetailsWebApp.Main"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>Visual Basic .NET - Master-Details Web Application</title>
    <meta content="Microsoft Visual Studio.NET 7.0" name="GENERATOR">
    <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
```

```
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
    <LINK href="styles.css" rel="stylesheet">
</HEAD>
<body>
    <form id="frmMain" method="post" runat="server">
      <hr>
      <TABLE id="Table1" width="100%" border="0">
        <TR>
          <TD valign="top"><asp:datagrid id="grdCustomers" runat="server"
AllowSorting="True" AutoGenerateColumns="False" AllowPaging="True">
              <SelectedItemStyle CssClass="DataGrid_SelectedItemStyle" />
              <ItemStyle CssClass="DataGrid_ItemStyle" />
              <HeaderStyle CssClass="DataGrid_HeaderStyle" />
              <FooterStyle CssClass="DataGrid_FooterStyle" />
              <Columns>
                <asp:BoundColumn DataField="CustomerID"
SortExpression="CustomerID" HeaderText="Customer ID" />
                <asp:BoundColumn DataField="CompanyName"
HeaderText="Company" />
                <asp:BoundColumn DataField="City" HeaderText="City" />
                <asp:ButtonColumn Text="Orders" ButtonType="PushButton"
CommandName="Select" />
              </Columns>
              <PagerStyle CssClass="DataGrid_PagerStyle"
Mode="NumericPages" />
            </asp:datagrid>
          </TD>
          <TD valign="top">
            <asp:datagrid id="grdOrders" runat="server" AllowPaging="True"
AutoGenerateColumns="False" Visible="False">
              <SelectedItemStyle CssClass="DataGrid_SelectedItemStyle" />
              <ItemStyle CssClass="DataGrid_ItemStyle" />
              <HeaderStyle CssClass="DataGrid_HeaderStyle" />
              <FooterStyle CssClass="DataGrid_FooterStyle" />
              <Columns>
                <asp:BoundColumn DataField="OrderID" HeaderText="Order ID"
/>
                <asp:BoundColumn DataField="OrderDate"
HeaderText="Ordered" DataFormatString="{0:d}" />
```

```
                    <asp:BoundColumn DataField="ShippedDate"
HeaderText="Shipped" DataFormatString="{0:d}" />
                    <asp:BoundColumn DataField="Freight" HeaderText="Freight"
DataFormatString="{0:c}" />
                    <asp:BoundColumn DataField="ShippedVia"
HeaderText="Shipped Via" />
                    <asp:ButtonColumn Text="Details" ButtonType="PushButton"
CommandName="Select" />
                  </Columns>
                  <PagerStyle CssClass="DataGrid_PagerStyle"
Mode="NumericPages" />
                </asp:datagrid>
              </TD>
            </TR>
            <TR>
              <TD class="details" valign="top" colSpan="2"><asp:datagrid
id="grdOrderDetails" runat="server" AutoGenerateColumns="False"
AllowPaging="False" Visible="False">
                  <SelectedItemStyle CssClass="DataGrid_SelectedItemStyle" />
                  <ItemStyle CssClass="DataGrid_ItemStyle" />
                  <HeaderStyle CssClass="DataGrid_HeaderStyle" />
                  <FooterStyle CssClass="DataGrid_FooterStyle" />
                  <Columns>
                    <asp:BoundColumn DataField="ProductName"
HeaderText="Product" />
                    <asp:BoundColumn DataField="UnitPrice" HeaderText="Price"
DataFormatString="{0:c}" />
                    <asp:BoundColumn DataField="Quantity" HeaderText="Quantity"
/>
                    <asp:BoundColumn DataField="Discount" HeaderText="Discount"
DataFormatString="{0:p2}" />
                    <asp:BoundColumn DataField="CategoryName"
HeaderText="Category" />
                  </Columns>
                </asp:datagrid>
              </TD>
            </TR>
          </TABLE>
        </form>
      </body>
    </HTML>
```

main.aspx.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class Main
    Inherits System.Web.UI.Page
    Protected WithEvents lblTitle As System.Web.UI.WebControls.Label
    Protected WithEvents grdCustomers, grdOrders, grdOrderDetails As
    System.Web.UI.WebControls.DataGrid
    Protected WithEvents btnAbout As System.Web.UI.WebControls.Button

    ' Code phát sinh Web form được bỏ qua

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL " & _
        "or MSDE with the Northwind database installed. "

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=northwind;" & _
        "Integrated Security=false;" & _
        "User ID=sa;Password="

    Protected strConn As String = SQL_CONNECTION_STRING

    ' Thủ tục xử lý sự kiện ItemCommand trên master DataGrid chính (grdCustomer).
    ' Sự kiện này xảy ra khi bất kỳ nút nào trong DataGrid được nhấn, kể cả LinkButton.
    ' Đây là sự kiện chính để quản lý ứng dụng Master-Details Web Application
    Private Sub grdCustomers_ItemCommand(ByVal source As Object, ByVal e As
    System.Web.UI.WebControls.DataGridCommandEventArgs) Handles
    grdCustomers.ItemCommand
        ' e.Item là một đối tượng DataGridItem tương ứng cho item được chọn. Đối tượng
        ' này không chỉ là các record mà có thể là các thanh tiêu đề (tên của các cột khi
        ' nhấn vào đó để sắp xếp) hoặc là các LinkButton sử dụng để phân trang.
        If e.Item.ItemType = ListItemType.Pager Then Exit Sub

        ' Tạo một đối tượng SqlParameter để truyền vào một thủ tục con để kết dữ liệu
        ' vào DataGrid cấp kế tiếp.
        Dim param As New SqlParameter("@CustomerID", SqlDbType.NChar, 5)
        param.Value = grdCustomers.DataKeys(e.Item.ItemIndex)
```

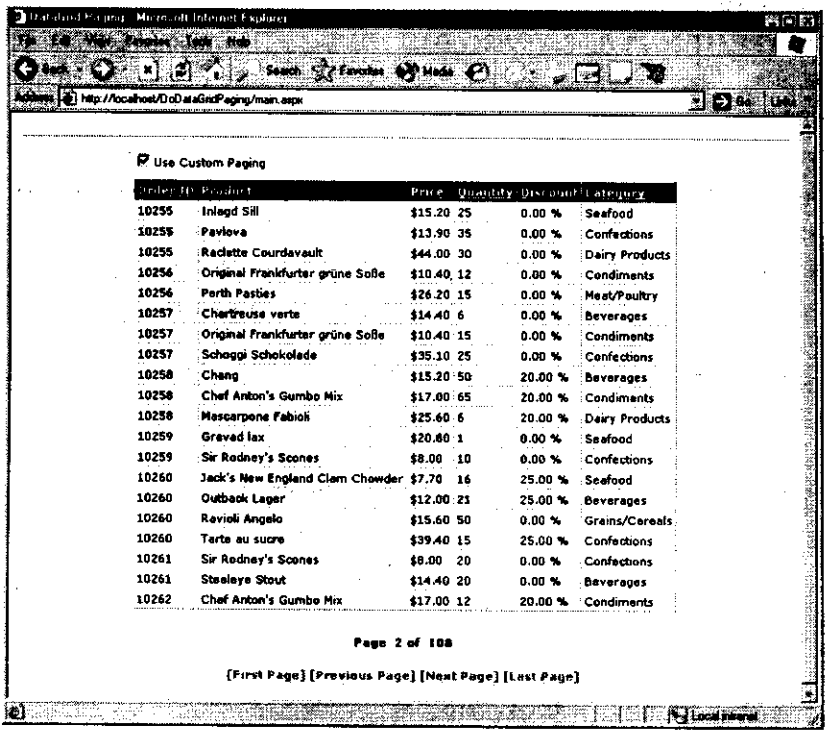
```
    BindOrdersGrid(param)
  End Sub

  Private Sub grdCustomers_PageIndexChanged(ByVal source As Object, ByVal e
  As System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles
  grdCustomers.PageIndexChanged
    grdCustomers.CurrentPageIndex = e.NewPageIndex
    BindCustomersGrid()
  End Sub

  ' Xử lý sự kiện ItemCommand cho master DataGrid cấp thứ hai (grdOrders)
  Private Sub grdOrders_ItemCommand(ByVal source As Object, ByVal e As
  System.Web.UI.WebControls.DataGridCommandEventArgs) Handles
  grdOrders.ItemCommand
    If e.Item.ItemType = ListItemType.Pager Then Exit Sub

    Dim param As New SqlParameter("@OrderID", SqlDbType.Int)
    param.Value = grdOrders.DataKeys(e.Item.ItemIndex)
    BindOrderDetailsGrid(param)
  End Sub

  Private Sub grdOrders_PageIndexChanged(ByVal source As Object, ByVal e As
  System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles
  grdOrders.PageIndexChanged
    grdOrders.CurrentPageIndex = e.NewPageIndex

    Dim param As New SqlParameter("@CustomerID", SqlDbType.NChar, 5)
    param.Value = grdCustomers.DataKeys(grdCustomers.SelectedIndex)
    BindOrdersGrid(param)
  End Sub

  Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles MyBase.Load
    If Not Page.IsPostBack Then
      ' Khi mới vào chỉ hiển thị master DataGrid chính
      BindCustomersGrid()
    End If
  End Sub


' Thủ tục này gọi CreateDataSet và kết DataGrid Customers vào giá trị trả về
```

```
Sub BindCustomersGrid()

    Dim strSQL As String = _
      "SELECT c.CustomerID, c.CompanyName, c.City, " & _
      "  COUNT(o.OrderDate) AS OrderCount " & _
      "FROM Customers c " & _
      "INNER JOIN Orders o ON c.CustomerID = o.CustomerID " & _
      "GROUP BY c.CustomerID, c.CompanyName, c.City"

    With grdCustomers
      .DataKeyField = "CustomerID"
      .DataSource = CreateDataSet(strSQL)
      .DataBind()
    End With
End Sub

' Thủ tục này gọi CreateDataSet và kết DataGric OrderDetails vào giá trị trả về
Sub BindOrderDetailsGrid(ByVal param As SqlParameter)

    Dim strSQL As String = _
      "SELECT p.ProductName, od.UnitPrice, od.Quantity, " & _
      "  od.Discount, c.CategoryName " & _
      "FROM [Order Details] od " & _
      "INNER JOIN Products p ON od.ProductID = p.ProductID " & _
      "INNER JOIN Categories c ON p.CategoryID = c.CategoryID " & _
      "WHERE OrderID = @OrderID"

    With grdOrderDetails
      .DataSource = CreateDataSet(strSQL, param)
      .DataBind()
      .Visible = True
    End With

End Sub
```

```
    ' Thủ tục này gọi CreateDataSet và kết DataGrid Orders vào giá trị trả về
    Sub BindOrdersGrid(ByVal param As SqlParameter)

      Dim strSQL As String = _
        "SELECT o.OrderID, o.OrderDate, o.ShippedDate, " & _
        "  o.Freight, s.CompanyName As ShippedVia " & _
        "FROM Orders o " & _
        "INNER JOIN Shippers s ON o.ShipVia = s.ShipperID " & _
        "WHERE CustomerID = @CustomerID"

      With grdOrders
        ' Gán DataKeyField để tập hợp DataKeys được nhập dữ liệu và có thể sử
        ' dụng trong envent handler ItemCommand
        .DataKeyField = "OrderID"
        .DataSource = CreateDataSet(strSQL, param)
        .DataBind()
        .Visible = True
      End With
    End Sub

    ' Thủ tục này được sử dụng bởi cả ba DataGrid
    Private Function CreateDataSet(ByVal strSQL As String, _
      Optional ByVal sqlParam As SqlParameter = Nothing) As DataSet

      Dim scnnNW As New SqlConnection(strConn)

      Dim scmd As New SqlCommand(strSQL, scnnNW)

      If Not IsNothing(sqlParam) Then
        scmd.Parameters.Add(sqlParam)
      End If

      Dim sda As New SqlDataAdapter(scmd)
      Dim ds As New DataSet()
      sda.Fill(ds)
      Return ds
    End Function
End Class
```

33. Phân trang DataGrid trên Web Forms

Phân trang là hiển thị một tập kết quả con trên một trang tại một thời điểm, và sẽ làm cho việc hiển thị kết quả nhanh hơn.



Order ID	Product	Price	Quantity	Discount	Category
10255	Inlagd Sill	$15.20	25	0.00 %	Seafood
10255	Pavlova	$13.90	35	0.00 %	Confections
10255	Raclette Courdavault	$44.00	30	0.00 %	Dairy Products
10256	Original Frankfurter grüne Soße	$10.40	12	0.00 %	Condiments
10256	Perth Pasties	$26.20	15	0.00 %	Meat/Poultry
10257	Chartreuse verte	$14.40	6	0.00 %	Beverages
10257	Original Frankfurter grüne Soße	$10.40	15	0.00 %	Condiments
10257	Schoggi Schokolade	$35.10	25	0.00 %	Confections
10258	Chang	$15.20	50	20.00 %	Beverages
10258	Chef Anton's Gumbo Mix	$17.00	65	20.00 %	Condiments
10258	Mascarpone Fabioli	$25.60	6	20.00 %	Dairy Products
10259	Gravad lax	$20.80	1	0.00 %	Seafood
10259	Sir Rodney's Scones	$8.00	10	0.00 %	Confections
10260	Jack's New England Clam Chowder	$7.70	16	25.00 %	Seafood
10260	Outback Lager	$12.00	21	25.00 %	Beverages
10260	Ravioli Angelo	$15.60	50	0.00 %	Grains/Cereals
10260	Tarte au sucre	$39.40	15	25.00 %	Confections
10261	Sir Rodney's Scones	$8.00	20	0.00 %	Confections
10261	Steeleye Stout	$14.40	20	0.00 %	Beverages
10262	Chef Anton's Gumbo Mix	$17.00	12	20.00 %	Condiments

Các chức năng chính:

- DataGrid phân trang được biểu diễn theo hai cách: Cách phân trang được xây dựng sẵn đi kèm với control DataGrid; Cách phân trang tùy biến sử dụng một cột "RowID" thêm vào DataSet cùng với một DataView RowFilter.
- DataGrid vẫn còn được sắp xếp trong quá trình phân trang. Sort expression được lưu trong một control HTMLInputHidden.
- Application Cache được sử dụng để lưu DataView làm DataSource cho cả hai cách phân trang. Với cách này, dữ liệu hơn 20000 records chỉ được download một lần.
- DataFormatString được sử dụng để định dạng price và discount.

Chú ý:

- Chương trình sử dụng database Northwind trong SQL Server.
- Xem cách cài đặt một ứng dụng Web trong các chương trình trước.

Thư mục: **Phan trang DataGrid tren Web Form**

main.aspx

```
<%@Page Language="vb" AutoEventWireup="false" Codebehind="main.aspx.vb"
Inherits="DoDataGridPaging.Main"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>DataGrid Paging</title>
    <meta content="Microsoft Visual Studio.NET 7.0" name="GENERATOR">
    <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
    <LINK href="styles.css" rel="stylesheet">
  </HEAD>
  <body>
    <form id="frmMain" method="post" runat="server">
      <hr>
      <table align="center">
        <tr>
          <td>
            <asp:checkbox id="chkUseCustomPaging" runat="server" Text="Use
Custom Paging" AutoPostBack="True" />
          </td>
        </tr>
        <tr>
          <td>
            <asp:datagrid id="grdOrderDetails" runat="server" AllowSorting="True"
AutoGenerateColumns="False" AllowPaging="True" PageSize="20">
              <SelectedItemStyle CssClass="DataGrid_SelectedItemStyle" />
              <ItemStyle CssClass="DataGrid_ItemStyle" />
              <HeaderStyle CssClass="DataGrid_HeaderStyle" />
              <FooterStyle CssClass="DataGrid_FooterStyle" />
              <Columns>
                <asp:BoundColumn DataField="OrderID"
SortExpression="OrderID" HeaderText="Order ID" />
                <asp:BoundColumn DataField="ProductName"
SortExpression="ProductName" HeaderText="Product" />
                <asp:BoundColumn DataField="UnitPrice"
SortExpression="UnitPrice" HeaderText="Price" DataFormatString="{0:c}" />
```

```
                    <asp:BoundColumn DataField="Quantity"
SortExpression="Quantity" HeaderText="Quantity" />
                    <asp:BoundColumn DataField="Discount"
SortExpression="Discount" HeaderText="Discount" DataFormatString="{0:p2}" />
                    <asp:BoundColumn DataField="CategoryName"
SortExpression="CategoryName" HeaderText="Category" />
                  </Columns>
                  <PagerStyle CssClass="DataGrid_PagerStyle"
Mode="NumericPages" />
                </asp:datagrid>
              </td>
            </tr>
            <tr>
              <td>
                <input id="htmlHiddenSortExpression" type="hidden" value="OrderID"
runat="server" NAME="htmlHiddenSortExpression"><br>
                <asp:Panel id="pnlCustomPaging" Runat="server"
CssClass="CustomPaging">
                  <P>Page 
                    <asp:Label id="lblCurrentPage"
runat="server"></asp:Label> of 
                    <asp:Label id="lblTotalPages" runat="server"></asp:Label></P>
                  <P>
                    <asp:LinkButton id="lbtnFirstPage" runat="server" Text="[First
Page]" OnCommand="NavigationLink_Click"
CommandName="First"></asp:LinkButton>
                    <asp:LinkButton id="lbtnPreviousPage" runat="server"
Text="[Previous Page]" OnCommand="NavigationLink_Click"
CommandName="Prev"></asp:LinkButton>
                    <asp:LinkButton id="lbtnNextPage" runat="server" Text="[Next
Page]" OnCommand="NavigationLink_Click"
CommandName="Next"></asp:LinkButton>
                    <asp:LinkButton id="lbtnLastPage" runat="server" Text="[Last
Page]" OnCommand="NavigationLink_Click"
CommandName="Last"></asp:LinkButton></P>
                </asp:Panel>
              </td>
            </tr>
          </table>
        </form>
      </body>
    </HTML>
```

main.aspx.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class Main
    Inherits System.Web.UI.Page
    Protected WithEvents lblTitle, lblCurrentPage, lblTotalPages As Label
    Protected WithEvents lbtnFirstPage, lbtnPreviousPage, lbtnNextPage,
    lbtnLastPage As LinkButton
    Protected WithEvents grdOrderDetails As DataGrid
    Protected WithEvents btnAbout As Button
    Protected WithEvents chkUseCustomPaging As
    System.Web.UI.WebControls.CheckBox
    Protected WithEvents htmlHiddenSortExpression As HtmlInputHidden
    Protected WithEvents pnlCustomPaging As Panel

' Code phát sinh được bỏ qua

    Private dvOrderDetails As DataView
    ' Sử dụng cho kiểu phân trang tùy biến (custom paging)
    Private intCurrentPage As Int32 = 1

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=northwind;" & _
        "Integrated Security=false;" & _
        "User ID=sa;Password="

    ' Xử lý sự kiện CheckedChanged cho control CheckBox.
    Private Sub chkUseCustomPaging_CheckedChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    chkUseCustomPaging.CheckedChanged
        ' Reset thuộc tính CurrentPageIndex và loại DataView ra khỏi Cache
        grdOrderDetails.CurrentPageIndex = 0
        Cache.Remove("dvOrderDetails")
        BindOrderDetailsGrid()
    End Sub
```

```
' Xử lý sự kiện PageIndexChanged cho DataGrid OrderDetails.
 ' Thủ tục này phục vụ cho chức năng phân trang có sẵn khi AllowPaging = True
Private Sub OrderDetails_PageIndexChanged(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles
grdOrderDetails.PageIndexChanged
    grdOrderDetails.CurrentPageIndex = e.NewPageIndex
     ' Lấy sort expression hiện hành (sắp xếp theo cột nào) từ HTML input control
    BindOrderDetailsGrid()
End Sub

' Xử lý khi người sử dụng nhấn vào tiêu đề các cột để sắp xếp
Private Sub grdOrderDetails_SortCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridSortCommandEventArgs) Handles
grdOrderDetails.SortCommand
     ' Lưu sort expression hiện hành vào HTML input control
    htmlHiddenSortExpression.Value = e.SortExpression
    If chkUseCustomPaging.Checked Then
        ' Quy định trang hiện hành
      intCurrentPage = CInt(lblCurrentPage.Text)
    End If
     ' Kết dữ liệu vào DataGrid
    BindOrderDetailsGrid()
End Sub

' Thủ tục này xử lý sự kiện click trên bốn control LinkButton thuộc custom paging
Protected Sub NavigationLink_Click(ByVal sender As [Object], ByVal e As
CommandEventArgs)
    Select Case e.CommandName
      Case "First"
        intCurrentPage = 1
      Case "Last"
        intCurrentPage = CInt(lblTotalPages.Text)
      Case "Next"
        intCurrentPage = CInt(lblCurrentPage.Text) + 1
      Case "Prev"
        intCurrentPage = CInt(lblCurrentPage.Text) - 1
    End Select

    BindOrderDetailsGrid()
End Sub
```

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    If Not Page.IsPostBack Then
        Cache.Remove("dvOrderDetails")
        BindOrderDetailsGrid()
    End If
End Sub

' Thủ tục này kết DataGrid OrderDetails vào một DataView có filter hoặc
' không filter như được xác định trong phương thức SetPagingControls
Private Sub BindOrderDetailsGrid()
    GetDataSource()
    SetPagingControls()

    With grdOrderDetails
        .DataSource = dvOrderDetails
        .DataBind()
    End With
End Sub

' Trả về một DataSet với một cột RowID được thêm vào
Private Function CreateDataSet() As DataSet

    Dim strSQL As String = _
        "SELECT od.OrderID, p.ProductName, od.UnitPrice, od.Quantity, " & _
        "  od.Discount, c.CategoryName " & _
        "FROM [Order Details] od " & _
        "INNER JOIN Products p ON od.ProductID = p.ProductID " & _
        "INNER JOIN Categories c ON p.CategoryID = c.CategoryID " & _
        "GROUP BY od.OrderID, p.ProductName, od.UnitPrice, od.Quantity, " & _
        "  od.Discount, c.CategoryName"

    Dim scnnNW As New SqlConnection(SQL_CONNECTION_STRING)

    Dim scmd As New SqlCommand(strSQL, scnnNW)

    Dim sda As New SqlDataAdapter(scmd)
    Dim ds As New DataSet()
    sda.Fill(ds)
```

```
    ' Thêm một cột RowID phục vụ cho custom paging.
    Dim dt As DataTable = ds.Tables(0)
    dt.Columns.Add("RowID", Type.GetType("System.Int32"))
    dt.AcceptChanges()

    Return ds
End Function

' Thủ tục này trả về một DataView được sắp xếp và đánh số lấy từ Application
Cache. Nếu DataView không tìm thấy trong Cache, nó được tạo mới.
Private Sub GetDataSource()

    ' Nếu tìm thấy DataView trong cache, sử dụng nó
    If Not IsNothing(Cache("dvOrderDetails")) Then
        ' Tất cả các item trong Cache đều có kiểu Object, do đó phải ép
        ' kiểu trước khi sử dụng nó.
        dvOrderDetails = CType(Cache("dvOrderDetails"), DataView)
    Else ' DataView không tìm thấy trong cache
        ' Tạo mới một DataView
        dvOrderDetails = CreateDataSet().Tables(0).DefaultView

        ' Đưa dataview vào Cache
        Cache("dvOrderDetails") = dvOrderDetails
    End If

    ' Sắp xếp DataView dựa trên expression lưu trong control HtmlInputHidden.
    dvOrderDetails.Sort = htmlHiddenSortExpression.Value

    ' Bởi vì các số RowID đã được thêm vào, nếu không đánh số lại thì sau một
    ' sắp xếp mới các số này sẽ không đúng thứ tự và không sử dụng được
    ' cho custom paging
    If chkUseCustomPaging.Checked Then
        RenumberRowsForPaging(dvOrderDetails)
    End If

End Sub
```

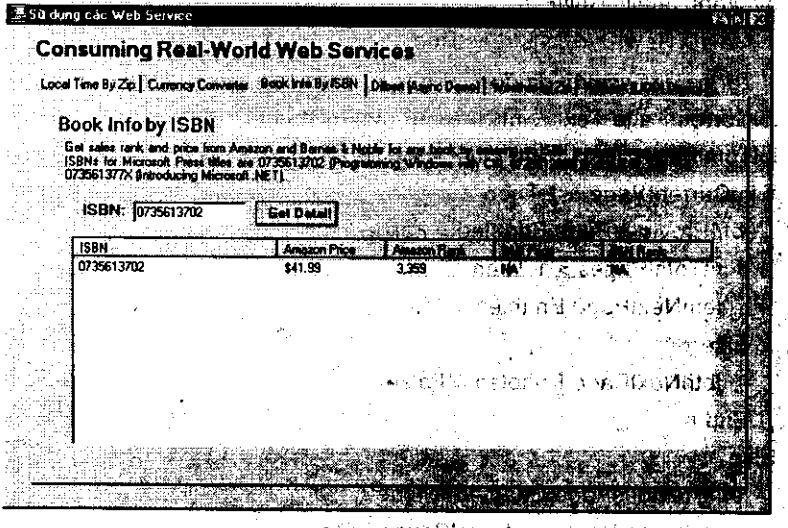
```
' Thủ tục này đánh số cột RowID trong DataView sau khi được sắp xếp.
' Chú ý DataView được truyền vào ByRef để tránh sử dụng thêm bộ nhớ.
Private Sub RenumberRowsForPaging(ByRef dv As DataView)
    ' Reset RowFilter trên DataView trước khi áp dụng một RowFilter mới.
    ' Nếu không, RowFilter cũ vẫn còn tác dụng, và dv.Count bằng với PageSize,
    ' sẽ gây ra lỗi khi chạy.
    dvOrderDetails.RowFilter = ""

    Dim drwv As DataRowView
    Dim i As Int32
    For Each drwv In dv
        drwv("RowID") = i
        i += 1
    Next
End Sub

' Thủ tục này thiết lập các control phân trang – cho cả cách chuẩn hoặc
' cách tùy biến
Private Sub SetPagingControls()

    ' Nếu không sử dụng custom paging thì bật tính năng phân trang của DataGrid,
    ' ẩn các control phân trang của custom paging và thoát thủ tục.
    If Not chkUseCustomPaging.Checked Then
        pnlCustomPaging.Visible = False
        grdOrderDetails.AllowPaging = True
        Exit Sub
    End If

    ' Phần code còn lại áp dụng cho custom paging
    pnlCustomPaging.Visible = True
    grdOrderDetails.AllowPaging = False
    ' Xác định các giá trị RowID cho record đầu tiên và cuối cùng của trang hiện
    ' hành của tập kết quả trên DataGrid.
    Dim strFirstRec As String = _
        ((intCurrentPage - 1) * grdOrderDetails.PageSize).ToString

    Dim strLastRec As String = _
        ((intCurrentPage * grdOrderDetails.PageSize) + 1).ToString
```

```
        ' Xác định tổng số trang của DataGrid. Làm điều này trước khi áp
        ' dụng RowFilter vào DataView.
        Dim intTotalPages As Int32 = _
          CInt(System.Math.Ceiling(dvOrderDetails.Count / grdOrderDetails.PageSize))
        ' Áp dụng một RowFilter mới tính được
        dvOrderDetails.RowFilter = "RowID > " & strFirstRec & _
          " AND RowID < " & strLastRec
        ' Tất cả các control đã được tính toán, hiển thị các control cho custom paging
        lblCurrentPage.Text = intCurrentPage.ToString
        lblTotalPages.Text = intTotalPages.ToString
        If intCurrentPage = 1 Then
          lbtnPreviousPage.Enabled = False
          If intTotalPages > 1 Then
            lbtnNextPage.Enabled = True
          Else
            lbtnNextPage.Enabled = False
          End If
        Else
          lbtnPreviousPage.Enabled = True
          If intCurrentPage = intTotalPages Then
            lbtnNextPage.Enabled = False
          Else
            lbtnNextPage.Enabled = True
          End If
        End If
      End Sub
    End Class
```

Phần VI - Web Services

34. Sử dụng các Web Service

Chương trình này biểu diễn cách sử dụng nhiều Web Service thực tế.



Các chức năng chính:

Chương trình có một TabControl cung cấp 6 ví dụ về sử dụng Web Service, các ví dụ có độ phức tạp tăng dần:

- Lấy thông tin ngày giờ địa phương dựa trên US Zip code. Địa chỉ Web Service: http://www.alethea.net/webservices/LocalTime.asmx
- Chuyển đổng Euro sang tiền tệ của 12 quốc gia khác. Địa chỉ Web Service: http://www.xml-webservices.net/services/conversions/euro_convert/euro_conver.asmx
- Lấy thông tin giá cả, tỉ lệ bán của bất kỳ cuốn sách nào từ Amazon và Barnes & Noble bằng cách nhập vào một số ISBN. Địa chỉ Web Service: http://www.perfectxml.net/WebServices/SalesRankNPrice/BookService.asmx
- Lấy thông tin dự báo thời tiết hiện hành theo US Zip code. Địa chỉ Web Service: http://www.learnxmlws.com/services/weatherRetriever.asmx
- Xem Dilbert cartoon-of-the-Day từ Daily Dilbert Web service. Ví dụ này thực hiện Web Service một cách không đồng bộ. Địa chỉ Web service: http://www.esynaps.com/WebServices/DailyDiblert.asmx

- Tìm hiểu cách sử dụng UDDI để quản lý các lỗi địa chỉ Web service. Cho phép người lập trình thực hiện kế hoạch quay lui (fallback plan). Địa chỉ Web Service: http://www.vbws.com/services/ServerTime.asmx

Chương trình này đòi hỏi "Microsoft Universal Description Discovery and Integration (UDDI) SDK 1.76", có kèm theo trong thư mục của project này, hoặc có thể download từ trang Web của Microsoft: http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28001204

Các Web Service thực tế có thể tìm thấy ở http://www.xmethods.com/

Để sử dụng một Web Service, cần thêm một Web Reference vào Visual Studio .NET project. Trong hộp thoại Add Web Reference gõ vào địa chỉ của Web Service (ví dụ như các địa chỉ ở trên) và nhấn nút **Add Reference**. Thao tác này làm cho VS .NET truy vấn WSDL document và tạo một class ủy quyền (proxy class) như là API cục bộ để người lập trình sử dụng.

Thường thì tên đặt cho namespace của Web Service khó hiểu. Ta có thể thay đổi chúng bằng cách mở cây Web References trong Solution Explorer và nhấn chuột phải vào tên muốn thay đổi, chọn Rename.

Trong Visual Studio .NET có thể sử dụng UDDI để tìm các Web Service có sẵn. Nhấn vào liên kết "XML Web Services" trên Start Page. Nếu không thấy Start page có thể hiển thị nó bằng chọn menu Help | Show Start Page.

Project: **Su dung Cac Web Service**

Chú ý: Project đã thêm các Web References ở trên vào và với tên các namespace lần lượt là: Alethea, XML-WebServices, PerfectXML, LearnXMLWS, Esynaps, VBWS.

frmMain.vb

```
Option Strict On
Imports System.IO
Imports System.Net
Imports System.Threading
Imports Microsoft.Uddi

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Const WS_DATA_RETRIEVAL_ERROR As String = _
        "An error occurred trying to retrieve the data from the Web Service." & _
        "The Web service may currently be down. You might attempt to access " & _
        "it directly: "
```

```
    Const WS_CONNECTION_STATUS As String = _
      "Contacting the Web Service and retrieving data. Please stand by..."

    ' Dành cho tab Fallback (UDDI)
    Const SERVICE_KEY As String = "906e9751-6677-454e-ad7c-dca3c6699ccd"

    Protected frmStatus As frmStatus

' Code phát sinh form được bỏ qua

    ' Sử dụng các class thừa kế WebRequest để lấy một stream trả lời chứa một hình
    ' ảnh sử dụng để tạo một Bitmap cho PictureBox điều kiện thời tiết hiện hành
    Private Function GetImage(ByVal url As String) As System.Drawing.Image
      Dim httpRequest As HttpWebRequest
      Dim httpResponse As HttpWebResponse

        ' Tạo một yêu cầu dùng WebRequestFactory
      httpRequest = CType(WebRequest.Create(url), HttpWebRequest)
      With httpRequest
        .UserAgent = "Mozilla/4.0 (compatible; MSIE 6.0b; Windows NT 5.1)"
        .Method = "GET"
        .Timeout = 10000
      End With

      Try
          ' Lấy Stream trả lời và trả về một Bitmap được phát sinh từ nó.
        httpResponse = CType(httpRequest.GetResponse(), HttpWebResponse)
        Return New Bitmap(httpResponse.GetResponseStream())
      Catch exp As Exception
        MessageBox.Show("An error occurred trying to retrieve the weather " & _
          "conditions picture.", "Web Service Demo Error", _
          MessageBoxButtons.OK, MessageBoxIcon.Error)
      Finally
        httpResponse.Close()
      End Try

    End Function
```

```
' Lấy một địa chỉ (URL) từ Microsoft UDDI service
Public Function GetUrlFromUDDI() As String
    Dim strNewUrl As String = ""
    'Sử dụng đăng ký UDDI của microsoft
    Inquire.Url = "http://uddi.microsoft.com/inquire"
    Try
        Dim gsd As New GetServiceDetail()
        gsd.ServiceKeys.Add(SERVICE_KEY)

        ' Gọi UDDI và nhận về các chi tiết dịch vụ
        Dim sd As ServiceDetail = gsd.Send()
        ' Trả về URL được trả về từ Alethea Local Time Web service.
        strNewUrl = _
            sd.BusinessServices.Item(0).BindingTemplates.Item(0).AccessPoint.Text
        Return strNewUrl

    Catch ex As Exception
        MessageBox.Show("Unable to retrieve an updated Url for the " & _
            "Web Service. Please try again later.", _
            "Web Service UDDI Demo Error", _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        Return strNewUrl
    End Try
End Function

' Tạo ComboBox chứa các loại tiền tệ để chuyển đổng Euro
Private Sub LoadCurrencyConverterComboBox()
    ' Tạo và đưa dữ liệu từ một tài liệu XML vào DataSet
    Dim dsCountries As New DataSet()

    Try
        dsCountries.ReadXml("..\ecc_countries.xml")
    Catch exp As Exception
        MessageBox.Show("Error loading DataSet from XML: " & _
            exp.Message, _
            "Web Service Demo Error", _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        Exit Sub
    End Try
```

```
    ' Kết DataSet vào ComboBox để hiển thị
    With cboConvertTo
      .DataSource = dsCountries.Tables(0)
      .DisplayMember = "Description"
      .ValueMember = "Currency"
    End With
  End Sub

  Sub PersistCurrencyConverterCountriesToXML()
    ShowStatusIndicators("Connecting to the Euro Converter Web Service to " & _
      "download the Euro conversion currencies. Please stand by...")

    ' Tạo một instance cua Web Service proxy class và một DataSet
    Dim wsEuroConverter As New XML_WebServices.EuroConvert()
    Dim dsCountries As DataSet

    Try
      ' Lấy các quốc gia từ Web Service và chuyển qua XML
      dsCountries = wsEuroConverter.CurrenciesToDataSet
      ' File XML được lưu trong thư mục gốc của project (cha của thư mục bin)
      dsCountries.WriteXml("..\ecc_countries.xml")
    Catch exp As Exception
      MessageBox.Show("Error retrieving Euro currencies." & _
        wsEuroConverter.Url, _
        "Web Service Demo Error", _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
      Exit Sub
    Finally
      ResetStatusIndicators()
    End Try
  End Sub

  ' Ẩn form Status được hiển thị trong ShowStatusIndicators()
  Private Sub ResetStatusIndicators()
    Me.Cursor = Cursors.Default
    frmStatus.Hide()
  End Sub
```

```
' Hiển thi trạng thái công việc
Private Sub ShowStatusIndicators(ByVal strMessage As String)
    frmStatus = New frmStatus()
    frmStatus.Show(strMessage)
    Me.Cursor = Cursors.WaitCursor
    Application.DoEvents()
End Sub

' Xử lý nút " Convert" trên tab Currency.
Private Sub btnConvert_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnConvert.Click

    ' Tạo một instance của Web service proxy class
    Dim wsEuroConverter As New XML_WebServices.EuroConvert()

    ShowStatusIndicators(WS_CONNECTION_STATUS)

    ' Lấy dữ liệu từ Web Service
    Try
        lblConvertedAmount.Text = _
            wsEuroConverter.FromEurosToAmount(CLng(txtAmount.Text), _
            cboConvertTo.SelectedValue.ToString).ToString
    Catch exp As Exception
        MessageBox.Show(WS_DATA_RETRIEVAL_ERROR & _
            wsEuroConverter.Url, _
            "Web Service Demo Error", _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        Exit Sub
    Finally
        ResetStatusIndicators()
    End Try
End Sub

' Xử lý nút "Get Cartoon!" trên tab Dilbert. Thủ tục này kết nối vào một Web
' Service một cách không đồng bộ sử dụng phương thức AsyncWaitHandle.WaitOne
' trong namespace System.Threading. người sử dụng có thể nhập vào số giây sẽ
' đợi lấy dữ liệu từ Web Service.
Private Sub btnCartoon_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCartoon.Click
```

```
If txtAsyncWaitPeriod.Text = "" Then
  MessageBox.Show("You must enter a number of seconds you wish to wait.", _
    "Web Service Demo Information", _
    MessageBoxButtons.OK, MessageBoxIcon.Information)
  Return
End If

Dim wsDailyDilbert As New Esynaps.DailyDilbert()

ShowStatusIndicators("Connecting to the Daily Dilbert Web Service and " & _
  "downloading today's cartoon. Please stand by...")

' Lấy dữ liệu từ Web Service. Web Service này cũng có một phương thức
' trả về URL của một hình ảnh.
Dim arrPicture() As Byte
Try
    ' Gọi phương thức Web bất đồng bộ bằng cách gọi phương thức Begin____
    ' của proxy
  Dim result As IAsyncResult = _
    wsDailyDilbert.BeginDailyDilbertImage(Nothing, Nothing)

    ' Đợi một số giây do người sử dụng nhập vào.
  result.AsyncWaitHandle.WaitOne _
    (New TimeSpan(0, 0, CInt(txtAsyncWaitPeriod.Text)), False)

    ' Nếu kết quả hoàn tất đúng giờ, hiển thị cartoon, nếu không, hiển thị thông báo
  If (result.IsCompleted) Then
      ' Đưa mảng byte vào một stream bộ nhớ bằng cách gọi phương thức End__
    arrPicture = _
      CType(wsDailyDilbert.EndDailyDilbertImage(result), Byte())
    Dim ms As New MemoryStream(arrPicture)

      ' Hiển thị trong PictureBox bằng phương thức FromStream
    With picDilbert
      .Image = Image.FromStream(ms)
      .SizeMode = PictureBoxSizeMode.CenterImage
      .BorderStyle = BorderStyle.Fixed3D
    End With
  Else
```

```
            MessageBox.Show("The cartoon was not retrieved in the " & _
               "time you specified.", "Web Service Demo Information", _
               MessageBoxButtons.OK, MessageBoxIcon.Information)
         End If

      Catch exp As Exception
         MessageBox.Show(WS_DATA_RETRIEVAL_ERROR & _
            wsDailyDilbert.Url, _
            "Web Service Demo Error", _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
         Exit Sub
      Finally
         ResetStatusIndicators()
      End Try
   End Sub

   ' Xử lý nút "Get Time!" trên tab Fallback. Thủ tục này cố tình truy xuất đến một địa
   ' chỉ sai. Sau đó nó thực hiện kế hoạch quay lui bằng cách kết nối vào UDDI
   ' Service. Nó giả lập nhận được một địa chỉ đúng của Web service và kết nối lại
   ' để hiển thị kết quả.
   Private Sub btnFallbackTime_Click(ByVal sender As System.Object, ByVal e As
   System.EventArgs) Handles btnFallbackTime.Click

      Dim wsLocalTime As New Alethea.LocalTime()

      ' Đổi thành một địa chỉ sai, do đó phát sinh một WebException khi thực hiện
      ' gọi phương thức này
      wsLocalTime.Url = "http://www.alethea.net"

      ShowStatusIndicators(WS_CONNECTION_STATUS)
      Try
         ' Thực hiện lấy thời gian từ Web service
         lblFallbackTime.Text = _
               wsLocalTime.LocalTimeByZipCode(txtZipCodeForTime.Text)

      Catch expWeb As System.Net.WebException
         ResetStatusIndicators()
         ShowStatusIndicators("Could not call Web service. Attempting to " & _
            "retrieve current Web Service location from UDDI. " & _
            "Please stand by...")
```

```
        Dim strNewUrl As String = GetUrlFromUDDI()

        If strNewUrl.Length > 0 Then

          ResetStatusIndicators()
          ShowStatusIndicators("Updated Url retrieved from UDDI. " & _
            "Reattempting contact...")

          Dim wsNewLocalTime As New VBWS.ServerTime()

          Try
              ' Thực hiện lấy kết quả từ Web Service có địa chỉ được UDDI cung cấp
            lblFallbackTime.Text = wsNewLocalTime.GetTime.ToString

          Catch exp As Exception
            MessageBox.Show(WS_DATA_RETRIEVAL_ERROR & _
              wsNewLocalTime.Url, _
              "Web Service Demo Error", _
              MessageBoxButtons.OK, MessageBoxIcon.Error)
            Exit Sub
          Finally
            ResetStatusIndicators()
          End Try
        End If
      Finally
        ResetStatusIndicators()
      End Try
    End Sub

    ' Xử lý nút "Get Data" trên tab Books.
    Private Sub btnGetBookData_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnGetBookData.Click

      Dim wsSalesRankNPrice As New PerfectXML.SalesRankNPrice()
      Dim strBookData As PerfectXML.All

      ShowStatusIndicators(WS_CONNECTION_STATUS)
      Try
        strBookData = wsSalesRankNPrice.GetAll(txtISBN.Text)
```

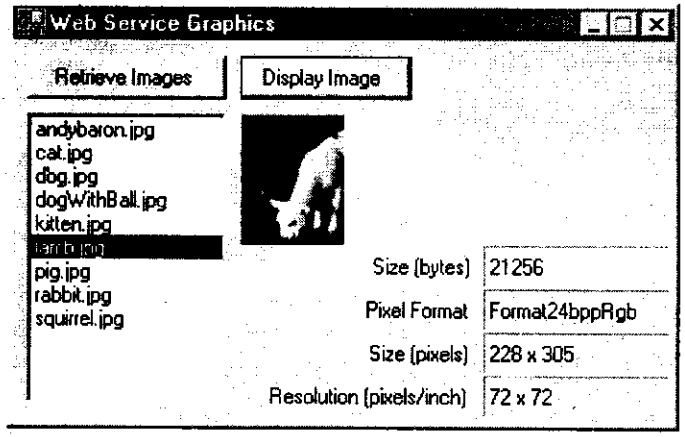
```
    Catch exp As Exception
      MessageBox.Show(WS_DATA_RETRIEVAL_ERROR & _
        wsSalesRankNPrice.Url, _
        "Web Service Demo Error", _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
      Exit Sub
    Finally
      ResetStatusIndicators()
    End Try

    Dim lvItem As New ListViewItem()
    lvItem.Text = txtISBN.Text

    With lvItem.SubItems
      .Add(strBookData.AmazonPrice)
      .Add(strBookData.AmazonSalesRank)
      .Add(strBookData.BNPrice)
      .Add(strBookData.BNSalesRank)
    End With
    lvwBooks.Items.Add(lvItem)
  End Sub

  ' Thủ tục xử lý nút "Get Time" trên tab Local Time. Có thể sử dụng Web Service
  ' này chỉ với hai dòng code:
  '
  '  Dim wsLocalTime As New Alethea.LocalTime()
  '  lblTime.Text = wsLocalTime.LocalTimeByZipCode(txtZipCodeForTime.Text)
  '
  ' Tuy nhiên, đoạn code bên dưới cho thấy toàn bộ thủ tục với cấu trúc bắt lỗi
  Private Sub btnGetTime_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetTime.Click

    ' Tạo một instance của Web service proxy class
    Dim wsLocalTime As New Alethea.LocalTime()
    ShowStatusIndicators(WS_CONNECTION_STATUS)
    Try
      ' Lấy thời gian từ Web service
      lblTime.Text = wsLocalTime.LocalTimeByZipCode(txtZipCodeForTime.Text)
```

```
    Catch exp As Exception
      MessageBox.Show(WS_DATA_RETRIEVAL_ERROR & _
        wsLocalTime.Url, _
        "Web Service Demo Error", _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
      Exit Sub
    Finally
      ResetStatusIndicators()
    End Try
  End Sub

  ' Xử lý nút "Get Weather" trên tab Weather.
  Private Sub btnGetWeather_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetWeather.Click
    ' Ẩn các thông tin về thời tiết cho đến khi dữ liệu được lấy
    pnlWeatherInfo.Visible = False

    ' Tạo một instance của Web service proxy class và một class cung cấp bởi
    ' Web Service để chứa dữ liệu trả về.
    Dim wsWeather As New LearnXMLWS.WeatherRetriever()
    Dim weatherData As LearnXMLWS.CurrentWeather
    ShowStatusIndicators(WS_CONNECTION_STATUS)
    Try
      ' Lấy thông tin thời tiết từ Web Service
      weatherData = wsWeather.GetWeather(txtZipCodeForWeather.Text)

      ' Hiển thị hình ảnh điều kiện thời tiết trong PictureBox
      With picConditions
        .Image = GetImage(weatherData.IconUrl)
        .SizeMode = PictureBoxSizeMode.CenterImage
      End With
    Catch exp As Exception
      MessageBox.Show(WS_DATA_RETRIEVAL_ERROR & _
        wsWeather.Url, _
        "Web Service Demo Error", _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
      Exit Sub
    Finally
      ResetStatusIndicators()
    End Try
```

```
        ' Hiển thị dữ liệu thời tiết
        With weatherData
            lblCurrentTemp.Text = CType(.CurrentTemp, String)
            lblconditions.Text = .Conditions
            lblInfo.Text = " Barometer at " & CType(.Barometer, String) & _
                " and " & .BarometerDirection & vbCrLf & " Humidity at " & _
                CType(.Humidity * 100, String) & "%" & vbCrLf & _
                " Last updated on " & .LastUpdated
        End With
        pnlWeatherInfo.Visible = True
    End Sub

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        ' Currency Convert Web service cung cấp một phương thức trả về một DataSet
        ' chứa danh sách các quốc gia. Danh sách này không thay đổi thường xuyên
        ' và được lưu vào một file. Đưa dữ liệu vào ComboBox trước để giảm thời gian
        ' đợi khi chọn tab Currency.
        LoadCurrencyConverterComboBox()
    End Sub
End Class
```

35. Truyền dữ liệu nhị phân thông qua HTTP

Ví dụ này chỉ cách truyền dữ liệu nhị phân (trong trường hợp này là các hình ảnh) trên một Web service.



Các chức năng chính:

Trong ví dụ này chứa hai project:

- **Web Service Graphics** (trong thư mục Client), project này có chứa một Web Reference cung cấp bởi GraphicServer Project.
- **GraphicsServer** (trong thư mục WebService). Tạo một Virtual Directory (thư mục ảo) trong IIS để ánh xạ đến thư mục này, tên thư mục ảo (alias) là: GraphicsServer.

GraphicsServer Web Service trả về các hình ảnh, và ứng dụng Windows Web Service Graphics sử dụng các phương thức của Web Service. Cụ thể là trang ImageService.asmx cung cấp các phương thức sau:

- **Browser**: Trả về một mảng các đối tượng ImageInfo, dữ liệu là thông tin các hình ảnh tìm thấy trong thư mục .\Images.
- **GetThumbnail**: Trả về một mảng các byte chứa một hình ảnh nhỏ (cao 80 pixel, bề rộng theo tỷ lệ). Truyền vào một chuỗi chứa tên file hình ảnh.
- **GetImage**: Trả về một mảng các byte chứa hình ảnh đầy đủ, phải truyền vào một tên file.

Solution: **Truyen du lieu nhi phan**

Project: **GraphicsServer**

ImageService.asmx

```
<%@WebService Language="vb" Codebehind="ImageService.asmx.vb"
  Class="VB.NET.HowTo.ImageService" %>
```

ImageService.asmx.vb

```
Option Strict On
Imports System.Web.Services
Imports System.IO
Imports System.Drawing

<WebService()> _
Public Class ImageService
    Inherits System.Web.Services.WebService

' Code tự động phát sinh được bỏ qua
```

```
Public Class ImageInfo
    Public Name As String
    Public Size As Long
    Public Height As Integer
    Public Width As Integer
    Public HorizontalResolution As Single
    Public VerticalResolution As Single
    Public PixelFormat As Imaging.PixelFormat
    Public Thumbnail As Byte()
End Class

' Trả về một mảng chứa các tên file hình ảnh
<WebMethod()> _
Public Function Browse() As ImageInfo()
    Dim i As Integer
    Dim fi As FileInfo
    Dim aImages As ImageInfo()

    Dim di As New DirectoryInfo(Server.MapPath("./Images"))
    Dim afi As FileInfo() = di.GetFiles("*.jpg")

    ReDim Preserve aImages(afi.Length - 1)

  For Each fi In afi
        aImages(i) = New ImageInfo()
        aImages(i).Name = fi.Name
        aImages(i).Size = fi.Length
        FillImageInfo(aImages(i), fi.Name)
        i += 1
    Next
    Return aImages
End Function

Private Sub FillImageInfo(ByVal Info As ImageInfo, ByVal FileName As String)
    Dim ms As New MemoryStream()
    Dim bmp As New Bitmap(Server.MapPath("./Images/" & FileName))

    Info.Width = bmp.Width
    Info.Height = bmp.Height
```

```
    Info.HorizontalResolution = bmp.HorizontalResolution
    Info.VerticalResolution = bmp.VerticalResolution
    Info.Thumbnail = GetThumbnailInfo(bmp)
    Info.PixelFormat = bmp.PixelFormat
    bmp.Dispose()
End Sub

' Trả về một hình ảnh nhỏ
<WebMethod()> _
Public Function GetThumbnail(ByVal FileName As String) As Byte()
    Dim bmp As New Bitmap(Server.MapPath("./Images/" & FileName))
    Return GetThumbnailInfo(bmp)
End Function

Private Function ThumbnailCallback() As Boolean

    ' Ta phải cung cấp thủ tục này, mặc dù thực sự không sử dụng nó.
    Return False
End Function

Private Function GetThumbnailInfo(ByVal bmp As Bitmap) As Byte()
    Dim ms As New MemoryStream()
    Dim intWidth As Integer
    Dim intHeight As Integer
    Dim decRatio As Decimal

    ' Chọn chiều cao là 80, nhưng có thể thay bằng giá trị tùy ý
    Const THUMBNAIL_HEIGHT As Integer = 80

    intWidth = bmp.Width
    intHeight = bmp.Height
    decRatio = CDec(intWidth / intHeight)

    Dim bmpTemp As Bitmap
    bmpTemp = CType(bmp.GetThumbnailImage(CInt(decRatio * _
                THUMBNAIL_HEIGHT), THUMBNAIL_HEIGHT, _
            AddressOf ThumbnailCallback, IntPtr.Zero), System.Drawing.Bitmap)
    bmpTemp.Save(ms, Imaging.ImageFormat.Jpeg)
    Return ms.ToArray
End Function
```

```
    ' Trả về một hình ảnh đầy đủ
    <WebMethod()> _
    Public Function GetImage(ByVal FileName As String) As Byte()
        Return ReadFile(Server.MapPath("./Images/" & FileName))
    End Function

    Private Shared Function ReadFile(ByVal FilePath As String) As Byte()
        Dim fs As FileStream

        Try
            ' Đọc và trả về nội dung của file
            fs = File.Open(FilePath, FileMode.Open, FileAccess.Read)

            Dim lngLen As Long = fs.Length
            Dim abytBuffer(CInt(lngLen - 1)) As Byte
            fs.Read(abytBuffer, 0, CInt(lngLen))
            Return abytBuffer
        Finally
            ' Đảm bảo là file được đóng, cho dù có lỗi xảy ra
            fs.Close()
        End Try
    End Function
End Class
```

Project: **Web Service Graphics**

frmMain.vb

```
Option Strict On
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

    Private maInfo As GraphicsServer.ImageInfo()

    Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnDisplay.Click
        DisplayImage()
    End Sub
```

```
Private Sub btnRetrieve_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRetrieve.Click
    Dim imgsvc As GraphicsServer.ImageService
    Dim info As GraphicsServer.ImageInfo

    Try
      Me.Cursor.Current = Cursors.WaitCursor
       ' Lấy mảng thông tin các hình ảnh
      imgsvc = New GraphicsServer.ImageService()
      maInfo = imgsvc.Browse()

       ' Xóa nội dung trong danh sách, sau đó cho thông tin vào
      lstImages.Items.Clear()

      For Each info In maInfo
        lstImages.Items.Add(info.Name)
      Next

    Catch Exp As Exception
      MessageBox.Show(Exp.Message, Me.Text)

    Finally
      Me.Cursor.Current = Cursors.Default
      imgsvc.Dispose()
    End Try
End Sub

Private Sub lstImages_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles lstImages.SelectedIndexChanged
    Dim imgsvc As GraphicsServer.ImageService
    Dim Info As GraphicsServer.ImageInfo

     ' Hiển thị thông tin về hình ảnh được chọn
    Try
      imgsvc = New GraphicsServer.ImageService()
      Info = maInfo(lstImages.SelectedIndex)
      lblSize.Text = Info.Size.ToString
      lblPixelFormat.Text = Info.PixelFormat.ToString
```

```
        lblResolution.Text = String.Format("{0} x {1}", Info.HorizontalResolution,
Info.VerticalResolution)
        lblImageSize.Text = String.Format("{0} x {1}", Info.Width, Info.Height)
        picImage.Image = GetImage(Info.Thumbnail)
        btnDisplay.Enabled = True

    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)

    Finally
        imgsvc.Dispose()
    End Try
End Sub

Private Sub picImage_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles picImage.DoubleClick
    DisplayImage()
End Sub

Private Sub DisplayImage()
    Dim frm As frmImage
    Dim imgsvc As GraphicsServer.ImageService
    Dim Info As GraphicsServer.ImageInfo
    Dim bmp As Bitmap
    Dim strName As String

    ' Hiển thị hình ảnh được chọn
    Try
        imgsvc = New GraphicsServer.ImageService()
        strName = maInfo(lstImages.SelectedIndex).Name
        bmp = GetImage(imgsvc.GetImage(strName))

        ' Tạo một instance của form hiển thị hình ảnh
        frm = New frmImage()
        frm.Text = strName
        frm.Image = bmp
        frm.ClientSize = New Size(bmp.Width, bmp.Height)
        frm.Show()
    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
```

```
    Finally
      imgsvc.Dispose()
    End Try
  End Sub

  Private Function GetImage(ByVal abyt() As Byte) As Bitmap
    ' Đưa vào một mảng các byte, trả về một Bitmap thực sự. Thực hiện bằng
    ' cách tạo mới một đối tượng MemoryStream dựa trên mảng các byte, sau đó
    ' tạo một Bitmap mới dựa trên memory stream.
    Try
      Return New Bitmap(New MemoryStream(abyt))
    Catch exp As Exception
      MessageBox.Show(exp.Message, Me.Text)
    End Try
  End Function
End Class
```

frmImage.vb

```
Option Strict On
Public Class frmImage
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

  Private ClientSizeOriginal As Size

  Public WriteOnly Property Image() As Image
    ' Thuộc tính Public WriteOnly cho phép form cha quy định hình ảnh cho
    ' form này mà không biết gì khác về form này
    Set(ByVal Value As Image)
      Try
        picImage.Image = Value
      Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
      End Try
    End Set
  End Property
```

```
Private Sub frmImage_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ' Lưu lại kích thước cửa sổ, để sử dụng sau
    ClientSizeOriginal = Me.ClientSize
End Sub

Private Sub mnuClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuClose.Click
    Me.Close()
End Sub

Private Sub mnuSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuSave.Click
    ' Lưu hình ảnh với một trong các định dạng có sẵn
    Try
        With sdlgImage
            .Filter = "Bitmap (*.bmp)|*.bmp|JPEG (*.jpg, *.jpeg)|*.jpg;*.jpeg|GIF
(*.gif)|*.gif|TIFF (*.tif, *.tiff)|*.tif;*.tiff|PNG (*.png)|*.png"
            .AddExtension = True
            .OverwritePrompt = True
            .CheckPathExists = True
            .ValidateNames = True
            .Title = "Save Image"
            If .ShowDialog() = DialogResult.OK Then
                Dim bmp As New Bitmap(picImage.Image)
                Dim fmt As Imaging.ImageFormat
                Select Case .FilterIndex
                    Case 1
                        fmt = Imaging.ImageFormat.Bmp
                    Case 2
                        fmt = Imaging.ImageFormat.Jpeg
                    Case 3
                        fmt = Imaging.ImageFormat.Gif
                    Case 4
                        fmt = Imaging.ImageFormat.Tiff
                    Case 5
                        fmt = Imaging.ImageFormat.Png
                End Select
                bmp.Save(.FileName, fmt)
            End If
```

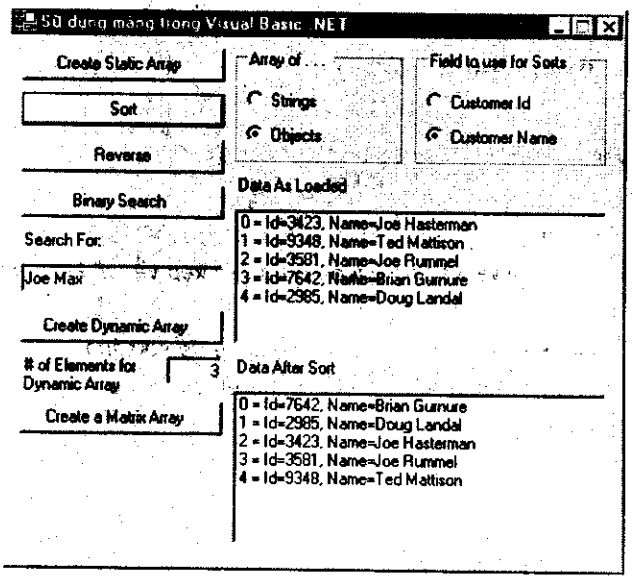
```
        End With
      Catch exp As Exception
        MessageBox.Show(exp.Message, "Saving Image")
      End Try
    End Sub

    Private Sub mnuStretch_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuStretch.Click
      Me.FormBorderStyle = FormBorderStyle.Sizable
    End Sub

    Private Sub mnuZoom50_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuZoom50.Click

      SetClientSize(50)
    End Sub

    Private Sub mnuZoom100_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuZoom100.Click
      SetClientSize(100)
    End Sub

    Private Sub mnuZoom200_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuZoom200.Click
      SetClientSize(200)
    End Sub

    Private Sub HandleZoom(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles mnuZoom100.Click, mnuZoom200.Click,
    mnuZoom50.Click, mnuStretch.Click
      ' Quản lý các dấu check bên cạnh các menu item
      ' Xóa tất cả các dấu check
      mnuZoom50.Checked = False
      mnuZoom100.Checked = False
      mnuZoom200.Checked = False
      mnuStretch.Checked = False

      ' Đánh dấu menu item được chọn
      CType(sender, MenuItem).Checked = True
    End Sub
```

```
    Private Sub SetClientSize(ByVal intPercent As Integer)
        ' Cung cấp vào số phần trăm (50, 100, hay 200) để quy định lại kích thước
        ' dựa trên kích thước gốc.

        Me.FormBorderStyle = FormBorderStyle.FixedSingle
        Me.ClientSize = New Size(CInt(intPercent / 100 * ClientSizeOriginal.Width),
    CInt(intPercent / 100 * ClientSizeOriginal.Height))
    End Sub
End Class
```

Phần VII - Language

36. Sử dụng mảng - Array

Chương trình này biểu diễn một số tính năng của các mảng trong Visual Basic .NET



Các chức năng chính:

Chương trình chỉ cách làm việc với mảng các kiểu giá trị, các chuỗi, và mảng các kiểu tham chiếu, mảng các đối tượng tự tạo đối tượng tự tạo Customer. Customer cung cấp hai trường public là **Id** và **Name**. Có sáu nút bấm thực hiện các chức năng khác nhau:

- **Create Static Array** – tạo một mảng sử dụng cú pháp mới { }. Phần Array of cung cấp các chọn lựa là mảng chứa các chuỗi hay các instance của các đối tượng (Customer). Sau đó mảng được hiển thị trong ListBox.
- **Sort** – Đưa dữ liệu vào mảng như trong ví dụ trên, sau đó sử dụng phương thức Sort (sắp xếp) của class Array để áp dụng vào dữ liệu được đưa vào ListBox lstAfter.
- **Reverse** – Làm việc giống như nút Sort nhưng với thứ tự sắp ngược lại.
- **Binary Search** – Thực hiện tìm kiếm nhị phân, đòi hỏi mảng phải được sắp.
- **Create Dynamic Array** – Sử dụng giá trị trong TextBox txtLength để ReDim (định lại kích thước) một mảng.
- **Tạo mảng ma trận** – Tạo một mảng hai chiều chứa hai cột và ba dòng, sau đó hiển thị trong một ListBox.

Chú ý: Để cho code rõ ràng, các mảng được tạo tại mỗi thủ tục khi nhấn chuột trên các nút.

Project: **Su dung mang - Array**

frmMain.vb

```
Option Strict On
Public Class Customer
    Implements IComparable ' Cho phép hỗ trợ sắp xếp và tìm kiếm nhị phân

    ' Sử dụng để xác định trường nào được sử dụng để so sánh
    Private Shared m_CompareField As CompareField
    Public Id As Integer
    Public Name As String

    Public Enum CompareField
        Name
        Id
    End Enum

    Shared Sub New()
        ' Quy định trường so sánh mặc nhiên
        m_CompareField = CompareField.Name
    End Sub
```

```
Public Shared Sub SetCompareKey(ByVal CompareKey As CompareField)
    ' Thay đổi trường so sánh
    m_CompareField = CompareKey
End Sub

Public Sub New()
    Me.New(-1, "No Name")
End Sub

Public Sub New(ByVal NewId As Integer, ByVal NewName As String)
    Me.Id = NewId
    Me.Name = NewName
End Sub

Public Function CompareTo(ByVal obj As Object) As Integer Implements
System.IComparable.CompareTo

    ' Trước tiên kiểm tra để bảo đảm là so sánh với một Customer khác
    If TypeOf obj Is Customer Then
        ' Tạo tham chiếu mạnh đến Customer
        Dim c As Customer = CType(obj, Customer)
        ' Xác định so sánh theo Name hoặc Id
        If Customer.m_CompareField = CompareField.Name Then
            If c.Name = Me.Name Then
                Return 0
            ElseIf c.Name < Me.Name Then
                Return 1
            Else
                Return -1
            End If
        Else
            If c.Id = Me.Id Then
                Return 0
            ElseIf c.Id < Me.Id Then
                Return 1
            Else
                Return -1
            End If
        End If
    Else
```

```
        Throw New ArgumentException("Customers can only be compared against
    other customers. The object being passed in was a " & obj.GetType.ToString())
      End If
    End Function

    Public Overrides Function ToString() As String
      ' Thông thường ToString trả về một tên đầy đủ của kiểu dữ liệu. Trong ví dụ
      ' này ta override nó để trả về một chuỗi.
      Return String.Format("Id={0}, Name={1}", Me.Id, Me.Name)
    End Function

End Class

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Dùng để xác định ListBox nào phương thức DisplayArrayData sẽ sử dụng
    Private Enum WhichListBox
      BoxOne
      BoxTwo
    End Enum
' Code phát sinh form được bỏ qua

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
      Me.optStrings.Checked = True
      Me.optName.Checked = True
    End Sub

    Private Sub cmdBinarySearch_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles cmdBinarySearch.Click
      Dim strMsg As String
      Dim intPos As Integer
      ' Bảo đảm có một giá trị để sử dụng làm điều kiện tìm kiếm
      If Me.txtBSearchFor.Text.Length = 0 Then
        MessageBox.Show("Please enter a value to search for", "Binary Search",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Me.txtBSearchFor.Select()
        Exit Sub
      End If
```

```
    Dim strDataToFind As String = Me.txtBSearchFor.Text
    Me.lstAfter.Items.Clear()

    If Me.optStrings.Checked Then
        Dim strData() As String = {"Joe Hasterman", "Ted Mattison", "Joe Rummel",
"Brian Gurnure", "Doug Landal"}

        ' Sắp xếp mảng,  nếu không phương thức BinarySearch sẽ không làm việc
        Array.Sort(strData)

        ' Hiển thị dữ liệu đã sắp xếp
        DisplayArrayData(strData)

        ' Thực hiện tìm kiếm
        intPos = Array.BinarySearch(strData, strDataToFind)

        If intPos > 0 Then
            strMsg = String.Format("The value {0} was found in the array at position
{1}.", strDataToFind, intPos.ToString())
        Else
            ' Nếu không tìm thấy, một số âm được trả về. Trị tuyệt đối của số này
            ' cho biết vị trí của giá trị cần tìm "nếu" như nó tồn tại trong mảng.
            Dim intBWC As Integer = (Not intPos)

            If intBWC = 0 Then
                strMsg = String.Format("The value {0} was NOT found in the array. If it
did exist it would be at position {1} before {2}", strDataToFind, intBWC.ToString(),
strData(1))
            Else
                Dim strItems() As String = {strDataToFind, intBWC.ToString(),
strData(intBWC - 1), strData(intBWC)}
                strMsg = String.Format("The value {0} was NOT found in the array. If it
did exist it would be at position {1} between {2} and {3}.", strItems)
            End If
        End If

        MessageBox.Show(strMsg, "Binary Search Results", MessageBoxButtons.OK,
MessageBoxIcon.Information)
    Else
```

```
        Dim objData() As Customer = {New Customer(3423, "Joe Hasterman"), New
    Customer(9348, "Ted Mattison"), New Customer(3581, "Joe Rummel"), New
    Customer(7642, "Brian Gurnure"), New Customer(2985, "Doug Landal")}

        Array.Sort(objData)

        DisplayArrayData(objData)

        ' Khi tìm kiếm trong một mảng các đối tượng, ta cần sử dụng một kiểu đối
        ' tượng tương thích. Trong trường hợp này ta sử dụng giá trị trong
        ' txtBSearchFor làm Name của Customer và cho một Id giả. Một điểm chính
        ' để tìm kiếm nhị phân làm việc là đối tượng phải sử dụng IComparable.
        Dim c As New Customer(1, Me.txtBSearchFor.Text)

        intPos = Array.BinarySearch(objData, c, Nothing)

        If intPos > 0 Then
            strMsg = String.Format("The value {0} was found in the array at position
    {1}.", strDataToFind, intPos.ToString())
        Else
            Dim intBWC As Integer = (Not intPos)

            If intBWC = 0 Then
                strMsg = String.Format("The value {0} was NOT found in the array. If it
    did exist it would be at position {1} before {2}", strDataToFind, intBWC.ToString(),
    objData(1).Name)

            Else

                Dim strItems() As String = {strDataToFind, intBWC.ToString(),
    objData(intBWC - 1).name, objData(intBWC).name}
                strMsg = String.Format("The value {0} was NOT found in the array. If it
    did exist it would be at position {1} between {2} and {3}.", strItems)
            End If
        End If

        MessageBox.Show(strMsg, "Binary Search Results", MessageBoxButtons.OK,
    MessageBoxIcon.Information)
    End If

End Sub
```

```
Private Sub cmdCreateDynamic_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCreateDynamic.Click
    ' Thủ tục này tạo một mảng động sử dụng giá trị trong txtLength làm số phần tử
  If Me.optStrings.Checked Then
    Dim strDynamicData() As String
    ReDim strDynamicData(System.Convert.ToInt32(Me.txtLength.Text) - 1)
    Dim i As Integer
    Dim strInput As String
    For i = 0 To strDynamicData.Length - 1
      strDynamicData(i) = InputBox("Enter a string", i.ToString(), "None " & i)
    Next
    DisplayArrayData(strDynamicData)
  Else
    Dim strDynamicData() As Customer
    ReDim strDynamicData(System.Convert.ToInt32(Me.txtLength.Text) - 1)
    Dim i As Integer
    Dim strInput As String

    For i = 0 To strDynamicData.Length - 1
      strDynamicData(i) = New Customer()
      strDynamicData(i).Id = ((i + 1) * 10)
      strDynamicData(i).Name = InputBox("Enter a string", ("Item " & (i + 1)),
("None " & i + 1))
    Next
    DisplayArrayData(strDynamicData)
  End If
End Sub

Private Sub cmdCreateMatrix_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCreateMatrix.Click
    ' Thủ tục này dùng cú pháp khởi tạo để tạo một mảng hai chiều (ma trận)
  Dim strMatrix(,) As String = {{"Bob", "Carol"}, {"Ted", "Alice"}, {"Joe", "Lisa"}}
  Me.DisplayArrayData(strMatrix)
End Sub

Private Sub cmdCreateStatic_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCreateStatic.Click
    ' Thủ tục này tạo một mảng chứa các chuỗi hoặc chứa các đối tượng Customer.
    ' Chú ý là mảng này có thể định lại kích thước bằng cách sử dụng ReDim
```

```
    If Me.optStrings.Checked Then
        Dim strData() As String = {"Joe Hasterman", "Ted Mattison", "Joe Rummel",
"Brian Gurnure", "Doug Landal"}
        DisplayArrayData(strData)
    Else
        ' Dòng lệnh dưới đây cho thấy điểm mạnh của các đối tượng trong .NET là
        ' có thể tạo với tham số. Điều này cho phéo tạo một mảng đối tượng với
        ' một dòng lệnh.
        Dim objData() As Customer = {New Customer(3423, "Joe Hasterman"), New
Customer(9348, "Ted Mattison"), New Customer(3581, "Joe Rummel"), New
Customer(7642, "Brian Gurnure"), New Customer(2985, "Doug Landal")}
        DisplayArrayData(objData)
    End If
End Sub

Private Sub cmdReverse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdReverse.Click
    ' Đảo thứ tự các phần tử trong mảng để có thứ tự ngược
    Me.lblDataAfter.Text = "Data After Reverse"
    If Me.optStrings.Checked Then
        Dim strData() As String = {"Joe Hasterman", "Ted Mattison", "Joe Rummel",
"Brian Gurnure", "Doug Landal"}
        Array.Reverse(strData)
        DisplayArrayData(strData, WhichListBox.BoxTwo)
    Else
        Dim objData() As Customer = {New Customer(3423, "Joe Hasterman"), New
Customer(9348, "Ted Mattison"), New Customer(3581, "Joe Rummel"), New
Customer(7642, "Brian Gurnure"), New Customer(2985, "Doug Landal")}
        Array.Reverse(objData)
        DisplayArrayData(objData, WhichListBox.BoxTwo)
    End If

End Sub

Private Sub cmdSort_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSort.Click
    ' Để thủ tục này làm việc, các item trong mảng phải hỗ trợ so sánh, bằng cách
    ' cho các đối tượng sử dụng interface IComparable

    Me.lblDataAfter.Text = "Data After Sort"
```

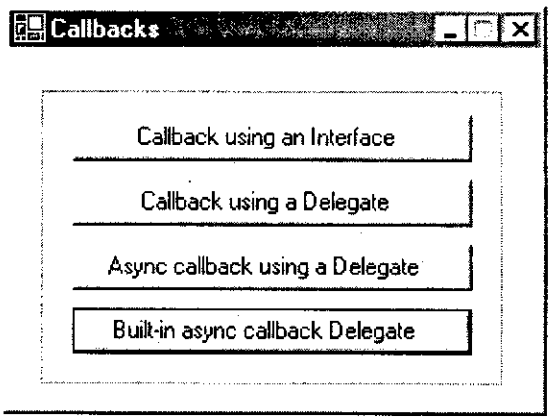
```
    If Me.optStrings.Checked Then
        Dim strData() As String = {"Joe Hasterman", "Ted Mattison", "Joe Rummel",
"Brian Gurnure", "Doug Landal"}
        Array.Sort(strData)
        DisplayArrayData(strData, WhichListBox.BoxTwo)
    Else
        Dim objData() As Customer = {New Customer(3423, "Joe Hasterman"), New
Customer(9348, "Ted Mattison"), New Customer(3581, "Joe Rummel"), New
Customer(7642, "Brian Gurnure"), New Customer(2985, "Doug Landal")}
        Array.Sort(objData)
        DisplayArrayData(objData, WhichListBox.BoxTwo)
    End If
End Sub

Private Sub txtLength_Validating(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles txtLength.Validating
    Dim txt As TextBox = CType(sender, TextBox)

    If Not IsNumeric(txt.Text) Then
        ' Bảo đảm là chỉ có các số được nhập vào TextBox này
        MessageBox.Show(txt.Text & " is not a valid number.", "Non-Numeric Data",
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        txt.Text = 1.ToString()
    Else
        Try
            ' Bảo đảm là giá trị nằm trong vùng giá trị của kiểu Integer
            Dim i As Integer = CInt(txt.Text)
        Catch exp As OverflowException
            Dim strMsg As String
            strMsg = String.Format("The value you entered, {0} is too large. Valid values
are between -{1} and {2}", txt.Text, Integer.MaxValue, Integer.MaxValue)

            MessageBox.Show(strMsg, "Value Overflow", MessageBoxButtons.OK,
MessageBoxIcon.Exclamation)
            txt.Text = 1.ToString()
        End Try
    End If

End Sub
```

```
Private Sub CompareKeyCheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles optId.CheckedChanged, optId.CheckedChanged
    Dim opt As RadioButton = CType(sender, RadioButton)
    ' Thay đổi trường sử dụng bởi đối tượng Customer khi sắp xếp hoặc tìm kiếm
    If opt.Name = "optId" Then
        ' SetCompareKey là một phương thức Shared sẽ ảnh hưởng tất cả các
        ' instacnce có kiểu Customer trong AppDomain.
        Customer.SetCompareKey(Customer.CompareField.Id)
    Else
        Customer.SetCompareKey(Customer.CompareField.Name)
    End If
    Me.lstAfter.Items.Clear()
    Me.lblDataAfter.Text = "Data Cleared. Sort Again"
    Me.cmdSort.Select()
End Sub

Private Sub DataTypeCheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles optStrings.CheckedChanged,
optObjects.CheckedChanged
    ' Enable grpCompareField chỉ khi mảng chứa các đối tượng
    Dim opt As RadioButton = CType(sender, RadioButton)
    Me.grpCompareField.Enabled = Not (opt.Name = "optStrings")
    Me.lstArrayData.Items.Clear()
    Me.lstAfter.Items.Clear()
End Sub

Private Sub DisplayArrayData(ByVal arr As Array)
    Me.DisplayArrayData(arr, WhichListBox.BoxOne)
End Sub

Private Sub DisplayArrayData(ByVal arr As Array, ByVal ListBox As WhichListBox)
    Dim i As Integer
    Dim u As Integer = (arr.Length - 1)
    Dim strOutput As String
    Dim lst As ListBox
    If ListBox = WhichListBox.BoxOne Then
        lst = Me.lstArrayData
    Else
        lst = Me.lstAfter
    End If
```

```
        lst.Items.Clear()
        ' Thực hiện theo số chiều của mảng
        Select Case arr.Rank
            Case 1
                For i = 0 To u
                    lst.Items.Add(String.Format("{0} = {1}", i, arr.GetValue(i).ToString()))
                Next
            Case 2
                Dim j As Integer
                For i = 0 To (arr.GetLength(0) - 1)
                    For j = 0 To (arr.GetLength(1) - 1)
                        lst.Items.Add(String.Format("({0}, {1}) = {2}", i, j, _
                                                    arr.GetValue(i, j).ToString()))
                    Next j
                Next i
            Case Else
                ' Không cài đặt cho mảng nhiều hơn hai chiều
                lst.Items.Add(String.Format("The array received has too many dimensions
    ({0})", arr.Rank))
        End Select
    End Sub
End Class
```

37. Callback

Chương trình này biểu diễn cách thực hiện callback sử dụng cả hai phương pháp interface và delegate.



Các chức năng chính:

- Ứng dụng này cho phép người sử dụng thực hiện nhiều cách gọi kể cả gọi trở lại vào client trong quá trình thực thi.
- Nút đầu tiên thực hiện kiểu gọi này bằng cách sử dụng một interface, và đòi hỏi bước đăng ký và hủy đăng ký.
- Nút thứ hai thực hiện cùng công việc sử dụng delegate, và cũng đòi hỏi đăng ký và hủy đăng ký.
- Nút thứ ba làm việc tương tự như nút thứ hai, ngoại trừ callback được thực hiện không đồng bộ.
- Nút thứ tư gọi không đồng bộ sử dụng delegate AsyncCallback để gọi trở lại client.

Project: **Callbacks**

ICallback.vb

```
Option Strict On

Interface ICallback
    Sub CallbackMethod()
End Interface
```

Class1.vb

```
Option Strict On
Delegate Sub Delegate1()

Friend Class Class1

    ' Trường này chứa một tham chiếu interface để gọi trở lại client
    Private icb As ICallback
    ' Trường này chứa một tham chiếu delegate để gọi trở lại client
    Private del1 As Delegate1

    Public Sub RegisterInterFaceForCallback(ByVal cb As ICallback)
        ' Phương thức này nhận một tham chiếu cho client
        ' Đăng ký client cho callback
        icb = cb
    End Sub
```

```
Public Sub UnRegisterInterfaceForCallback()

    ' Hủy đăng ký client
    icb = Nothing

End Sub

Public Sub UseInterfaceCallback()
    ' Phương thức này sử dụng client đã đăng ký tham chiếu đến call back
    ' Bảo đảm là đã có đăng ký client
    If Not icb Is Nothing Then
        'Gọi vào client code
        icb.CallbackMethod()
    End If
End Sub

Public Sub RegisterDelegateForCallback(ByVal d As Delegate1)
    ' Phương thức này nhận một tham chiếu cho client

    'Đăng ký client cho callback
    del1 = d
End Sub

Public Sub UnRegisterDelegateForCallback()

    ' Hủy đăng ký client
    del1 = Nothing
End Sub

Public Sub UseDelegateCallback()
    ' Phương thức này sử dụng client đã đăng ký tham chiếu đến call back

    ' Bảo đảm đã có đăng ký client
    If Not del1 Is Nothing Then
        ' Call back vào client code
        del1.Invoke()
    End If
End Sub
```

```
    Public Sub UseAsyncDelegateCallback()
        ' Phương thức này sử dụng client đã đăng ký tham chiếu đến call back vào client
        ' một cách không đồng bộ

        If Not del1 Is Nothing Then
            ' Call back bất đồng bộ vào client code, có thể bỏ qua các đối số
            del1.BeginInvoke(Nothing, Nothing)
        End If
    End Sub

    Public Sub UseBuiltInDelegateCallback()
        ' Phương thức này không cần sử dụng một client đã đăng ký.
        ' Callback là một phần cơ sở của delagate.
        ' không cần code

    End Sub

End Class
```

frmMain.vb

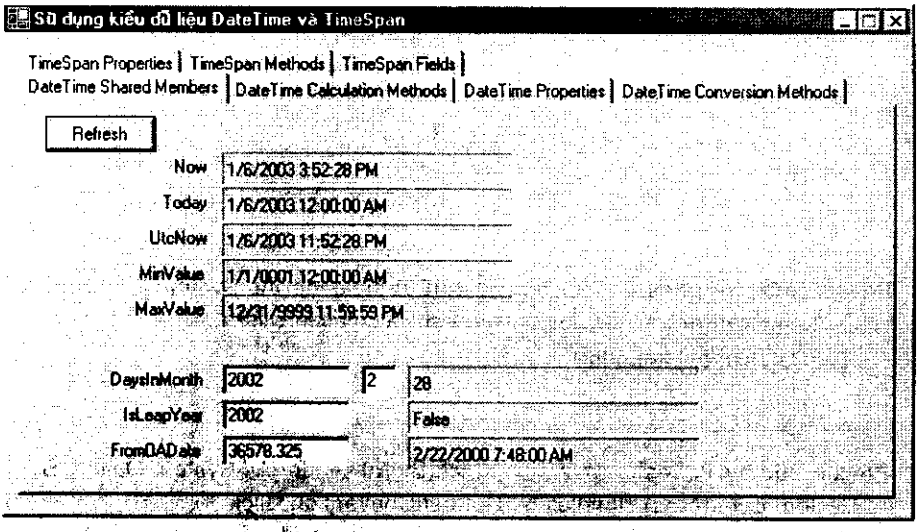
```
Option Strict On

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Sử dụng interface ICallback
    Implements ICallback

' Code phát sinh form được bỏ qua

    Public Sub CallbackMethod() Implements ICallback.CallbackMethod
        ' Phương thức này được gọi từ Class1 sau khi form được đăng ký

        MessageBox.Show("In the callback method", Me.Text, MessageBoxButtons.OK,
    MessageBoxIcon.Information)

    End Sub
```

```
Private Sub cmdInterfaceCallback_Click(ByVal sender As System.Object, _
                        ByVal e As System.EventArgs) _
                        Handles cmdInterfaceCallback.Click
    ' Phương thức này gọi vào một phương thức của instance class1, nó gọi trở lại
    ' vào client thông qua ICallback
    Dim refClass1 As New Class1()

    ' Đăng ký client, truyền vào một tham chiếu đến chính nó.
    refClass1.RegisterInterFaceForCallback(Me)

    ' Gọi phương thức mà nó gọi trở lại client
    refClass1.UseInterfaceCallback()

    ' Hủy đăng ký client
    refClass1.UnRegisterInterfaceForCallback()

End Sub

Private Sub cmdDelegateCallback_Click(ByVal sender As System.Object, _
                        ByVal e As System.EventArgs) _
                        Handles cmdDelegateCallback.Click
    ' Phương thức này gọi một phương thức của instance Class1, nó gọi trở lại
    ' client thông qua một delegate

    Dim refClass1 As New Class1()

    ' Tạo mới một instance của delegate tương ứng cho phương thức callback
    Dim d As New Delegate1(AddressOf Me.CallbackMethod)
    ' Đăng ký client, truyền vào một tham chiếu đến instance của delegate
    refClass1.RegisterDelegateForCallback(d)
    ' Gọi phương thức mà nó gọi trở lại client
    refClass1.UseDelegateCallback()
    ' Hủy đăng ký client
    refClass1.UnRegisterDelegateForCallback()
End Sub

Private Sub cmdAsyncDelegateCallback_Click(ByVal sender As System.Object, _
                        ByVal e As System.EventArgs) _
                        Handles cmdAsyncDelegateCallback.Click
```

```
        ' Phương thức này gọi một phương thức của instacne Class1, nó gọi trở lại
        ' client một cách không đồng bộ thông qua một delegate.

        Dim refClass1 As New Class1()
        Dim d As New Delegate1(AddressOf Me.CallbackMethod)
        refClass1.RegisterDelegateForCallback(d)
        refClass1.UseAsyncDelegateCallback()
        refClass1.UnRegisterDelegateForCallback()
    End Sub

    Private Sub cmdBuiltInCallback_Click(ByVal sender As System.Object, _
                                         ByVal e As System.EventArgs) _
                                         Handles cmdBuiltinCallback.Click
        ' Các delegate có một kỹ thuật xây dựng sẵn để gọi trở lại client.

        Dim refClass1 As New Class1()
        Dim d As New Delegate1(AddressOf refClass1.UseBuiltInDelegateCallback)

        ' Tạo một delegate thứ hai để truyền vào đối tượng được gọi (called).
        ' Called sẽ sử dụng delegate thứ hai này để gọi trở lại client.
        Dim ac As New AsyncCallback(AddressOf Me.BuiltInCallback)
        d.BeginInvoke(ac, Nothing)
    End Sub

    Private Sub BuiltInCallback(ByVal ia As IAsyncResult)
        ' Phương thức này được gọi sử dụng AsyncCallback được truyền vào từ phương
        ' thức BeginInvoke trong thủ tục BuiltInCallback ở trên.
        MessageBox.Show("In the BuiltInCallback routine", Me.Text,
    MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub
End Class
```

38. Sử dụng các kiểu dữ liệu DateTime và TimeSpan

Biểu diễn cách sử dụng các class DateTime và TimeSpan.



Project: **Su dung DateTime va TimeSpan**

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
    Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        ' Gọi các thủ tục lấy các giá trị vào form
        LoadSharedMembers()
        LoadProperties()
        LoadConversionMethods()
        LoadCalculationMethods()
        LoadTimeSpanFields()
        LoadTSMethods()
    End Sub

    Private Sub btnCalcParse_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCalcParse.Click
        ' Hiển thị các thuộc tính của TimeSpan
        ' Phân tích một chuỗi chứa một giá trị TimeSpan
        Try
            Dim ts As TimeSpan
            ts = TimeSpan.Parse(txtParse.Text)
```

```
        ' Hiển thị các thuộc tính của instance TimeSpan vừa tạo
        DisplayTSProperties(ts)
    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
    End Try
End Sub

Private Sub btnRefreshCalculation_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnRefreshCalculation.Click
    LoadCalculationMethods()
End Sub

Private Sub btnRefreshConversion_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnRefreshConversion.Click
    LoadConversionMethods()
End Sub

Private Sub btnRefreshProperties_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnRefreshProperties.Click
    LoadProperties()
End Sub

Private Sub btnRefreshShared_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefreshShared.Click
    LoadSharedMembers()
End Sub

Private Sub btnRefreshTSMethods_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnRefreshTSMethods.Click
    LoadTSMethods()
End Sub

Private Sub btnRefreshTSProperties_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnRefreshTSProperties.Click
    Try
        ' Tạo một instance TimeSpan dựa trên các giá trị DateTime có trên form
        Dim ts As TimeSpan
        Dim dtStart As DateTime
        Dim dtEnd As DateTime
```

```
        ' Phân tách văn bản trong các TextBox
        dtStart = DateTime.Parse(txtStart.Text)
        dtEnd = DateTime.Parse(txtEnd.Text)
        ts = dtEnd.Subtract(dtStart).Duration

        ' Hiển thị các thuộc tính của instance TimeSpan vừa tạo
        DisplayTSProperties(ts)
    Catch Exp As Exception
        MessageBox.Show(Exp.Message, Me.Text)
    End Try
End Sub

Private Sub tabTimeSpan_SelectedIndexChanged(ByVal sender As Object, ByVal
e As System.EventArgs) Handles tabTimeSpan.SelectedIndexChanged
    ' Xóa tất cả dữ liệu trên trang chứa tabTimeSpan
    lblDays.Text = String.Empty
    lblHours.Text = String.Empty
    lblMilliseconds.Text = String.Empty
    lblMinutes.Text = String.Empty
    lblSeconds.Text = String.Empty
    lblTimeSpanTicks.Text = String.Empty
    lblTotalDays.Text = String.Empty
    lblTotalHours.Text = String.Empty
    lblTotalMilliseconds.Text = String.Empty
    lblTotalMinutes.Text = String.Empty
    lblTotalSeconds.Text = String.Empty
End Sub

Private Sub DisplayTSProperties(ByVal ts As TimeSpan)
    ' Sử dụng các thuộc tính của kiểu TimeSpan:
    ' TimeSpan.Days
    ' TimeSpan.Hours
    ' TimeSpan.Milliseconds
    ' TimeSpan.Minutes
    ' TimeSpan.Seconds
    ' TimeSpan.Ticks
    ' TimeSpan.TotalDays
    ' TimeSpan.TotalHours
    ' TimeSpan.TotalMilliseconds
```

```
        ' TimeSpan.TotalMinutes
        ' TimeSpan.TotalSeconds
      Try
        lblDays.Text = ts.Days.ToString
        lblHours.Text = ts.Hours.ToString
        lblMilliseconds.Text = ts.Milliseconds.ToString
        lblMinutes.Text = ts.Minutes.ToString
        lblSeconds.Text = ts.Seconds.ToString
        lblTimeSpanTicks.Text = ts.Ticks.ToString
        lblTotalDays.Text = ts.TotalDays.ToString
        lblTotalHours.Text = ts.TotalHours.ToString
        lblTotalMilliseconds.Text = ts.TotalMilliseconds.ToString
        lblTotalMinutes.Text = ts.TotalMinutes.ToString
        lblTotalSeconds.Text = ts.TotalSeconds.ToString
      Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
      End Try
    End Sub

    Private Sub LoadCalculationMethods()
        ' Sử dụng các phương thức của kiểu DateTime:
        ' DateTime.AddDays
        ' DateTime.AddHours
        ' DateTime.AddMilliseconds
        ' DateTime.AddMinutes
        ' DateTime.AddMonths
        ' DateTime.AddSeconds
        ' DateTime.AddTicks
        ' DateTime.AddYears
      Try
        Dim dt As DateTime = DateTime.Now
        lblNow3.Text = dt.ToString

        lblAddDays.Text = dt.AddDays(CDbl(txtDays.Text)).ToString
        lblAddHours.Text = dt.AddHours(CDbl(txtHours.Text)).ToString
        lblAddMilliseconds.Text =
dt.AddMilliseconds(CDbl(txtMilliseconds.Text)).ToString
        lblAddMinutes.Text = dt.AddMinutes(CDbl(txtMinutes.Text)).ToString
        lblAddMonths.Text = dt.AddMonths(CInt(txtMonths.Text)).ToString
```

```
        lblAddSeconds.Text = dt.AddSeconds(CDbl(txtSeconds.Text)).ToString
        lblAddTicks.Text = dt.AddTicks(CLng(txtTicks.Text)).ToString
        lblAddYears.Text = dt.AddYears(CInt(txtYears.Text)).ToString
    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
    End Try
End Sub

Private Sub LoadConversionMethods()
    ' Sử dụng các phương thức của kiểu DateTime:
    ' DateTime.ToFileTime
    ' DateTime.ToLocalTime
    ' DateTime.ToLongDateString
    ' DateTime.ToLongTimeString
    ' DateTime.ToOADate
    ' DateTime.ToShortDateString
    ' DateTime.ToShortTimeString
    ' DateTime.ToString
    ' DateTime.ToUniversalTime
    Try
        Dim dt As DateTime = DateTime.Now

        lblNow2.Text = dt.ToString
        lblToFileTime.Text = dt.ToFileTime.ToString
        lblToLocalTime.Text = dt.ToLocalTime.ToString
        lblToLongDateString.Text = dt.ToLongDateString
        lblToLongTimeString.Text = dt.ToLongTimeString
        lblToOADate.Text = dt.ToOADate.ToString
        lblToShortDateString.Text = dt.ToShortDateString
        lblToShortTimeString.Text = dt.ToShortTimeString
        lblToString.Text = dt.ToString
        lblToUniversalTime.Text = dt.ToUniversalTime.ToString

    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
    End Try

End Sub
```

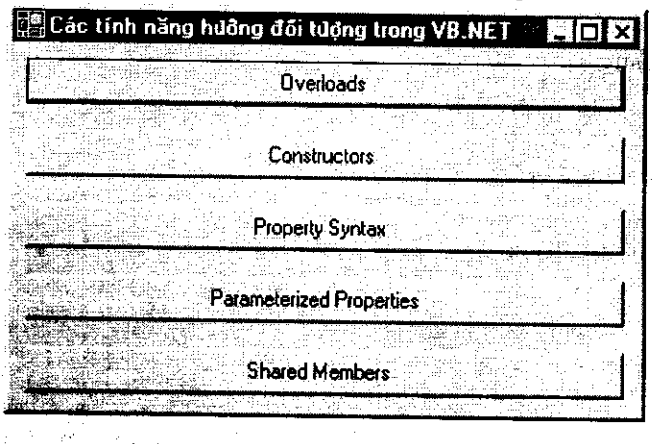
```
Private Sub LoadProperties()
    ' Sử dụng các thuộc tính của kiểu DateTime:
    ' DateTime.Now
    ' DateTime.Date
    ' DateTime.Day
    ' DateTime.DayOfYear
    ' DateTime.Hour
    ' DateTime.Millisecond
    ' DateTime.DayOfWeek
    ' DateTime.Minute
    ' DateTime.Month
    ' DateTime.Second
    ' DateTime.Ticks
    ' DateTime.TimeOfDay
    ' DateTime.Year
  Try
    Dim dt As DateTime = DateTime.Now

    lblNow1.Text = dt.ToString
    lblDate.Text = dt.Date.ToString
    lblDay.Text = dt.Day.ToString
    lblDayOfYear.Text = dt.DayOfYear.ToString
    lblHour.Text = dt.Hour.ToString
    lblMillisecond.Text = dt.Millisecond.ToString
    lblDayOfWeek.Text = dt.DayOfWeek.ToString
    lblMinute.Text = dt.Minute.ToString
    lblMonth.Text = dt.Month.ToString
    lblSecond.Text = dt.Second.ToString
    lblTicks.Text = dt.Ticks.ToString
    lblTimeOfDay.Text = dt.TimeOfDay.ToString
    lblYear.Text = dt.Year.ToString

  Catch exp As Exception
    MessageBox.Show(exp.Message, Me.Text)
  End Try

End Sub
```

```
Private Sub LoadSharedMembers()
    ' Sử dụng các thành phần Shared của class DateTime:
    ' Các thuộc tính:
    '  DateTime.Now,'  DateTime.UtcNow,'  DateTime.MinValue
    '  DateTime.MaxValue
    ' Các phương thức:
    '  DateTime.FromOADate,  DateTime.IsLeapYear
    '  DateTime.DaysInMonth
  Try
    lblNow.Text = DateTime.Now.ToString
    lblToday.Text = DateTime.Today.ToString
    lblUtcNow.Text = DateTime.UtcNow.ToString
    lblMinValue.Text = DateTime.MinValue.ToString
    lblMaxValue.Text = DateTime.MaxValue.ToString

    lblFromOADate.Text = DateTime.FromOADate(_
      CDbl(txtFromOADate.Text)).ToString
    lblIsLeapYear.Text = DateTime.IsLeapYear(CInt(txtIsLeapYear.Text)).ToString
    lblDaysInMonth.Text = DateTime.DaysInMonth( _
     CInt(txtYear.Text), CInt(txtMonth.Text)).ToString
  Catch exp As Exception
    MessageBox.Show(exp.Message, Me.Text)
  End Try
End Sub

Private Sub LoadTimeSpanFields()
    ' Sử dụng các thành phần Shared của class TimeSpan:
    '  TimeSpan.MaxValue
    '  TimeSpan.Minvalue
    '  TimeSpan.TicksPerDay
    '  TimeSpan.TicksPerHour
    '  TimeSpan.TicksPerMillisecond
    '  TimeSpan.TicksPerMinute
    '  TimeSpan.TicksPerSecond
    '  TimeSpan.Zero
  Try
    lblMaxValueTS.Text = TimeSpan.MaxValue.ToString
    lblMinValueTS.Text = TimeSpan.MinValue.ToString
    lblTicksPerDay.Text = TimeSpan.TicksPerDay.ToString
```

```
        lblTicksPerHour.Text = TimeSpan.TicksPerHour.ToString
        lblTicksPerMillisecond.Text = TimeSpan.TicksPerMillisecond.ToString
        lblTicksPerMinute.Text = TimeSpan.TicksPerMinute.ToString
        lblTicksPerSecond.Text = TimeSpan.TicksPerSecond.ToString
        lblZero.Text = TimeSpan.Zero.ToString
      Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
      End Try
    End Sub

    Private Sub LoadTSMethods()
      ' Sử dụng các phương thức Shared của class TimeSpan:
      ' TimeSpan.FromDays
      ' TimeSpan.FromHours
      ' TimeSpan.FromMilliseconds
      ' TimeSpan.FromMinutes
      ' TimeSpan.FromSeconds
      ' TimeSpan.FromTicks
      Try
        lblFromDays.Text = TimeSpan.FromDays(CDbl(txtFromDays.Text)).ToString
        lblFromHours.Text = _
            TimeSpan.FromHours(CDbl(txtFromHours.Text)).ToString
        lblFromMilliseconds.Text = _
            TimeSpan.FromMilliseconds(CDbl(txtFromMilliseconds.Text)).ToString
        lblFromMinutes.Text = _
            TimeSpan.FromMinutes(CDbl(txtFromMinutes.Text)).ToString
        lblFromSeconds.Text = _
            TimeSpan.FromSeconds(CDbl(txtFromSeconds.Text)).ToString
        lblFromTicks.Text = _
            TimeSpan.FromTicks(CLng(txtFromTicks.Text)).ToString
      Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text)
      End Try
    End Sub

End Class
```

39. Các tính năng hướng đối tượng trong Visual Basic .NET

Chương trình này biểu diễn một số tính năng hướng đối tượng (object oriented - OO) mới trong VB.NET. Nhấn F5 để chạy với chế độ Debug tại các breakpoint. Nhấn Ctrl+F5 để chạy chương trình và bỏ qua các breakpoint



Các chức năng chính:

Các nút trên form chính tương ứng với một tính năng OO trong VB.NET:

- Overloads
- Constructors
- Cú pháp Property
- Các thuộc tính có tham số
- Các thành phần chia sẻ (Shared)

Mỗi tính năng có một class và một file source riêng. Tuy nhiên, có thể đặt năm class này vào một source file.

Project: **Huong Doi Tuong trong VB.NET**

Customer.vb

```
Option Strict On
Public Class Customer

    Private custAccount As String
    Private custFirstName As String
    Private custLastName As String
```

```
    Public Property AccountNumber() As String
      Get
        Return custAccount
      End Get
      Set(ByVal Value As String)
        custAccount = Value
      End Set
    End Property

    ' Các thuộc tính chuẩn
    Public Property FirstName() As String
      Get
        Return custFirstName
      End Get
      Set(ByVal Value As String)
        custFirstName = Value
      End Set
    End Property

    Public Property LastName() As String
      Get
        Return custLastName
      End Get
      Set(ByVal Value As String)
        custLastName = Value
      End Set
    End Property
End Class
```

CustomerPropertySyntax.vb

```
Public Class CustomerPropertySyntax
    Private custAccount As String
    Private custFirstName As String
    Private custLastName As String

    Sub New(ByVal AccountNumber As String)
        ' Đây là bộ khởi tạo (Constructor) cho class này.
        ' Thông thường, AccountNumber lấy từ database, hoặc từ một tập hợp.
        ' Ta sử dụng một số trong ví dụ này
```

```
        ' Bởi vì thuộc tính AccountNumber là thuộc tính chỉ đọc, do đó
        ' biến custAccount phải khai báo với private
        custAccount = "1101"
        Me.FirstName = "Carmen"
        Me.LastName = "Smith"
    End Sub

    ' Khai báo thuộc tính AccountNumber
    Public ReadOnly Property AccountNumber() As String

        ' Hầu hết các thuộc tính có thể đọc/ghi, có nghĩa là có thể lấy giá trị và
        ' thay đổi giá trị bằng code. Khi thuộc tính có thể đọc/ghi thì nó có hai
        ' phần Get và Set bên trong code của thuộc tính.

        ' Bởi vì thuộc tính này khai báo ReadOnly, chỉ cần có phần Get.
        Get
            Return custAccount
        End Get
        ' Ngoài ra có thể khai báo thuộc tính với WriteOnly, thì chỉ cần phần Set
    End Property

    Public Property FirstName() As String
        Get
            Return custFirstName
        End Get
        Set(ByVal Value As String)
            custFirstName = Value
        End Set
    End Property

    Public Property LastName() As String
        Get
            Return custLastName
        End Get
        Set(ByVal Value As String)
            custLastName = Value
        End Set
    End Property
End Class
```

CustomerWithConstructor.vb

```
Option Strict On
Public Class CustomerWithConstructor

    Private custAccount As String
    Private custFirstName As String
    Private custLastName As String

    Sub New(ByVal AccountNumber As String, ByVal FirstName As String, ByVal
    LastName As String)
        ' Đây là Constructor của class này
        Me.AccountNumber = AccountNumber
        Me.FirstName = FirstName
        Me.LastName = LastName
    End Sub

    Public Property AccountNumber() As String
        Get
            Return custAccount
        End Get
        Set(ByVal Value As String)
            custAccount = Value
        End Set
    End Property

    Public Property FirstName() As String
        Get
            Return custFirstName
        End Get
        Set(ByVal Value As String)
            custFirstName = Value
        End Set
    End Property

    Public Property LastName() As String
        Get
            Return custLastName
        End Get
```

```
    Set(ByVal Value As String)
      custLastName = Value
    End Set
  End Property
End Class
```

CustomerWithParameterizedProperty.vb

```
Option Strict On
Public Class CustomerWithParameterizedProperty

  Private custAccount As String
  Private custFirstName As String
  Private custLastName As String
  Private custDefaultQuantity As Integer = 12

  ' Để tạo một thuộc tính có tham số, chỉ đơn giản thêm tham số vào câu lệnh
  ' khai báo thuộc tính, trong trường hợp này là "Multiplier"
  Public Property DefaultQuantity(ByVal Multiplier As Integer) As Integer
    Get
      ' Thông thường thủ tục Get của thuộc tính không sử dụng tham số truyền vào,
      ' nhưng ở đây cho thấy cách làm việc của thuộc tính có tham số. Tham số của
      ' thuộc tính được xem như là tham số bình thường được truyền vào.
      Return custDefaultQuantity * Multiplier
    End Get

    Set(ByVal Value As Integer)
      custDefaultQuantity = Value
    End Set
  End Property

  Public Property AccountNumber() As String
    Get
      Return custAccount
    End Get
    Set(ByVal Value As String)
      custAccount = Value
    End Set
  End Property
```

```
    Public Property FirstName() As String
        Get
            Return custFirstName
        End Get

        Set(ByVal Value As String)
            custFirstName = Value
        End Set
    End Property

    Public Property LastName() As String
        Get
            Return custLastName
        End Get
        Set(ByVal Value As String)
            custLastName = Value
        End Set
    End Property
End Class
```

CustomerWithSharedMembers.vb

```
Option Strict On
Public Class CustomerWithSharedMembers
    Private custAccount As String
    Private custFirstName As String
    Private custLastName As String

    ' Biến bên dưới (CompanyName) được khai báo là Shared. Tất cả các instance
    ' của class này sử dụng cùng một giá trị của biến này. Nếu nó bị thay đổi trong
    ' một instance của class này, nó sẽ thay đổi trong toàn bộ các instance đang tồn tại.
    Public Shared CompanyName As String

    Public Shared Function LastOrderDate() As Date
        ' Đây là một hàm chia sẻ (Shared), xem cách gọi trong frmMain để biết thêm
        Dim orderDate As Date
        orderDate = Now
        Return orderDate
    End Function
```

```
    Public Property AccountNumber() As String
        Get
            Return custAccount
        End Get
        Set(ByVal Value As String)
            custAccount = Value
        End Set
    End Property

    Public Property FirstName() As String
        Get
            Return custFirstName
        End Get
        Set(ByVal Value As String)
            custFirstName = Value
        End Set
    End Property

    Public Property LastName() As String
        Get
            Return custLastName
        End Get
        Set(ByVal Value As String)
            custLastName = Value
        End Set
    End Property
End Class
```

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

    Private Sub cmdConstructors_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles cmdConstructors.Click
        ' Một Constructor trong VB.NET chỉ đơn giản là code được thực  hiện khi tạo
        ' một đối tượng class mới (instanced). Bộ khởi tạo (constructor) có thể có
        ' hoặc không đòi hỏi các tham số truyền vào.
```

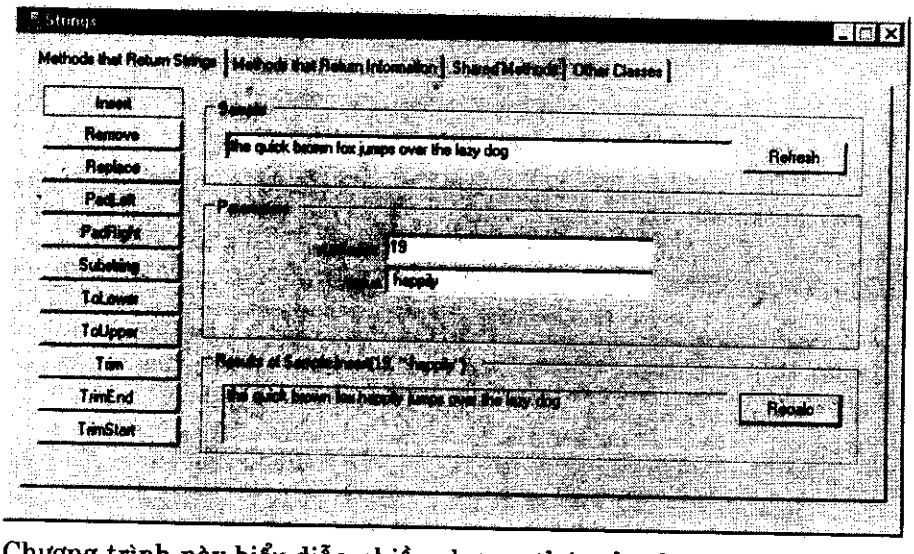
```
    ' Đây là cú pháp tạo một đối tượng mới với constructor trên cùng một dòng.
    Dim cust As New CustomerWithConstructor("1101", "Carmen", "Smith")

    ' Ngoài ra, có thể tạo mới đối tượng sau khi khai báo biến
    Dim cust2 As CustomerWithConstructor
    cust2 = New CustomerWithConstructor("1101", "Carmen", "Smith")

    ' Người lập trình có thể overload (chồng lên) thủ tục Sub New trong một class.
    ' Bằng cách này, có thể tạo các phiên bản constructor khác nhau. Ví dụ, nếu
    ' muốn tạo một constructor chỉ cho phép gọi từ bên trong class, khai báo trong
    ' class như sau:
    '
    ' Private Overloads Sub New().
    '
    ' End Sub
End Sub

Private Sub cmdOverloads_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdOverloads.Click

    Dim myCust1 As Customer
    Dim myCust2 As Customer
    ' Mục đích của từ khóa "Overloads" trong VB.NET là cho phép tạo nhiều thủ tục
    ' có cùng tên nhưng thực hiện code khác nhau
    '
    ' Giả sử ta muốn lấy thông tin Customer theo ID hoặc theo Name. Tạo hai hàm
    ' có cùng tên GetCustomer với từ khóa Overloads nhưng với các tham số khác
    ' nhau (xem code thực sự của hai hàm bên dưới):

    myCust1 = GetCustomer("Smith")
    myCust2 = GetCustomer(1101)

End Sub

Private Sub cmdParamProperties_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdParamProperties.Click

    Dim cust As New CustomerWithParameterizedProperty()
```

```
    ' Để tạo một thuộc tính có tham số, thêm tham số vào câu lệnh Property.
    '
    ' Với các thuộc tính có tham số, ta phải chỉ định các tham số cho dù thủ tục thuộc
    ' tính không sử dụng các tham số này. Dòng lệnh sau gán vào DefaultQuantity
    ' giá trị 100:

    cust.DefaultQuantity(0) = 100

    MessageBox.Show("The default quantity for this customer is " &
cust.DefaultQuantity(3), Me.Text, MessageBoxButtons.OK,
MessageBoxIcon.Information)

End Sub

Private Sub cmdPropertySyntax_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdPropertySyntax.Click

    ' Đoạn code sau tạo một instance của class CustomerPropertySyntax và lấy
    ' giá trị từ thuộc tính ReadOnly, AccountNumber.

    Dim myAccount As String
    Dim cust As New CustomerPropertySyntax("1101")

    myAccount = cust.AccountNumber
End Sub

Private Sub cmdSharedMembers_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdSharedMembers.Click
    ' Có hai kiểu thành phần chia sẻ trong một class: các biến chia sẻ, và các kiểu
    ' thủ tục chia sẻ.
    '
    ' Bên dưới ta tạo hai instance của class customer và thay đổi giá trị của biến chia
    ' sẻ "CompanyName". Khi thực hiện đoạn code này ta sẽ thấy một thông báo là
    ' Tailspin Toys. Điều này có nghĩa là thay đổi giá trị trong bất kỳ instance nào
    ' của class sẽ thay đổi giá trị trong tất cả các instance.

    Dim cust1 As New CustomerWithSharedMembers()
    Dim cust2 As New CustomerWithSharedMembers()
```

```
        ' Có thể sử dụng cú pháp chuẩn cho đối tượng như sau:
        cust1.CompanyName = "Wingtip Toys"
        cust2.CompanyName = "Tailspin Toys"

        ' Tuy nhiên, cú pháp nên dùng cho các thành phần chia sẽ là
        ' ClassName.SharedMemberName, ví dụ:
        ' CustomerWithSharedMembers.CompanyName = "Tailspin Toys".
        MessageBox.Show("The company name is " & cust1.CompanyName, Me.Text)

        ' Các thủ tục chia sẻ của class có thể gọi thông qua một instance của class hoặc
        ' không cần một instance với cú pháp:
        ' Classname.ProcedureName, và cú pháp nên dùng cho các thủ tục chia sẻ là:
        '
        ' ClassName.SharedProcedureName syntax.

        ' Gọi phương thức chia sẻ thông qua instance:
        MessageBox.Show("The last order was placed on " & _
        cust1.LastOrderDate, Me.Text, MessageBoxButtons.OK,
    MessageBoxIcon.Information)

        ' Gọi phương thức chia sẻ thông qua tên class
        MessageBox.Show("The last order was placed on " & _
        CustomerWithSharedMembers.LastOrderDate, Me.Text,
    MessageBoxButtons.OK, MessageBoxIcon.Information)
    End Sub

    Private Overloads Function GetCustomer(ByVal CustID As Integer) As Customer

        ' Hàm này được gọi nếu truyền vào một số nguyên
        Dim cust As New Customer()

        ' ta sử dụng dữ liệu giả
        cust.AccountNumber = "1101"
        cust.FirstName = "Carmen"
        cust.LastName = "Smith"

        Return cust
    End Function
```

```
    Private Overloads Function GetCustomer(ByVal CustLastName As String) As
    Customer

        ' Hàm này được gọi nếu truyền vào một chuỗi
        Dim cust As New Customer()

        ' Sử dụng dữ liệu giả
        cust.AccountNumber = "1101"
        cust.FirstName = "Carmen"
        cust.LastName = "Smith"

        Return cust
    End Function

End Class
```

40. Chuỗi – String



Chương trình này biểu diễn nhiều phương thức của class **String**. Các phương thức được chia ra làm ba nhóm:

- Các phương thức trả về các chuỗi (như Insert, Remove...)
- Các phương thức trả về thông tin (như IndexOf)
- Các phương thức chia sẻ (Shared) (như String.Format)

Ngoài ra còn có các class hữu dụng khác là **StringBuilder** và **StringWriter**.

Project: **Chuoi - String**

frmMain.vb

```
Option Strict On
Imports System.Text
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Const DEFAULT_TEXT As String = "the quick brown fox jumps over the
    lazy dog"
    Private Const QUOTE As String = """
    Private mstrSample As String = DEFAULT_TEXT
    Private rbtnShared As RadioButton
    Private rbtnStringReturn As RadioButton
    Private rbtnStringInfo As RadioButton

    Private Enum SelectedMethod
        Compare
        CompareOrdinal
        Concat
        EndsWith
        Format
        IndexOf
        IndexOfAny
        Insert
        Join
        LastIndexOf
        LastIndexOfAny
        PadLeft
        PadRight
        Remove
        Replace
        Split
        StartsWith
        Substring
```

```
    ToLower
    ToUpper
    Trim
    TrimEnd
    TrimStart
End Enum

Private sm As SelectedMethod

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
    txtResults.Clear()
End Sub

Private Sub btnRecalc_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRecalc.Click
    Recalc()
End Sub

Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
    RefreshText()
End Sub

Private Sub btnStringBuilder_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStringBuilder.Click
    ' demo class StringBuilder

    If chkDebug.Checked Then
        ' Vào chế độ debug
        System.Diagnostics.Debugger.Break()
    End If

    ' StringBuilder nằm trong namespace System.Text, đã imports ở trên
    Dim sb As New StringBuilder("The quick brown fox jumps over the lazy dog")

    sb.Insert(19, " happily")
    sb.Remove(10, 6)
    sb.Replace("jumps", "leaps")
    sb.AppendFormat(" {0} times in {1} minutes", 17, 2)
```

```
        ' Giả sử muốn thêm một dấu phẩy vào sau từ "dog". Cần xác định vị trí từ này
        ' bằng cách sử dụng IndexOf, nhưng StringBuilder không cung cấp phương thức
        ' này. Do đó, phải sử dụng phương thức ToString để nhận một chuỗi, và làm việc
        ' với đối tượng này.
        Dim intPos As Integer
        intPos = sb.ToString.IndexOf("dog")
        If intPos > 0 Then
            ' Chèn dấu phẩy vào vị trí tìm thấy cộng với độ dài của từ "dog"
            sb.Insert(intPos + "dog".Length, ",")
        End If

        ' Code tương tự sử dụng trực tiếp đối tượng String
        Dim str As String = "The quick brown fox jumps over the lazy dog"
        str = str.Insert(19, " happily")
        str = str.Remove(10, 6)
        str = str.Replace("jumps", "leaps")
        str &= String.Format("{0} times in {1} minutes", 17, 2)
        intPos = str.IndexOf("dog")
        If intPos > 0 Then
            str = str.Insert(intPos + "dog".Length, ",")
        End If

        txtResults.AppendText("StringBuilder output: " & sb.ToString &
    Environment.NewLine)
        txtResults.AppendText("String output: " & str & Environment.NewLine)
    End Sub

    Private Sub btnStringWriter_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnStringWriter.Click
        ' Demo class StringWriter
        ' Ghi vào StringWriter giống như là ghi vào một file –
        ' nhưng ghi vào bộ nhớ thay vì ghi vào file
        ' StringWriter nằm trong namespace System.IO

        Dim sw As New StringWriter()

        If chkDebug.Checked Then
            System.Diagnostics.Debugger.Break()
        End If
```

```
' Giả sử nhận được một cấu trúc dữ liệu, một mảng các chuỗi chứa thông tin địa chỉ.
    Dim addressInfo() As String = {"John Smith", "123 Main Street", "Centerville",
"WA", "98111"}

' Tạo một địa chỉ có định dạng
    ' Cho các dòng tên và địa chỉ vào StringWriter
    sw.WriteLine(addressInfo(0))
    sw.WriteLine(addressInfo(1))

    ' Có thể sử dụng phương thức String.Format để tạo dòng địa chỉ, nhưng ở đây
    ' chỉ cách sử dụng cả hai phương thức WriteLine và Write.
    sw.Write(addressInfo(2))
    sw.Write(", ")
    sw.Write(addressInfo(3))
    sw.Write(" ")
    sw.WriteLine(addressInfo(4))

    ' hoặc:
    ' sw.WriteLine(String.Format("{0}, {1} {2}", addressInfo(2), _
    '                                    addressInfo(3), addressInfo(4)))

    ' Sử dụng class String, code khó chịu hơn
    Dim str As String
    str = addressInfo(0) & Environment.NewLine
    str &= addressInfo(1) & Environment.NewLine
    str &= addressInfo(2) & ", "
    str &= addressInfo(3) & " " & addressInfo(4)
    str &= Environment.NewLine

    ' hoặc:
    ' str &= String.Format("{0}, {1} {2}{3}", addressInfo(2), addressInfo(3), _
    '                                    addressInfo(4), Environment.NewLine)

    ' Hiển thị kết quả
    txtResults.AppendText("StringWriter output: " & _
        Environment.NewLine & sw.ToString)
    txtResults.AppendText("String output: " & _
        Environment.NewLine & str)
End Sub
```

```
Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    ' Khởi tạo các giá trị
    ' Trang được chọn
    tabStringDemo.SelectedIndex = 0

    ' Các nút được chọn trên mỗi trang
    ' Chọn các nút đầu tiên trên mỗi trang
    rbtnStringInfo = optIndexOf
    rbtnStringReturn = optInsert
    rbtnShared = optCompare
    ' Hiển thị văn bản trên trang đầu tiên
    DisplayText(optInsert)
End Sub

Private Sub tabStringDemo_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
tabStringDemo.SelectedIndexChanged
    Dim pagSelected As TabPage = tabStringDemo.SelectedTab
    ' Code này chạy trước sự kiện Load của form.
    ' Không thực hiện nếu sự kiện Load của form chưa thực hiện.
    If Not rbtnStringReturn Is Nothing Then
        ' Di chuyển panel chứa thông tin đến trang hiện hành.
        ' Nhưng không làm đối với trang "Other"
        If Not pagSelected Is pagOther Then
            pnlDemo.Parent = pagSelected
        End If
        ' Dựa trên trang được chọn, hiển thị kết quả
        If pagSelected Is pagReturnStrings Then
            grpSample.Visible = True
            DisplayText(rbtnStringReturn)
        ElseIf pagSelected Is pagInfo Then
            grpSample.Visible = True
            DisplayText(rbtnStringInfo)
        ElseIf pagSelected Is pagShared Then
            grpSample.Visible = False
            DisplayText(rbtnShared)
        End If
    End If
End Sub
```

```
Private Function AddQuotes(ByVal strValue As String) As String
    ' Thêm các dấu ngoặc kép để bọc văn bản.
    ' Thay thế một dấu ngoặc (")kép thành hai dấu ngoặc kép ("")
    Return QUOTE & Replace(strValue, QUOTE, QUOTE & QUOTE) & QUOTE
End Function

Private Sub DisplayText(ByVal rbtn As RadioButton)
    ' Thực hiện công việc dựa trên radio button được chọn
    If rbtn Is optCompare Then
        rbtnShared = rbtn
        sm = SelectedMethod.Compare
        HandleParameters("strA", "This is a test", "strB", "this is a test", _
                                                    "ignoreCase", "True")

    ElseIf rbtn Is optCompareOrdinal Then
        rbtnShared = rbtn
        sm = SelectedMethod.CompareOrdinal
        HandleParameters("strA", "This is a test", "strB", "this is a test")

    ElseIf rbtn Is optConcat Then
        rbtnShared = rbtn
        sm = SelectedMethod.Concat
        HandleParameters("strA", "This is a test", "strB", " of how this works", _
                                "strC", " when you concatenate")

    ElseIf rbtn Is optEndsWith Then
        rbtnStringInfo = rbtn
        sm = SelectedMethod.EndsWith
        mstrSample = DEFAULT_TEXT
        HandleParameters("value", "dog")

    ElseIf rbtn Is optFormat Then
        rbtnShared = rbtn
        sm = SelectedMethod.Format
        HandleParameters("Format", "Your {0:N0} items total {1:C}.", _
                                "Param1", "12", "Param2", "12.35")

    ElseIf rbtn Is optIndexOf Then
        rbtnStringInfo = rbtn
        sm = SelectedMethod.IndexOf
```

```
        mstrSample = DEFAULT_TEXT
        HandleParameters("value", "o", "startIndex", "", "count", "")

    ElseIf rbtn Is optIndexOfAny Then
        rbtnStringInfo = rbtn
        sm = SelectedMethod.IndexOfAny
        mstrSample = DEFAULT_TEXT
        HandleParameters("anyOf()", "abc", "startIndex", "", "count", "")

    ElseIf rbtn Is optInsert Then
        rbtnStringReturn = rbtn
        sm = SelectedMethod.Insert
        mstrSample = DEFAULT_TEXT
        HandleParameters("startIndex", "19", "value", " happily")

    ElseIf rbtn Is optJoin Then
        rbtnShared = rbtn
        sm = SelectedMethod.Join
        HandleParameters("separator", "-", "values()", "item1,item2,item3,item4")

    ElseIf rbtn Is optLastIndexOf Then
        rbtnStringInfo = rbtn
        sm = SelectedMethod.LastIndexOf
        mstrSample = DEFAULT_TEXT
        HandleParameters("value", "o", "startIndex", "", "count", "")

    ElseIf rbtn Is optLastIndexOfAny Then
        rbtnStringInfo = rbtn
        sm = SelectedMethod.LastIndexOfAny
        mstrSample = DEFAULT_TEXT
        HandleParameters("anyOf()", "abc", "startIndex", "", "count", "")

    ElseIf rbtn Is optPadLeft Then
        rbtnStringReturn = rbtn
        sm = SelectedMethod.PadLeft
        mstrSample = "123.45"
        HandleParameters("totalWidth", "10", "paddingChar", "$")

    ElseIf rbtn Is optPadRight Then
        rbtnStringReturn = rbtn
```

```
    sm = SelectedMethod.PadRight
    mstrSample = "Pad"
    HandleParameters("totalWidth", "10", "paddingChar", "_")

  ElseIf rbtn Is optRemove Then
    rbtnStringReturn = rbtn
    sm = SelectedMethod.Remove
    mstrSample = DEFAULT_TEXT
    HandleParameters("startIndex", "10", "count", "6")

  ElseIf rbtn Is optReplace Then
    rbtnStringReturn = rbtn
    sm = SelectedMethod.Replace
    mstrSample = DEFAULT_TEXT
    HandleParameters("oldValue", "jumps", "newValue", "leaps")

  ElseIf rbtn Is optSplit Then
    rbtnStringInfo = rbtn
    sm = SelectedMethod.Split
    mstrSample = DEFAULT_TEXT
    HandleParameters("separator()", "abc", "count", "")

  ElseIf rbtn Is optStartsWith Then
    rbtnStringInfo = rbtn
    sm = SelectedMethod.StartsWith
    mstrSample = DEFAULT_TEXT
    HandleParameters("value", "The qui")

  ElseIf rbtn Is optSubstring Then
    rbtnStringReturn = rbtn
    sm = SelectedMethod.Substring
    mstrSample = DEFAULT_TEXT
    HandleParameters("startIndex", "12", "length", "5")

  ElseIf rbtn Is optToLower Then
    rbtnStringReturn = rbtn
    sm = SelectedMethod.ToLower
    mstrSample = "This Sample has SOME mixed-CASE Text!"
    HandleParameters()
```

```
    ElseIf rbtn Is optToUpper Then
        rbtnStringReturn = rbtn
        sm = SelectedMethod.ToUpper
        mstrSample = "This Sample has SOME mixed-CASE Text!"
        HandleParameters()
    ElseIf rbtn Is optTrim Then
        rbtnStringReturn = rbtn
        sm = SelectedMethod.Trim
        mstrSample = DEFAULT_TEXT
        HandleParameters("trimChars()", "the dog")
    ElseIf rbtn Is optTrimEnd Then
        rbtnStringReturn = rbtn
        sm = SelectedMethod.TrimEnd
        mstrSample = DEFAULT_TEXT
        HandleParameters("trimChars()", "the dog")
    ElseIf rbtn Is optTrimStart Then
        rbtnStringReturn = rbtn
        sm = SelectedMethod.TrimStart
        mstrSample = DEFAULT_TEXT
        HandleParameters("trimChars()", "the dog")
    End If
    ' Hiển thị văn bản ví dụ,và gọi Recalc để hiển thị kết quả
    RefreshText()
    Recalc()
End Sub

Private Sub HandleCheckedChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles optInsert.CheckedChanged,
optPadLeft.CheckedChanged, optPadRight.CheckedChanged,
optRemove.CheckedChanged, optReplace.CheckedChanged,
optSubstring.CheckedChanged, optToLower.CheckedChanged,
optToUpper.CheckedChanged, optTrim.CheckedChanged,
optTrimEnd.CheckedChanged, optTrimStart.CheckedChanged,
optEndsWith.CheckedChanged, optIndexOf.CheckedChanged,
optIndexOfAny.CheckedChanged, optLastIndexOfAny.CheckedChanged,
optStartsWith.CheckedChanged, optLastIndexOf.CheckedChanged,
optCompare.CheckedChanged, optCompareOrdinal.CheckedChanged,
optConcat.CheckedChanged, optFormat.CheckedChanged,
optJoin.CheckedChanged
    DisplayText(CType(sender, RadioButton))
End Sub
```

```
Private Sub HandleParameters()
    ' Hiện thị không tham số
    HandleParameterDisplay(0)
End Sub

Private Sub HandleParameters(ByVal Param1 As String, ByVal Param1Value As
String)
    ' Hiển thị một tham số
    lblPrm1.Text = Param1
    txtPrm1.Text = Param1Value
    HandleParameterDisplay(1)
End Sub

Private Sub HandleParameters(ByVal Param1 As String, ByVal Param1Value As
String, ByVal Param2 As String, ByVal Param2Value As String)
    ' Hiển thị hai tham số
    lblPrm1.Text = Param1
    txtPrm1.Text = Param1Value
    lblPrm2.Text = Param2
    txtPrm2.Text = Param2Value
    HandleParameterDisplay(2)
End Sub

Private Sub HandleParameters(ByVal Param1 As String, ByVal Param1Value As
String, ByVal Param2 As String, ByVal Param2Value As String, ByVal Param3 As
String, ByVal Param3Value As String)
    ' Hiển thị ba tham số
    lblPrm1.Text = Param1
    txtPrm1.Text = Param1Value
    lblPrm2.Text = Param2
    txtPrm2.Text = Param2Value
    lblPrm3.Text = Param3
    txtPrm3.Text = Param3Value
    HandleParameterDisplay(3)
End Sub

Private Sub HandleParameterDisplay(ByVal intCount As Integer)
    ' Hiển thị hoặc ẩn các Label tham số và TextBox dựa trên số tham số
    lblPrm1.Visible = (intCount > 0)
```

```
    txtPrm1.Visible = (intCount > 0)
    lblPrm2.Visible = (intCount > 1)
    txtPrm2.Visible = (intCount > 1)
    lblPrm3.Visible = (intCount > 2)
    txtPrm3.Visible = (intCount > 2)
  End Sub

  Private Sub HandleResultsLabel(ByVal MethodName As String)
    SetResultsLabel("Sample", MethodName, Nothing)
  End Sub

  Private Sub HandleResultsLabel(ByVal MethodName As String, ByVal Param1 As
String)
    SetResultsLabel("Sample", MethodName, Param1)
  End Sub

  Private Sub HandleResultsLabel(ByVal MethodName As String, ByVal Param1 As
String, ByVal Param2 As String)
    SetResultsLabel("Sample", MethodName, Param1 & ", " & Param2)
  End Sub

  Private Sub HandleResultsLabel(ByVal MethodName As String, ByVal Param1 As
String, ByVal Param2 As String, ByVal Param3 As String)
    SetResultsLabel("Sample", MethodName, Param1 & ", " & Param2 & ", " &
Param3)
  End Sub

  Private Sub HandleSharedResultsLabel(ByVal MethodName As String)
    SetResultsLabel("String", MethodName, Nothing)
  End Sub

  Private Sub HandleSharedResultsLabel(ByVal MethodName As String, ByVal
Param1 As String)
    SetResultsLabel("String", MethodName, Param1)
  End Sub

  Private Sub HandleSharedResultsLabel(ByVal MethodName As String, ByVal
Param1 As String, ByVal Param2 As String)
    SetResultsLabel("String", MethodName, Param1 & ", " & Param2)
  End Sub
```

```
Private Sub HandleSharedResultsLabel(ByVal MethodName As String, ByVal
Param1 As String, ByVal Param2 As String, ByVal Param3 As String)
    SetResultsLabel("String", MethodName, Param1 & ", " & Param2 & ", " &
Param3)
End Sub

Private Sub Recalc()
    Dim strSample As String = txtSample.Text
    Dim strParam1 As String = txtPrm1.Text.TrimEnd
    Dim strParam2 As String = txtPrm2.Text.TrimEnd
    Dim strParam3 As String = txtPrm3.Text.TrimEnd

    Try
      Select Case sm
        Case SelectedMethod.Compare
          If strParam3 <> String.Empty Then
            HandleSharedResultsLabel("Compare", AddQuotes(strParam1), _
                                     AddQuotes(strParam2), strParam3)
            lblResults.Text = String.Compare(strParam1, strParam2, _
                                     CBool(strParam3)).ToString
          Else

            HandleSharedResultsLabel("Compare", AddQuotes(strParam1), _
                                     AddQuotes(strParam2))
            lblResults.Text = String.Compare(strParam1, strParam2).ToString
          End If

        Case SelectedMethod.CompareOrdinal
          HandleSharedResultsLabel("CompareOrdinal", AddQuotes(strParam1))
          lblResults.Text = String.CompareOrdinal(strParam1, strParam2).ToString

        Case SelectedMethod.Concat
          HandleSharedResultsLabel("Concat", AddQuotes(strParam1), _
                                   AddQuotes(strParam2), AddQuotes(strParam3))
          lblResults.Text = String.Concat(strParam1, strParam2, strParam3)

        Case SelectedMethod.EndsWith
          HandleResultsLabel("EndsWith", AddQuotes(strParam1))
            ' EndsWith trả về một giá trị boolean – chuyển sang một chuỗi
          lblResults.Text = strSample.EndsWith(strParam1).ToString
```

```
Case SelectedMethod.Format
  HandleSharedResultsLabel("Format", AddQuotes(strParam1), _
                                        strParam2, strParam3)
  lblResults.Text = String.Format(strParam1, CInt(strParam2), _
                                        CDbl(strParam3))
Case SelectedMethod.IndexOf
  If strParam1 <> String.Empty Then
    If strParam2 <> String.Empty And strParam3 <> String.Empty Then
      HandleResultsLabel("IndexOf", AddQuotes(strParam1), _
                                        strParam2, strParam3)
      lblResults.Text = strSample.IndexOf(strParam1, CInt(strParam2), _
                                        CInt(strParam3)).ToString
    ElseIf strParam2 <> String.Empty Then
      HandleResultsLabel("IndexOf", AddQuotes(strParam1), strParam2)
      lblResults.Text = strSample.IndexOf(strParam1, _
                                        CInt(strParam2)).ToString
    Else
      HandleResultsLabel("IndexOf", AddQuotes(strParam1))
      lblResults.Text = strSample.IndexOf(strParam1).ToString
    End If
  End If

Case SelectedMethod.IndexOfAny
  If strParam1 <> String.Empty Then
    If strParam2 <> String.Empty And strParam3 <> String.Empty Then
      HandleResultsLabel("IndexOfAny", AddQuotes(strParam1), _
                                        strParam2, strParam3)
      lblResults.Text = strSample.IndexOfAny(strParam1.ToCharArray, _
                    CInt(strParam2), CInt(strParam3)).ToString
    ElseIf strParam2 <> String.Empty Then
      HandleResultsLabel("IndexOfAny", AddQuotes(strParam1), _
                                        strParam2)
      lblResults.Text = strSample.IndexOfAny(strParam1.ToCharArray, _
                                        CInt(strParam2)).ToString
    Else
      HandleResultsLabel("IndexOfAny", AddQuotes(strParam1))
      lblResults.Text = _
            strSample.IndexOfAny(strParam1.ToCharArray).ToString
    End If
  End If
```

```
Case SelectedMethod.Insert
    HandleResultsLabel("Insert", strParam1, AddQuotes(strParam2))
    lblResults.Text = strSample.Insert(CInt(strParam1), strParam2)

Case SelectedMethod.Join
        ' Đoạn code này nhận một danh sách các giá trị cách
        ' nhau bằng dấu phẩy.
    HandleResultsLabel("Join", strParam1, AddQuotes(strParam2))
    Dim astrValues() As String
    astrValues = strParam2.Split(", ".ToCharArray)
    lblResults.Text = String.Join(strParam1, astrValues)

Case SelectedMethod.LastIndexOf
    If strParam1 <> String.Empty Then
        If strParam2 <> String.Empty And strParam3 <> String.Empty Then
            HandleResultsLabel("LastIndexOf", AddQuotes(strParam1), _
                                              strParam2, strParam3)
            lblResults.Text = strSample.LastIndexOf(strParam1, _
                                CInt(strParam2), CInt(strParam3)).ToString
        ElseIf strParam2 <> String.Empty Then
            HandleResultsLabel("LastIndexOf", AddQuotes(strParam1), _
                                              strParam2)
            lblResults.Text = strSample.LastIndexOf(strParam1, _
                                              CInt(strParam2)).ToString
        Else
            HandleResultsLabel("LastIndexOf", AddQuotes(strParam1))
            lblResults.Text = strSample.LastIndexOf(strParam1).ToString
        End If
    End If

Case SelectedMethod.LastIndexOfAny
    If strParam1 <> String.Empty Then
        If strParam2 <> String.Empty And strParam3 <> String.Empty Then
            HandleResultsLabel("LastIndexOfAny", AddQuotes(strParam1), _
                                strParam2, strParam3)
            lblResults.Text = trSample.LastIndexOfAny(_
                            strParam1.ToCharArray, CInt(strParam2), _
                            CInt(strParam3)).ToString
```

```
        ElseIf strParam2 <> String.Empty Then
          HandleResultsLabel("LastIndexOfAny", AddQuotes(strParam1), _
                                                  strParam2)
          lblResults.Text = strSample.LastIndexOfAny(_
                    strParam1.ToCharArray, CInt(strParam2)).ToString
        Else
          HandleResultsLabel("LastIndexOfAny", AddQuotes(strParam1))
          lblResults.Text = strSample.LastIndexOfAny(_
                                strParam1.ToCharArray).ToString
        End If
      End If

    Case SelectedMethod.PadLeft
      If strParam2 = String.Empty Then
          ' Nếu không có tham số thứ hai, String.PadLeft sử dụng
          ' ký tự khoảng trắng
        HandleResultsLabel("PadLeft", strParam1)
        lblResults.Text = strSample.PadLeft(CInt(strParam1))
      Else
        HandleResultsLabel("PadLeft", strParam1, AddQuotes(strParam2))
        lblResults.Text = strSample.PadLeft(CInt(strParam1), _
                                            CChar(strParam2))
      End If

    Case SelectedMethod.PadRight
          ' Nếu không có tham số thứ hai, String.PadRight sử dụng
          ' ký tự khoảng trắng
      If strParam2 = String.Empty Then
        HandleResultsLabel("PadRight", strParam1)
        lblResults.Text = strSample.PadRight(CInt(strParam1))
      Else
        HandleResultsLabel("PadRight", strParam1, AddQuotes(strParam2))
        lblResults.Text = strSample.PadRight(CInt(strParam1), _
                                             CChar(strParam2))
      End If

    Case SelectedMethod.Remove
      HandleResultsLabel("Remove", strParam1, strParam2)
      lblResults.Text = strSample.Remove(CInt(strParam1), CInt(strParam2))
```

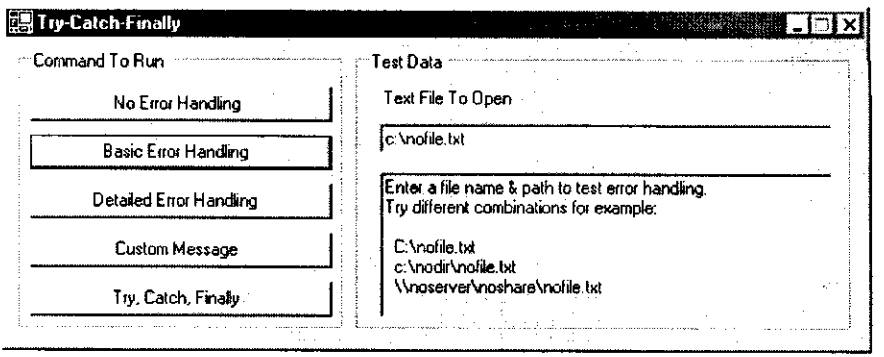
```
Case SelectedMethod.Replace
   HandleResultsLabel("Replace", AddQuotes(strParam1), _
                                 AddQuotes(strParam2))
   lblResults.Text = strSample.Replace(strParam1, strParam2)

Case SelectedMethod.Split
      ' Hiển thị một mảng chuỗi, được định dạng như sau:
      ' {"item1", "item2", "item3"}
      ' Ví dụ này sử dụng phương thức Join và Split
   If strParam2 = String.Empty Then
      HandleResultsLabel("Split", AddQuotes(strParam1))
      lblResults.Text = "{" & QUOTE & String.Join(QUOTE & "," & QUOTE, _
               strSample.Split(strParam1.ToCharArray)) & QUOTE & "}"
   Else
      HandleResultsLabel("Split", AddQuotes(strParam1), strParam2)
      lblResults.Text = "{" & QUOTE & String.Join(QUOTE & "," & QUOTE, _
                  strSample.Split(strParam1.ToCharArray, _
                  CInt(strParam2))) & QUOTE & "}"
   End If

Case SelectedMethod.StartsWith
   HandleResultsLabel("StartsWith", AddQuotes(strParam1))
   lblResults.Text = strSample.StartsWith(strParam1).ToString

Case SelectedMethod.Substring
      ' Nếu để trống tham số thứ hai, String.Substring trả về tất
      ' cả các ký tự còn lại
   If strParam2 = String.Empty Then
      HandleResultsLabel("Substring", strParam1)
      lblResults.Text = strSample.Substring(CInt(strParam1))
   Else
      HandleResultsLabel("Substring", strParam1, strParam2)
      lblResults.Text = strSample.Substring(CInt(strParam1), _
                                    CInt(strParam2))
   End If

Case SelectedMethod.ToLower
   HandleResultsLabel("ToLower")
   lblResults.Text = strSample.ToLower
```

```
        Case SelectedMethod.ToUpper
            HandleResultsLabel("ToUpper")
            lblResults.Text = strSample.ToUpper

        Case SelectedMethod.Trim
            ' Nếu không truyền vào tham số, Trim mặc định là là bỏ khoảng trắng
            If strParam1 = String.Empty Then
                HandleResultsLabel("Trim")
                lblResults.Text = strSample.Trim()
            Else
                HandleResultsLabel("Trim", AddQuotes(strParam1))
                lblResults.Text = strSample.Trim(strParam1.ToCharArray())
            End If
        Case SelectedMethod.TrimEnd
            ' Nếu không có tham số, TrimEnd chỉ bỏ khoảng-trắng
            If strParam1 = String.Empty Then
                HandleResultsLabel("TrimEnd")
                lblResults.Text = strSample.TrimEnd()
            Else
                HandleResultsLabel("TrimEnd", AddQuotes(strParam1))
                lblResults.Text = strSample.TrimEnd(strParam1.ToCharArray())
            End If

        Case SelectedMethod.TrimStart
            ' Nếu không truyền tham số, TrimStart chỉ bỏ khoảng trắng
            If strParam1 = String.Empty Then
                HandleResultsLabel("TrimStart")
                lblResults.Text = strSample.TrimStart()
            Else
                HandleResultsLabel("TrimStart", AddQuotes(strParam1))
                lblResults.Text = strSample.TrimStart(strParam1.ToCharArray())
            End If
    End Select

    Catch exp As Exception
        ' Nếu có lỗi xảy ra, hiển thị thông tin
        lblResultsLabel.Text = "<<ERROR>>"
        lblResults.Text = exp.Message
    End Try
End Sub
```

```
    Private Sub RefreshText()
        txtSample.Text = mstrSample
    End Sub

    Private Sub SetResultsLabel(ByVal Source As String, ByVal Method As String,
    ByVal Parameters As String)
        lblResultsLabel.Text = String.Format("Results of {0}.{1}({2})", Source, Method,
    Parameters)
    End Sub
End Class
```

41. Try – Catch – Finally

Chương trình này biểu diễn cách sử dụng cấu trúc bắt lỗi mới trong Visual Basic .NET: **Try – Catch – Finally**.



Các chức năng chính:

Chương trình có năm nút. Mỗi nút thực hiện mở một file chỉ định trong TextBox có nhãn **Text File To Open**. Mỗi nút có một kiểu bắt lỗi khác nhau sử dụng khối **Try-Catch-Finally**, ngoại trừ nút có nhãn **No Error Handing**.

Project: **Try-Catch-Finally**

frmMain.vb

```
Option Strict On
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form
```

```
' Code phát sinh form được bỏ qua

    ' Đưa nội dung vào văn bản bên dưới TextBox tên file
  Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
  Handles MyBase.Load
      ' Code bên dưới sử dụng một StringWriter để tạo một chuỗi trong bộ nhớ.
    Dim txt As StringWriter = New StringWriter()

    With txt
      .WriteLine("Enter a file name & path to test error handling. ")
      .WriteLine("Try different combinations for example:")
      .WriteLine("")
      .WriteLine("  C:\nofile.txt")
      .WriteLine("  c:\nodir\nofile.txt")
      .WriteLine("  \\noserver\noshare\nofile.txt")
    End With

      ' Chuyển StringWriter thành chuỗi và hiển thị nó
    Me.txtMessage.Text = txt.ToString()
  End Sub

  Private Sub cmdNoTryCatch_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdNoTryCatch.Click
      ' Hỏi để bảo đảm là người sử dụng muốn chạy chức năng này
    Dim strMsg As String = "The following code has no error handling and will cause
  an unhandled exception if a file is not found. Do you want to continue?"
    If MessageBox.Show(strMsg, Me.Text, MessageBoxButtons.YesNo, _
            MessageBoxIcon.Question, MessageBoxDefaultButton.Button2) = _
                                                    DialogResult.Yes Then
        ' Sử dụng class FileStream trong namespcae System.IO
      Dim fs As FileStream

        ' Lệnh này sẽ bị lỗi nếu file không tồn tại
      fs = File.Open(Me.txtFileName.Text, FileMode.Open)
      MessageBox.Show("The size of the file is: " & fs.Length, Me.Text, _
                          MessageBoxButtons.OK, MessageBoxIcon.Information)
      fs.Close()
    End If
  End Sub
```

```
Private Sub cmdBasicTryCatch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdBasicTryCatch.Click
    ' Thủ tục này thực hiện bắt lỗi với khối Try-Catch cơ bản
    Dim fs As FileStream
    Try
        ' Lệnh này sẽ gây ra lỗi nếu file không tồn tại
        fs = File.Open(Me.txtFileName.Text, FileMode.Open)
        MessageBox.Show("The size of the file is: " & fs.Length, Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
        fs.Close()

    Catch exp As Exception
        ' Sẽ bắt tất cả các lỗi không rõ ràng
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Stop)
    End Try
End Sub

Private Sub cmdDetailedTryCatch_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdDetailedTryCatch.Click
    ' Thủ tục này thực hiện Try-Catch bắt các lỗi liên quan đến IO
    Dim fs As FileStream

    Try
        ' Lệnh này gây ra lỗi nếu file không tồn tại
        fs = File.Open(Me.txtFileName.Text, FileMode.Open)
        MessageBox.Show("The size of the file is: " & fs.Length, Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
        fs.Close()

    Catch exp As FileNotFoundException
        ' Sẽ bắt lỗi khi file yêu cầu không tồn tại
        MessageBox.Show("The file you requested does not exist.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Stop)

    Catch exp As DirectoryNotFoundException
        ' Bắt lỗi khi thư mục yêu cầu không tồn tại
        MessageBox.Show("The directory you requested does not exist.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Stop)
```

```
Private Sub cmdBasicTryCatch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdBasicTryCatch.Click
    ' Thủ tục này thực hiện bắt lỗi với khối Try-Catch cơ bản
    Dim fs As FileStream
    Try
        ' Lệnh này sẽ gây ra lỗi nếu file không tồn tại
        fs = File.Open(Me.txtFileName.Text, FileMode.Open)
        MessageBox.Show("The size of the file is: " & fs.Length, Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
        fs.Close()

    Catch exp As Exception
        ' Sẽ bắt tất cả các lỗi không rõ ràng
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Stop)
    End Try
End Sub

Private Sub cmdDetailedTryCatch_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdDetailedTryCatch.Click
    ' Thủ tục này thực hiện Try-Catch bắt các lỗi liên quan đến IO
    Dim fs As FileStream

    Try
        ' Lệnh này gây ra lỗi nếu file không tồn tại
        fs = File.Open(Me.txtFileName.Text, FileMode.Open)
        MessageBox.Show("The size of the file is: " & fs.Length, Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
        fs.Close()

    Catch exp As FileNotFoundException
        ' Sẽ bắt lỗi khi file yêu cầu không tồn tại
        MessageBox.Show("The file you requested does not exist.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Stop)

    Catch exp As DirectoryNotFoundException
        ' Bắt lỗi khi thư mục yêu cầu không tồn tại
        MessageBox.Show("The directory you requested does not exist.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Stop)
```

```
    Catch exp As IOException
        ' Các lỗi IO chung
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                                                MessageBoxIcon.Stop)

    Catch exp As Exception
        ' Bắt các lỗi khác
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Stop)
    End Try
End Sub

Private Sub cmdCustomMessage_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdCustomMessage.Click
    ' Thực hiện bắt lỗi với khối Try-Catch để bắt các lỗi IO
    Dim fs As FileStream

    Try
        ' Lệnh này sẽ gây ra lỗi nếu file không tồn tại
        fs = File.Open(Me.txtFileName.Text, FileMode.Open)
        MessageBox.Show("The size of the file is: " & fs.Length, Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
        fs.Close()

    Catch exp As IOException
        ' Bắt bất kỳ lỗi IO nào
        Dim strMsg As String
        strMsg = "I was unable to open the file you requested, " & _
                 Me.txtFileName.Text & vbCrLf & vbCrLf & _
         "Detailed Error Information below:" & vbCrLf & vbCrLf & _
         " Message: " & exp.Message & vbCrLf & _
         " Source: " & exp.Source & vbCrLf & vbCrLf & _
         " Stack Trace:" & vbCrLf

        Dim strStackTrace As String

        ' Truy xuất đế đối tượng StackTree có thể gây ra exception vì vậy cần
        ' đặt đoạn code trong một khối Try-Cath
        Try
            strStackTrace = exp.StackTrace()
```

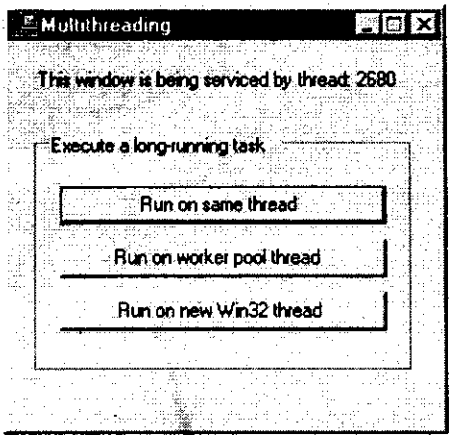
```
        Catch stExp As Security.SecurityException
            strStackTrace = "Unable to access stack trace due to security restrictions."

        Catch stExp As Exception
            strStackTrace = "Unable to access stack trace."
        End Try

        strMsg = strMsg & strStackTrace
        MessageBox.Show(strMsg, Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Stop)

    Catch exp As System.Exception
        ' Catch này bắt tất cả các lỗi không mong đợi
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Stop)

    End Try
End Sub

Private Sub cmdTryCatchFinally_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdTryCatchFinally.Click
    ' Thủ tục này thực  hiện một khối cơ bản Try-Catch-Finally.
    Dim fs As FileStream

    Try
        ' Lệnh này sẽ gây ra lỗi nếu file không tồn tại
        fs = File.Open(Me.txtFileName.Text, FileMode.Open)
        MessageBox.Show("The size of the file is: " & fs.Length, Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)

    Catch exp As Exception
        ' Bắt tất cả các lỗi xảy ra
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Stop)

    Finally
        ' Nếu không mở file thành công, tham chiếu sẽ là Nothing.
        If Not fs Is Nothing Then
            fs.Close()
```

```
            MessageBox.Show("File closed successfully in Finally block", Me.Text, _
                            MessageBoxButtons.OK, MessageBoxIcon.Information)
        End If
    End Try
  End Sub
End Class
```

Phần VIII - Các chủ đề cao cấp

42. Đa tiểu trình – Multithreading

Chương trình này chỉ cách sử dụng đa tiểu trình để thực hiện nhiều công việc đồng thời. Chương trình này không có đồng bộ, bởi vì không có dữ liệu được xử lý bởi nhiều tiểu trình.



Các chức năng chính:

Chương trình này cho phép làm nhiều công việc trên các tiểu trình khác nhau. Nút đầu tiên thực hiện công việc trong cùng tiểu trình với ứng dụng, người sử dụng không thể tương tác với form chính cho đến khi công việc kết thúc.

Nút thứ hai và nút thứ ba thực hiện công việc trong các tiểu trình khác, cho phép người sử dụng tiếp tục tương tác với form. Nút thứ hai thực hiện công việc từ một tiểu trình trong ứng dụng (worker pool). Và nút thứ ba tạo một tiểu trình Win32 mới và thực hiện công việc trong nó.

Bởi vì việc debug các chương trình Multithreading rất phức tạp, nên các thuộc tính **DebuggerStepThrough** được áp dụng tại các điểm có sử dụng nhiều tiểu trình để bỏ qua debug.

Project: **Da tieu trinh - Multithreading**

frmTaskProgress.vb

```
Option Strict On
Public Class frmTaskProgress
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub frmTaskProgress_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        'Hiển thị ID của tiểu trình gọi form này
        Me.lblThreadID.Text &= CStr(AppDomain.GetCurrentThreadId())
    End Sub
End Class
```

frmMain.vb

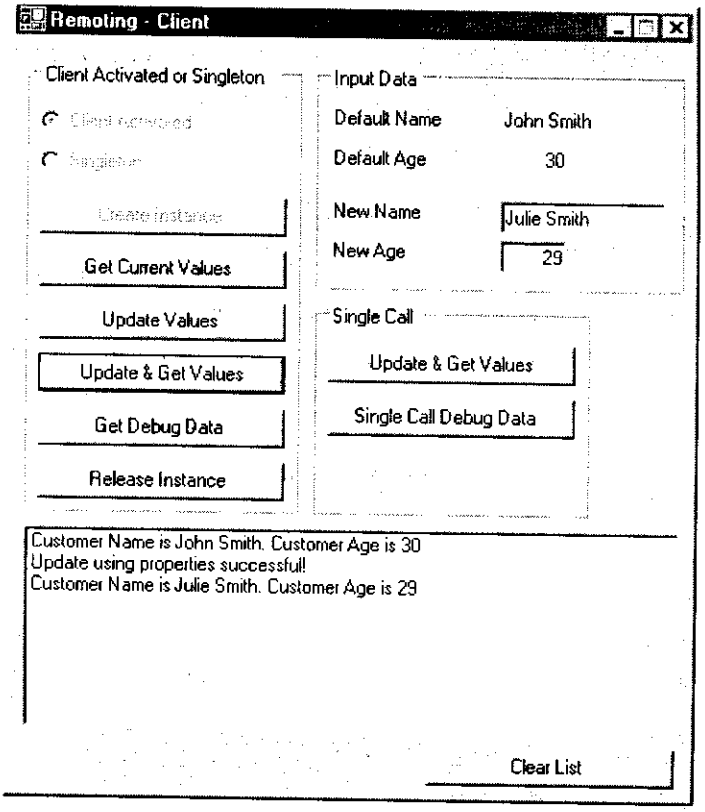
```
Option Strict On
Imports System.Threading ' Namespce chứa class Thread

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub frmMain_Load(ByVal sender As System.Object, _
                ByVal e As System.EventArgs) _
                Handles MyBase.Load
        ' Hiển thị ID tiểu trình chính của ứng dụng

        lblThreadID.Text &= CStr(AppDomain.GetCurrentThreadId())
    End Sub

    Private Sub cmdSameThread_Click(ByVal sender As System.Object, _
                    ByVal e As System.EventArgs) _
                    Handles cmdSameThread.Click
        ' Thực hiện công việc trong cùng một tiểu trình quản lý cửa sổ frmMain
        TheLongRunningTask()
    End Sub
```

```
' Để chạy ứng dụng trong một tiểu trình worker pool, sử dụng một hàm ủy quyền.
Delegate Sub TaskDelegate()

<DebuggerStepThrough()> Private Sub cmdWorkerPoolThread_Click( _
                        ByVal sender As System.Object, _
                        ByVal e As System.EventArgs) _
                        Handles cmdWorkerPoolThread.Click
    ' Để thực hiện công việc trong một tiểu trình worker pool, tạo một instance
    ' của một hàm ủy quyền, sau đó gọi BeginInvoke trên ủy quyền đó
    Dim td As New TaskDelegate(AddressOf TheLongRunningTask)
    td.BeginInvoke(Nothing, Nothing)

End Sub

<DebuggerStepThrough()> Private Sub cmdRunOnNewWin32Thread_Click( _
                        ByVal sender As System.Object, _
                        ByVal e As System.EventArgs) _
                        Handles cmdRunOnNewWin32Thread.Click
    ' Để thực hiện công việc trong một tiểu trình mới của hệ điều hành,
    ' tạo một instance mới của tiểu trình quản lý công việc.

    ' Tạo mới một tiểu trình
    Dim t As New Thread(AddressOf TheLongRunningTask)
    ' Bắt đầu thực hiện tiểu trình mới
    t.Start()
End Sub

<DebuggerStepThrough()> Private Sub TheLongRunningTask()
    ' Phương thức này giả lập một công việc tốn nhiều thời gian. Để đại diện cho
    ' công việc đang được thực hiện, một form với một thanh progress sẽ hiển thị
    ' trong quá trình thực hiện. Sau khi hoàn tất công việc form tự đóng lại.

    Dim f As New frmTaskProgress()
    f.Show()
    f.Refresh()

    ' Tăng dần thanh progress
    Dim i As Integer
```

```
        For i = 1 To 10
            f.prgTaskProgress.Value += 10
            ' Đợi nữa giây
            Thread.CurrentThread.Sleep(500)
        Next

        ' Đóng form sau khi hoàn tất công việc
        f.Hide()
        f.Dispose()
    End Sub
End Class
```

43. TCP Remoting

Ví dụ này được thiết kế để chỉ cách sử dụng kiến trúc .NET Remoting sử dụng ba kiểu đối tượng từ xa. Ngoài rá, nó sử dụng các file cấu hình XML để chỉ rõ các đối tượng và thông tin server thay vì đưa chúng vào code hay để trong registry.



Các chức năng chính:

Ví dụ này chứa 3 solution:

- Remoting – Server
- Remoting – Host
- Remoting – Client

Mỗi solution chứa một project.

Một trong các project (RemoteCustomer) là một Class Library cung cấp ba kiểu đối tượng có thể tạo:

- Customer (client activated)
- SingleCallCustomer (WellKnown singlecall)
- SingleonCustomer (WellKnown singleon)

Host Project chứa một ứng dụng Windows Forms đọc file cấu hình XML có tên **Host.exe.config** để làm cho các class đã định nghĩa trong project RemoteCustomer sẵn sàng phục vụ.

Client project cũng có một ứng dụng Windows Forms. Nó nhờ vào file cấu hình có tên Client.exe.config để biết nơi cần tìm các đối tượng từ xa.

Trước khi chạy ứng dụng, cần build cả ba project. Build theo thứ tự sau:

- RemoteCustomer
- Host
- Client

Khi đã build thành công, chạy ứn dụng Host trước để bảo đảm các đối tượng sẵn sàng phục vụ từ xa. Tiếp theo chạy Client.

Nếu muốn chạy thử trên hai máy, cần thay đổi file cấu hình .config của client, đổi localhost thành địa chỉ IP hoặc tên DNS của máy server. Ngoài ra, cần thay đổi file cấu hình của server nếu muốn thay đổi (số) cổng TCP.

Project (Class Library): **RemoteCustomer**

Customer.vb

```
Option Strict On
Namespace RemotingSample
    Public Class Customer : Inherits MarshalByRefObject

        ' Các biến Private để lưu các giá trị thuộc tính của Customer
        Private m_Age As Byte = 0
        Private m_Name As String = "<<Uninitialized>>"
```

```
    ' Biến Private để lưu thời gian object instance được tạo
  Private ReadOnly mdteCreationTime As DateTime = Date.Now
    ' Constructor mặc nhiên không có đối số
  Public Sub New()
    MyBase.New()
  End Sub

    ' Constructor có tham số chỉ được gọi bởi các Client Activated Object
  Public Sub New(ByVal NewName As String, ByVal NewAge As Byte)
    Me.Name = NewName
    Me.Age = NewAge
  End Sub

    ' Cài đặt cho thuộc tính Age
  Public Property Age() As Byte
    Get
      Return m_Age
    End Get

    Set(ByVal Value As Byte)
      If Value > 0 Then
        m_Age = Value
      Else
        m_Age = 0
      End If
    End Set
  End Property

    ' Cài đặt thuộc tính Name
  Public Property Name() As String
    Get
      Return m_Name
    End Get
    Set(ByVal Value As String)
      m_Name = Value
    End Set
  End Property
```

```
        ' Trả về Customer Name và Age trong một chuỗi
        Public Function GetCustomerInfo() As String
            Return String.Format("Customer Name is {0}. Customer Age is {1}", _
                                 Me.Name, Me.Age)
        End Function


        ' Chấp nhận thay đổi Name & Age và trả về các gia trị trong một chuỗi
        Public Function UpdateCustomerInfo(ByVal NewName As String, ByVal NewAge
    As Byte) As String
            ' Cập nhật các thuộc tính cục bộ
            Me.Name = NewName
            Me.Age = NewAge
            ' Thực hiện công việc chính ở đây, ví dụ như cập nhật database
            Return String.Format("Customer Name is {0}. Customer Age is {1}", _
                                 Me.Name, Me.Age)
        End Function

        ' Trả về ngày và thời gian được tạo của đối tượng hiện hành
        Public ReadOnly Property DebugCreationTime() As DateTime
            Get
                Return mdteCreationTime
            End Get
        End Property

        ' Trả về tên máy tính của đối tượng đang chạy trên nó
        Public ReadOnly Property DebugHostName() As String
            Get
                Return System.Environment.MachineName
            End Get
        End Property
    End Class
End Namespace
```

SingleCallCustomer.vb

```
Option Strict On
Namespace RemotingSample
    Public Class SingleCallCustomer
        Inherits MarshalByRefObject
```

```
    Private m_Age As Byte = 0
    Private m_Name As String = "<<Uninitialized>>"

    Private ReadOnly mdteCreationTime As DateTime = Date.Now
    Public Sub New()
      MyBase.New()
    End Sub

    Public Property Age() As Byte
      Get
        Return m_Age
      End Get

      Set(ByVal Value As Byte)
        If Value > 0 Then
          m_Age = Value
        Else
          m_Age = 0
        End If
      End Set
    End Property

    Public Property Name() As String
      Get
        Return m_Name
      End Get

      Set(ByVal Value As String)
        m_Name = Value
      End Set
    End Property

    Public Function UpdateCustomerInfo(ByVal NewName As String, ByVal NewAge
As Byte) As String
      Me.Name = NewName
      Me.Age = NewAge
      Return String.Format("Customer Name is {0}. Customer Age is {1}", _
                           Me.Name, Me.Age)
    End Function
```

```
    Public ReadOnly Property DebugCreationTime() As DateTime
      Get
        Return mdteCreationTime
      End Get
    End Property

    Public ReadOnly Property DebugHostName() As String
      Get
        Return System.Environment.MachineName
      End Get
    End Property
  End Class
End Namespace
```

SingletonCustomer.vb

```
Option Strict On
Namespace RemotingSample
  Public Class SingletonCustomer
    Inherits MarshalByRefObject
    Implements IDisposable
    Private m_Age As Byte = 0
    Private m_Name As String = "<<Uninitialized>>"
    Private ReadOnly mdteCreationTime As DateTime = Date.Now()
     ' Biến để lưu số người sử dụng đã kết nối vào
    Private m_Connected As Integer

    Public Sub New()
      MyBase.New()
    End Sub

    Public Sub NewClient()
       ' Cập nhật số người sử dụng đã kết nối
      m_Connected += 1
    End Sub

    Public Property Age() As Byte
      Get
        Return m_Age
      End Get
```

```
        Set(ByVal Value As Byte)
            If Value > 0 Then
                m_Age = Value
            Else
                m_Age = 0
            End If
        End Set
    End Property

    Public Property Name() As String
        Get
            Return m_Name
        End Get
        Set(ByVal Value As String)
            m_Name = Value
        End Set
    End Property

    Public Function GetCustomerInfo() As String
        Return String.Format("Customer Name is {0}. Customer Age is {1}", _
                             Me.Name, Me.Age)
    End Function

    Public Function UpdateCustomerInfo(ByVal NewName As String, ByVal NewAge
As Byte) As String
        Me.Name = NewName
        Me.Age = NewAge

        Return String.Format("Customer Name is {0}. Customer Age is {1}", _
                             Me.Name, Me.Age)
    End Function

    Public ReadOnly Property DebugCreationTime() As DateTime
        Get
            Return mdteCreationTime
        End Get
    End Property
```

```
    Public ReadOnly Property DebugHostName() As String
      Get
        Return System.Environment.MachineName
      End Get
    End Property

    ' Trả về số client hiện hành kết nối vào đối tượng này
    Public ReadOnly Property Connected() As Integer
      Get
        Return m_Connected
      End Get
    End Property

    Public Sub Dispose() Implements System.IDisposable.Dispose
      ' Khi một client chấm dứt sử dụng, giảm biến đếm
      m_Connected -= 1
    End Sub
  End Class
End Namespace
```

Project: **Host**

References: **RemoteCustomer.dll** (được biên dịch từ project trên)

Để cho client 'load' các đối tượng, nó đọc file Host.exe.config khi bắt đầu chương trình. Chương trình này giả định file này nằm trong cùng thư mục với file .exe. Nội dung file này được lưu trong file app.coniig nằm ở thư mục chứa source. Trước khi biên dịch, chép file này vào thư mục chứa file .exe với tên mới là Host.exe.config.

app.config

```
<configuration>
  <system.runtime.remoting>
    <application>
      <service>
        <activated type="RemotingSample.Customer,RemoteCustomer"/>
        <wellknown
          type="RemotingSample.SingleCallCustomer,RemoteCustomer"
          objectUri="SingleCallCustomer"
          mode="SingleCall"
```

```
        />
        <wellknown
          type="RemotingSample.SingletonCustomer,RemoteCustomer"
          objectUri="SingletonCustomer"
          mode="Singleton"
        />
      </service>
      <channels>
      <channel ref="tcp" port="8080">
        <serverProviders>
          <formatter ref="binary"/>
        </serverProviders>
      </channel>
      </channels>
      <!--
      Below is an example of using a different port.
       -->
      <!--
      <channels>
      <channel ref="tcp" port="9090">
        <serverProviders>
          <formatter ref="binary"/>
        </serverProviders>
      </channel>
      </channels>
   -->

    </application>
  </system.runtime.remoting>
</configuration>
```

frmMain.vb

```
Option Strict On
Imports System.IO
Imports System.Runtime.Remoting

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua
```

```
' Hàm này duyệt qua tất cả các kiểu ClientActivatedService mà được load
' thông qua RemotingConfiguration.Configure(Remoting.config)
Private Sub ListClientActivatedServiceTypes()
    Dim entry As ActivatedServiceTypeEntry
    For Each entry In RemotingConfiguration.GetRegisteredActivatedServiceTypes()
        Me.lstOutput.Items.Add("Registered ActivatedServiceType: " & _
                                                    entry.TypeName)
    Next
End Sub

' Hàm này duyệt qua tất cả các kiểu WellKnownService mà được load thông
' qua RemotingConfiguration.Configure(Remoting.config)
Private Sub ListWellKnownServiceTypes()
    Dim entry As WellKnownServiceTypeEntry
    For Each entry In _
            RemotingConfiguration.GetRegisteredWellKnownServiceTypes()
        Me.lstOutput.Items.Add(entry.TypeName & " is available at " & entry.ObjectUri)
    Next
End Sub

Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    Try
        ' Đọc file Host.exe.config
        ' Gọi RemotingConfiguration.Configure đọc file cấu hình (xml) và để kiến trúc
        ' remoting biết những kiểu nào sẽ làm cho sẵn sàng phục vụ
        Me.lstOutput.Items.Add("Loading Activated Configuration File")

        RemotingConfiguration.Configure("Host.exe.config")

        ' Sau khi load file remoting.config, liệt kê danh sách các kiểu ClientActivated
        ' và các kiểu WellKnown vào ListBox trên form
        Me.ListClientActivatedServiceTypes()
        Me.ListWellKnownServiceTypes()

    Catch exp As Exception
        ' Bắt lỗi tổng quát
        Dim txt As String
```

```
        txt = "I was unable to load the file remoting.config or it is not in the correct
    format." & vbCrLf & _
        "Please make sure it is located in the same directory as this exe " & _
        " and that it is in the correct format." & vbCrLf & vbCrLf & _
        "Detailed Error Information below:" & vbCrLf & vbCrLf & _
        "  Message: " & exp.Message & vbCrLf & _
        "  Source: " & exp.Source & vbCrLf & vbCrLf & _
        "  Stack Trace:" & vbCrLf & _
        exp.StackTrace

        MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
    MessageBoxIcon.Stop)

        Me.lstOutput.Items.Add("Unabled to load objects.")
      End Try

    End Sub
End Class
```

Project: **Client**

References: RemoteCustomer.dll

app.config

```
<configuration>
  <system.runtime.remoting>
    <application>
      <client url="tcp://localhost:8080">
      <activated type="RemotingSample.Customer,RemoteCustomer"/>
        <wellknown type="RemotingSample.SingleCallCustomer,RemoteCustomer"
    url="tcp://localhost:8080/SingleCallCustomer" />
      <wellknown type="RemotingSample.SingletonCustomer,RemoteCustomer"
    url="tcp://localhost:8080/SingletonCustomer" />
      </client>
      <!--
      Below is an example using a DNS Entry with a different port.
      You will need to uncomment the block to use it (as well as
      comment the above block).
      -->

      <!-- <client url="tcp://www.myserver.com:9090">
```

```
    <activated type="RemotingSample.Customer,RemoteCustomer"/>
      <wellknown type="RemotingSample.SingleCallCustomer,RemoteCustomer"
  url="tcp://www.myserver.com:9090/SingleCallCustomer" />
    <wellknown type="RemotingSample.SingletonCustomer,RemoteCustomer"
  url="tcp://www.myserver.com:9090/SingletonCustomer" />
    </client>
    -->
  </application>
 </system.runtime.remoting>
</configuration>
```

frmMain.vb

```
Option Strict On
Imports System.Runtime.Remoting
Imports RemotingSample

Public Class frmMain
  Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

  ' Sử dụng để tham chiếu đến một kiểu Client Activated
  Private mCustomer As Customer

  ' Biến này được sử dụng khi muốn nhiều clients kết nối vào cùng một
  ' object instance trên server
  Private mSCustomer As SingletonCustomer

  Private Sub SetCtlState(ByVal NewState As Boolean)
    ' Enable nút Create
    Me.cmdCreate.Enabled = NewState

    ' Disable các nút khác
    Me.cmdGet.Enabled = (Not NewState)
    Me.cmdUpdate.Enabled = (Not NewState)
    Me.cmdUpdateAndGet.Enabled = (Not NewState)
    Me.cmdDebugData.Enabled = (Not NewState)
    Me.cmdRelease.Enabled = (Not NewState)
```

```
    ' Quy định các radio button
    Me.rbClient.Enabled = NewState
    Me.rbSingle.Enabled = NewState
End Sub

Private Sub cmdClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdClear.Click
    ' Xóa dữ liệu trong listbox
    Me.lstResponses.Items.Clear()
End Sub

Private Sub cmdCreate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdCreate.Click
    Dim txt As String
    Try
        Dim objNotCreated As Boolean = False

        If Me.rbClient.Checked = True Then
            ' Sử dụng Client Activated Type
            ' Chú ý là lệnh New này có thể truyền vào các tham số cho Constructor
            mCustomer = New Customer(Me.lblDefNameValue.Text, _
                                     CByte(Me.lblDefAgeValue.Text))
            objNotCreated = (mCustomer Is Nothing)
        Else
            ' Sử dụng Server Activated Type
            ' Không thể truyền tham số vào constructor trong trường hợp này.
            ' Bởi vì server tạo instance và làm cho nó sẵn sàng phục vụ, ta chỉ cần có
            ' một tham chiếu đến instance đang chạy thông qua một trung gian (proxy).
            Dim args() As Object
            mSCustomer = CType(Activator.CreateInstance(_
                        GetType(RemotingSample.SingletonCustomer), args), _
                        RemotingSample.SingletonCustomer)
            mSCustomer.NewClient()
            objNotCreated = (mSCustomer Is Nothing)
        End If

        ' Thay đổi trạng thái các nút
        SetCtlState(objNotCreated)
```

```
    Catch exp As RemotingException
        ' Bắt các exception Remoting
      txt = "I was unable to access the remote customer object." & vbCrLf & _
       "Detailed Error Information below:" & vbCrLf & vbCrLf & _
       " Message: " & exp.Message & vbCrLf & _
       " Source: " & exp.Source & vbCrLf & vbCrLf & _
       " Stack Trace:" & vbCrLf & _
       exp.StackTrace()

      MessageBox.Show(txt, "Remoting Exception", MessageBoxButtons.OK, _
                      MessageBoxIcon.Stop)

    Catch exp As System.Net.Sockets.SocketException
        ' Bắt các exception Socket
      txt = "I was unable to access the remote customer object." & vbCrLf & _
       "Is it possible the server is not running?" & vbCrLf & vbCrLf & _
       "Detailed Error Information below:" & vbCrLf & vbCrLf & _
       " Message: " & exp.Message & vbCrLf & _
       " Source: " & exp.Source & vbCrLf & _
       " Error Code: " & exp.ErrorCode.ToString() & vbCrLf & _
       " Native Error Code: " & exp.NativeErrorCode.ToString() & vbCrLf & _
       " Stack Trace:" & vbCrLf & _
       exp.StackTrace

      MessageBox.Show(txt, "Socket Exception", MessageBoxButtons.OK, _
                      MessageBoxIcon.Stop)

    Catch exp As Exception
        ' Bắt lỗi tổng quát
      txt = "I was unable to access the remote customer object." & vbCrLf & _
       "Detailed Error Information below:" & vbCrLf & vbCrLf & _
       " Message: " & exp.Message & vbCrLf & _
       " Source: " & exp.Source & vbCrLf & vbCrLf & _
       " Stack Trace:" & vbCrLf & _
       exp.StackTrace
      MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
MessageBoxIcon.Stop)

    End Try
End Sub
```

```
Private Sub cmdDebugData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdDebugData.Click
    ' Lấy dữ liệu kiểm tra từ kiểu Client Activated
  If ((Not mCustomer Is Nothing) Or (Not mSCustomer Is Nothing)) Then
    Try
      If Me.rbClient.Checked = True Then
        With Me.lstResponses.Items
          .Add("Debug data follows:")
          .Add(String.Format("  Creation Time: {0}", _
                             mCustomer.DebugCreationTime.ToString))
          .Add(String.Format("  Remote Host Name: {0}", _
                             mCustomer.DebugHostName))
          .Add("End Debug Data")
        End With
      Else
        With Me.lstResponses.Items
          .Add("Debug data follows:")
          .Add(String.Format("  Creation Time: {0}", _
                             mSCustomer.DebugCreationTime.ToString))
          .Add(String.Format("  Remote Host Name: {0}", _
                             mSCustomer.DebugHostName))
          .Add(String.Format("  Number of connected clients: {0}", _
                             mSCustomer.Connected.ToString()))
          .Add("End Debug Data")
        End With
      End If
    Catch exp As Exception
        ' Bắt lỗi tổng quát
      Dim txt As String
      txt = "I was unable to access the remote customer object." & vbCrLf & _
        "Detailed Error Information below:" & vbCrLf & vbCrLf & _
      "  Message: " & exp.Message & vbCrLf & _
      "  Source: " & exp.Source & vbCrLf & vbCrLf & "  Stack Trace:" & vbCrLf & _
        exp.StackTrace
      MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK, _
                      MessageBoxIcon.Stop)
    End Try
  End If
End Sub
```

```
Private Sub cmdGet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdGet.Click
    ' Yêu cầu server dữ liệu của customer
    Try
      If Me.rbClient.Checked = True Then
        Me.lstResponses.Items.Add(mCustomer.GetCustomerInfo())
      Else
        Me.lstResponses.Items.Add(mSCustomer.GetCustomerInfo())
      End If

    Catch exp As Exception
      Dim txt As String
      txt = "I was unable to access the remote customer object." & vbCrLf & _
       "Detailed Error Information below:" & vbCrLf & vbCrLf & _
       " Message: " & exp.Message & vbCrLf & _
       " Source: " & exp.Source & vbCrLf & vbCrLf & _
       " Stack Trace:" & vbCrLf & _
       exp.StackTrace

      MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
MessageBoxIcon.Stop)

    End Try
  End Sub

  Private Sub cmdUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUpdate.Click
    ' Cập nhật dữ liệu trên Server sử dụng các thuộc tính public
    If ((Not mCustomer Is Nothing) Or (Not mSCustomer Is Nothing)) Then
      Try
        If Me.rbClient.Checked = True Then
          mCustomer.Name = Me.txtNewName.Text
          mCustomer.Age = CByte(Me.txtNewAge.Text)
        Else
          mSCustomer.Name = Me.txtNewName.Text
          mSCustomer.Age = CByte(Me.txtNewAge.Text)
        End If

        Me.lstResponses.Items.Add("Update using properties successful!")
```

```
        Catch exp As Exception
            Dim txt As String
            txt = "I was unable to access the remote customer object." & vbCrLf & _
             "Detailed Error Information below:" & vbCrLf & vbCrLf & _
             "  Message: " & exp.Message & vbCrLf & _
             "  Source: " & exp.Source & vbCrLf & vbCrLf & _
             "  Stack Trace:" & vbCrLf & _
             exp.StackTrace

            MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
MessageBoxIcon.Stop)

        End Try
    End If
End Sub

Private Sub cmdUpdateAndGet_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUpdateAndGet.Click
    ' Cập nhật các giá trị trên server sử dụng hàm nhận vào hai giá trị và nhận
    ' về dữ liệu của kiểu Client Activated
    If ((Not mCustomer Is Nothing) Or (Not mSCustomer Is Nothing)) Then
        Try
            If Me.rbClient.Checked = True Then
                With Me.lstResponses.Items
                    .Add(mCustomer.UpdateCustomerInfo(Me.txtNewName.Text, _
                        CByte(Me.txtNewAge.Text)))
                End With
            Else
                With Me.lstResponses.Items
                    .Add(mSCustomer.UpdateCustomerInfo(Me.txtNewName.Text, _
                        CByte(Me.txtNewAge.Text)))
                End With
            End If

        Catch exp As Exception
            Dim txt As String
            txt = "I was unable to access the remote customer object." & vbCrLf & _
             "Detailed Error Information below:" & vbCrLf & vbCrLf & _
             "  Message: " & exp.Message & vbCrLf & _
```

```
            " Source: " & exp.Source & vbCrLf & vbCrLf & _
            " Stack Trace:" & vbCrLf & _
            exp.StackTrace

        MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
MessageBoxIcon.Stop)
      End Try
    End If
  End Sub

  Private Sub cmdRelease_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdRelease.Click
    ' Giải phóng tham chiếu đến đối tượng trên server
    If ((Not mCustomer Is Nothing) Or (Not mSCustomer Is Nothing)) Then
      Dim objReleased As Boolean = False
      If rbClient.Checked = True Then
        mCustomer = Nothing
        objReleased = (mCustomer Is Nothing)
      Else
          ' Do đối tượng có cung cấp phương thức, ta nên gọi nó.
          ' Trong ví dụ này nó không làm gì quan trọng, chỉ đơn giản giảm biến đếm
        mSCustomer.Dispose()
        mSCustomer = Nothing
        objReleased = (mSCustomer Is Nothing)
      End If

       ' Đổi trạng thái các nút
      SetCtlState(objReleased)
    End If
  End Sub

  Private Sub cmdSingleCall_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSingleCall.Click
    ' Các đối tượng SingleCall chỉ sống trong phương thức gọi
    ' Trong khi các thuộc tính tồn tại, chúng chỉ có thể được sử dụng khi đối tượng
    ' không được sử dụng trong chế độ SingleCall

    ' Mặc dù ta không cung cấp các đối số, nhưng ta phải truyền vào một
    ' mảng các đối tượng. Nothing sẽ không làm việc được.
    Dim args() As Object
```

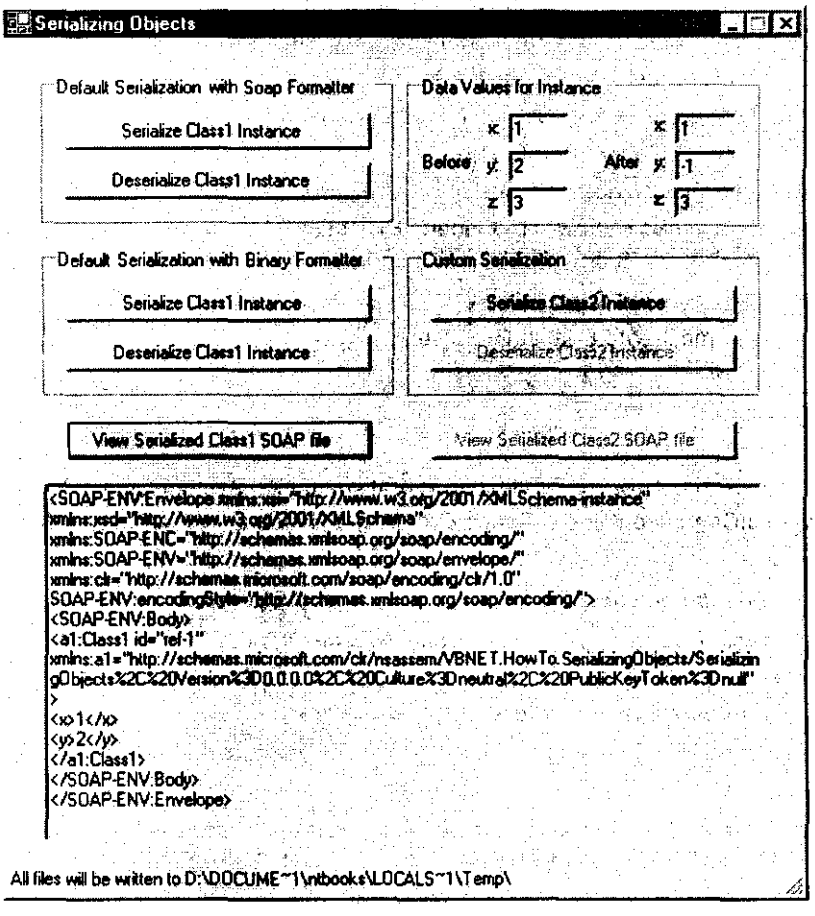
```
    Dim cust As SingleCallCustomer

    Try
      cust = CType(Activator.CreateInstance( _
                  GetType(RemotingSample.SingleCallCustomer), args), _
                  RemotingSample.SingleCallCustomer)
      With Me.lstResponses.Items
        .Add("SingleCall.UpdateCustomerInfo: " & _
             cust.UpdateCustomerInfo(Me.txtNewName.Text, _
             CByte(Me.txtNewAge.Text)))
        .Add("Update Successful!")
      End With
    Catch exp As Exception
      Dim txt As String
      txt = "I was unable to access the remote customer object." & vbCrLf & _
       "Detailed Error Information below:" & vbCrLf & vbCrLf & _
       " Message: " & exp.Message & vbCrLf & _
       " Source: " & exp.Source & vbCrLf & vbCrLf & _
       " Stack Trace:" & vbCrLf & _
       exp.StackTrace

      MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
MessageBoxIcon.Stop)
    End Try
  End Sub

  Private Sub cmdSingleDebug_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSingleDebug.Click
    ' Các đối tượng SingleCall chỉ sống trong phương thức gọi.
    ' Mỗi lần ta lấy dữ liệu debug, ta lấy nó từ một instance mới của đối tượng
    Dim args() As Object
    Dim scCust As SingleCallCustomer
    Try
      scCust = CType(Activator.CreateInstance( _
                  GetType(RemotingSample.SingleCallCustomer), args), _
                  RemotingSample.SingleCallCustomer)
      With Me.lstResponses.Items
        .Add("Debug data follows:")
        .Add(String.Format(" Creation Time: {0}", _
                  scCust.DebugCreationTime.ToString))
```

```
            .Add(String.Format("  Remote Host Name: {0}", scCust.DebugHostName))
              ' Chú ý sự khác nhau của thời gian tạo
            .Add(String.Format("  Creation Time: {0}", _
                                scCust.DebugCreationTime.ToString))
            .Add("End Debug Data")
          End With

        Catch exp As Exception
          Dim txt As String
          txt = "I was unable to access the remote customer object." & vbCrLf & _
           "Detailed Error Information below:" & vbCrLf & vbCrLf & _
           "  Message: " & exp.Message & vbCrLf & _
           "  Source: " & exp.Source & vbCrLf & vbCrLf & _
           "  Stack Trace:" & vbCrLf & _
           exp.StackTrace

          MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
MessageBoxIcon.Stop)

        End Try

      End Sub

      Private Sub frmMain_Closed(ByVal sender As Object, ByVal e As
System.EventArgs) Handles MyBase.Closed
        ' Giải phóng các đối tượng chưa được giải phóng
        If Not mCustomer Is Nothing Then
          mCustomer = Nothing
        End If
      End Sub

      Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        ' Đọc file cấu hình của ứng dụng (client.exe.config). File này chứa thông
        ' tin cấy hình remoting cho client.
        Try
          ' Giả sử file này trong cùng thư mục với file .exe
          RemotingConfiguration.Configure("client.exe.config")
```

```
    Catch exp As Exception
        Dim txt As String
        txt = "I was unable to load the file config.xml." & vbCrLf & _
          "Please make sure it is located in the same directory as this exe " & _
          " and that it is in the correct format." & vbCrLf & _
          "Detailed Error Information below:" & vbCrLf & vbCrLf & _
          "  Message: " & exp.Message & vbCrLf & _
          "  Source: " & exp.Source & vbCrLf & vbCrLf & _
          "  Stack Trace:" & vbCrLf & _
          exp.StackTrace
        MessageBox.Show(txt, "Generic Exception", MessageBoxButtons.OK,
MessageBoxIcon.Stop)
        ' Disable nút Create
        Me.cmdCreate.Enabled = False
    End Try
End Sub

Private Sub txtNewAge_Validating(ByVal sender As System.Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles txtNewAge.Validating

    ' Kiểm tra để bảo đảm chỉ có giá trị số được nhập vào và bảo
    ' đảm giá trị trong vùng giá trị của kiểu byte
    Try
        Dim d As Byte = CByte(Me.txtNewAge.Text)
    Catch exp As Exception
        Dim txt As String
        txt = "The value you entered, '{0}', for the Customer's New Age is incorrect." &
vbCrLf &  "Please enter a value in the range of 0 to 255." & vbCrLf & _
        "The value will be reset to 0 by default."
        MessageBox.Show(String.Format(txt, Me.txtNewAge.Text), Me.Text,
MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Me.txtNewAge.Text = "0"
        e.Cancel = True
    End Try
End Sub

End Class
```

44. Serializing và Deserializing các đối tượng

Chương trình này biểu diễn cách serialize (chuyển đối tượng thành Stream) và deserialize (chuyển stream thành đối tượng) một đối tượng.



Các chức năng chính:

- Cho phép người sử dụng serialize hai class, một cách serialize chuẩn và một cách serialize có tùy biến. Chương trình có sáu nút bấm được chia nhóm để thực hiện chuyển các đối tượng thành các stream và ngược lại. Hai nút ở phía dưới cho phép người sử dụng xem các SOAP envelope của các đối tượng đã được chuyển thành stream. Các hộp văn bản bên phải cho phép nhập các giá trị làm dữ liệu khởi tạo cho các instance. Các TextBox read-only (chỉ cho phép đọc) hiển thị các giá trị của các trường sau khi stream được chuyển lại thành đối tượng.

Project: **Serializing Objects**

References: System.Runtime.Serialization.Formatters.Soap.dll

Class1.vb

```
Option Strict On

' Class này được đánh dấu là Serializable. Trong ví dụ này, nó sẽ được chuyển
' sang một file và ngược lại. Nếu sử dụng định dạng SOAP, file có tên Class1File.xml.
' Nếu dùng định dạng Binary, file sẽ có tên Class1File.dat

' Thuộc tính này cho phép một class có thể serializable
<Serializable()> Public Class Class1

    ' Tất cả các field trong class này đều có thể serialize, ngoại trừ các thành phần
    ' có đánh dấu là NonSerialized, như z.

    Public x As Integer
    Private y As Integer
    <NonSerialized()> Public z As Integer

    ' Constructor đơn giản để đưa giá trị vào các field.
    Public Sub New(ByVal argx As Integer, ByVal argy As Integer, ByVal argz As
    Integer)
        Me.x = argx
        Me.y = argy
        Me.z = argz
    End Sub

    ' Thuộc tính để xem giá trị của biến private y
    Public ReadOnly Property GetY() As Integer
        Get
            Return y
        End Get
    End Property

End Class
```

Class2.vb

```
Option Strict On

' Class này được đánh dấu là Serializable. Ngoài ra, class này còn sử dụng
' ISerializable, interface này cho phép tùy biến việc chuyển các đối tượng.
' ISerializable đòi hỏi phải cài đặt phương thức GetObjectData, và một constructor
' sẽ được gọi trong Deserialization

Imports System.Runtime.Serialization 'Namespace chứa ISerializable

<Serializable()> Public Class Class2
    Implements ISerializable

    ' Bởi vì trong ví dụ này có sử dụng chuyển đổi tùy biến, nên thuộc tính NonSerialized
    ' sẽ không có tác dụng. Thay vào đó, người viết code cho class quyết định thành
    ' phần nào cho phép serialize, dựa trên phương thức GetObjectData. Chú ý là trong
    ' class này z được đánh dấu là NonSerialized, nhưng nó vẫn được chuyển đổi theo
    ' một cách khác.

    Public x As Integer
    Private y As Integer
    <NonSerialized()> Public z As Integer

    Public Sub New(ByVal argx As Integer, ByVal argy As Integer, ByVal argz As
    Integer)
        Me.x = argx
        Me.y = argy
        Me.z = argz
    End Sub

    ' Constructor đặc biệt được gọi trong quá trình deserialization.
    Public Sub New(ByVal info As SerializationInfo, ByVal context As
    StreamingContext)
        Me.x = info.GetInt32("x")
        Me.y = info.GetInt32("y")
        Me.z = info.GetInt32("z")
        Dim d As Date = info.GetDateTime("TimeStamp")
    End Sub
```

```
    ' Thuộc tính để xem giá trị của biến private y
    Public ReadOnly Property GetY() As Integer
     Get
       Return y
     End Get
    End Property

    ' Phương thức này điều khiển cách các thành phần sẽ được serialize
    ' Có thể truyền vào các giá trị hiện hành (như x và z) hoặc giá trị khác (như y).
    ' Ta cũng có thể serialize các item khác, cho dù chúng không phải là thành
    ' phần của class tương ứng (như 'TimeStamp')
    Public Sub GetObjectData(ByVal info As SerializationInfo, ByVal context As
    StreamingContext) Implements ISerializable.GetObjectData
     With info
       .AddValue("x", Me.x)
       .AddValue("y", -1)
       .AddValue("z", Me.z)
       .AddValue("TimeStamp", Date.Now)
     End With
    End Sub
End Class
```

frmMain.vb

```
Option Strict On
Imports System.IO
Imports System.Runtime.Serialization.Formatters.Binary
'Namespace chứa BinaryFormatter

' Cần có tham chiếu đến System.Runtime.Serialization.Formatters.Soap để dùng
' cho import này

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Các biến này được khởi tạo trong sự kiện Form_Load
    Private strFileName1 As String
    Private strFileName2 As String
    Private strFileName3 As String
```

```
' Code phát sinh form được bỏ qua

Private Sub cmdStandardSerializationSoap_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdStandardSerializationSoap.Click
    ' Thủ tục này tạo mới một instance của Class1, sau đó chuyển nó thành
    ' file Class1File.xml với định dạng SOAP

    ' Tạo đối tượng dùng để serialize
    Dim c As New Class1(CInt(txtX.Text), CInt(txtY.Text), CInt(txtZ.Text))

    ' FileStream dùng để ghi vào file chỉ định trong strFileName1
    Dim fs As New FileStream(strFileName1, FileMode.OpenOrCreate)

    ' Tạo một instance SOAP Formatter
    Dim sf As New SoapFormatter()

    ' Chuyển đối tượng c vào file strFileNam1
    sf.Serialize(fs, c)

    ' Đóng file và giải phóng tài nguyên
    fs.Close()

    ' Cho phép Deserialization
    cmdStandardDeserializationSoap.Enabled = True
    cmdViewClass1.Enabled = True

End Sub

Private Sub cmdStandardDeserializationSoap_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) _
    Handles cmdStandardDeserializationSoap.Click
    ' Thủ tục này chuyển một đối tượng từ file Class1File.xml và gán nó vào một
    ' tham chiếu Class1

    ' Khai báo tham chiếu sẽ trỏ đến đối tượng được chuyển từ file
    Dim c As Class1
    Dim fs As New FileStream(strFileName1, FileMode.Open)
    Dim sf As New SoapFormatter()
```

```
    ' Deserialize c từ strFileName1
    ' Chú ý là đối tượng được Deserialize phải được ép kiểu đúng của nó
    c = CType(sf.Deserialize(fs), Class1)

    fs.Close()

    ' Hiển thị các giá trị của đối tượng lấy từ file
    txtXAfter.Text = CStr(c.x)
    txtYAfter.Text = CStr(c.GetY)
    txtZAfter.Text = CStr(c.z)

    cmdStandardDeserializationSoap.Enabled = False
    cmdViewClass1.Enabled = False

End Sub

Private Sub cmdStandardSerializationBinary_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) _
    Handles cmdStandardSerializationBinary.Click
    ' Thủ tục này tạo một instance mới của Class1, sau đó serialize nó vào
    ' file Class1File.dat với định dạng nhị phân (Binary)

    ' Tạo đối tượng sẽ được serialize
    Dim c As New Class1(CInt(txtX.Text), CInt(txtY.Text), CInt(txtZ.Text))
    ' FileStream dùng để ghi vào file strFileName2
    Dim fs As New FileStream(strFileName2, FileMode.OpenOrCreate)
    ' Tạo một instance của Binary Formatter
    Dim bf As New BinaryFormatter()

    'Serialize c vào file strFileName2
    bf.Serialize(fs, c)

    ' Đóng file và giải phóng tài nguyên
    fs.Close()

    ' Cho phép Deserialization
    cmdStandardDeserializationBinary.Enabled = True
End Sub
```

```
Private Sub cmdStandardDeserializationBinary_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) _
    Handles cmdStandardDeserializationBinary.Click
    ' Thủ tục này deserialize một đối tượng từ file Class1File.dat và gán vào
    ' một tham chiếu Class1

    ' Khai báo tham chiếu sẽ trỏ đến đối tượng được deserialize
    Dim c As Class1
    Dim fs As New FileStream(strFileName2, FileMode.Open)
    Dim bf As New BinaryFormatter()

    'Deserialize c từ strFilename2
    c = CType(bf.Deserialize(fs), Class1)

    fs.Close()

    txtXAfter.Text = CStr(c.x)
    txtYAfter.Text = CStr(c.GetY)
    txtZAfter.Text = CStr(c.z)

    cmdStandardDeserializationBinary.Enabled = False
End Sub

Private Sub cmdCustomSerialization_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles cmdCustomSerialization.Click
    ' Thủ tục này tạo một instance của Class2, sau đó serialize nó vào file
    ' Class2File.xml với định dạng SOAP. Chú ý là mặc dù Class2 có serialization
    ' tùy biến, code thực hiện vẫn giống như serialization chuẩn. Sự khác biệt là
    ' ở trong code của class.
    Dim c As New Class2(CInt(txtX.Text), CInt(txtY.Text), CInt(txtZ.Text))
    Dim fs As New FileStream(strFileName3, FileMode.OpenOrCreate)
    Dim sf As New SoapFormatter()

    sf.Serialize(fs, c)
    fs.Close()
    cmdCustomDeserialization.Enabled = True
    cmdViewClass2.Enabled = True
End Sub
```

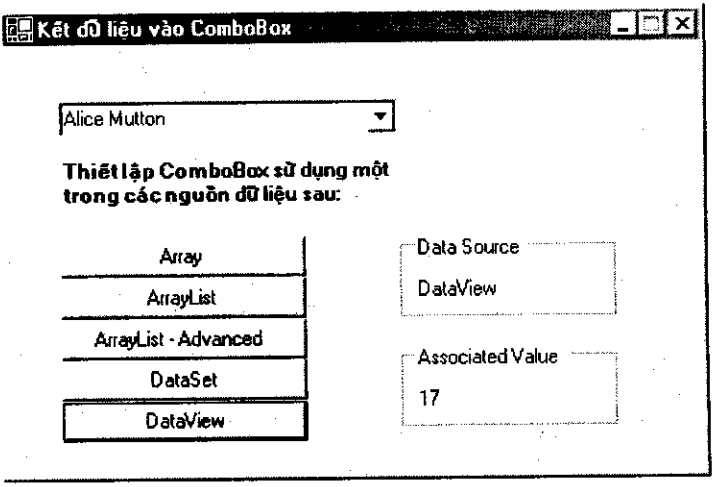
```
Private Sub cmdCustomDeserialization_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles cmdCustomDeserialization.Click
     ' Thủ tục này deserialize một đối tượng từ Class2File.xml và gán vào
     ' một tham chiếu Class2.

    Dim c As Class2
    Dim fs As New FileStream(strFileName3, FileMode.Open)
    Dim sf As New SoapFormatter()

    c = CType(sf.Deserialize(fs), Class2)
    fs.Close()
    txtXAfter.Text = CStr(c.x)
    txtYAfter.Text = CStr(c.GetY)
    txtZAfter.Text = CStr(c.z)

    cmdCustomDeserialization.Enabled = False
    cmdViewClass2.Enabled = False
End Sub

Private Sub cmdViewClass1_Click(ByVal sender As System.Object, _
  ByVal e As System.EventArgs) _
  Handles cmdViewClass1.Click
     ' Đọc nội dung của file và hiển thị trên trên TextBox
    Dim fs As New FileStream(strFileName1, FileMode.Open)
    Dim sr As New StreamReader(fs)
    txtView.Text = sr.ReadToEnd()
    sr.Close()
    fs.Close()
End Sub

Private Sub cmdViewClass2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdViewClass2.Click

    Dim fs As New FileStream(strFileName3, FileMode.Open)
    Dim sr As New StreamReader(fs)
    txtView.Text = sr.ReadToEnd()
    sr.Close()
    fs.Close()
End Sub
```

```
Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    ' Lấy đường dẫn TEMP sử dụng class Path trong namespace System.IO
    Dim strTempPath As String = Path.GetTempPath()
    strFileName1 = strTempPath & "Class1File.xml"
    strFileName2 = strTempPath & "Class1File.dat"
    strFileName3 = strTempPath & "Class2File.xml"

    Me.sbFilePath.Text = "All files will be written to " & strTempPath
End Sub

Private Function IsValidInt32(ByVal Data As String) As Boolean
    Try
        Dim i As Integer
        i = System.Convert.ToInt32(Data)
        Return True
    Catch exp As FormatException
        ' Chuyển kiểu dữ liệu bị lỗi
        Return False

    Catch exp As Exception
        ' Trường hợp lỗi khác
        Return False
    End Try
End Function

Private Sub ValidatingTextIsInt32(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles txtX.Validating,
txtY.Validating, txtZ.Validating
    ' Bảo đảm giá trị nhập vào có thể chuyển sang một số Int32 (Integer)
    Dim txt As TextBox = CType(sender, TextBox)
    If Not IsValidInt32(txt.Text) Then
        Dim strMsg As String
        strMsg = String.Format("The value you entered {0} is not a valid 32-bit integer.
Value will be changed to zero.", txt.Text)
        MsgBox(strMsg, MsgBoxStyle.Exclamation, "Validation Warning")
        txt.Text = "0"
    End If
End Sub
End Class
```

Phần IX - Xử lý dữ liệu

45. Kết dữ liệu vào một ComboBox

Chương trình này cho biết một số cách kết dữ liệu vào một control ComboBox.



Các chức năng chính:

Chương trình biểu diễn cách kết dữ liệu vào một ComboBox từ nhiều nguồn dữ liệu khác nhau. Dữ liệu có thể từ một mảng (Array), ArrayList, DataTable và DataView.

Chú ý:

Chương trình sử dụng database Northwind trong SQL Server hoặc MSDE.

Project: **Ket du lieu vao ComboBox**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected dsProducts As New DataSet()
    Protected dvProducts As DataView
```

```
Protected Const SQL_CONNECTION_STRING As String = _
    "Server=localhost;" & _
    "DataBase=Northwind;" & _
    "Integrated Security=SSPI"

Protected Const MSDE_CONNECTION_STRING As String = _
    "Server=(local)\NetSDK;" & _
    "DataBase=Northwind;" & _
    "Integrated Security=SSPI"

Protected Const CAPTION_TITLE As String = "Bind Data to a ComboBox"
Protected Const DEFAULT_SORT As String = "ProductName ASC"
Protected Const CAPTION_ICON_BUTTON As MsgBoxStyle =
CType(MsgBoxStyle.Information + MsgBoxStyle.OKOnly, MsgBoxStyle)
Protected Const PRODUCT_TABLE_NAME As String = "Products"

' Định nghĩa cấu trúc cho một phân loại, chứa tên phân loại và mã phân loại
Protected Structure Divisions
    Private divName As String
    Private divId As Integer
    Public Sub New(ByVal name As String, ByVal id As Integer)
        divName = name
        divId = id
    End Sub

    Public ReadOnly Property getName() As String
        Get
            Return divName
        End Get
    End Property

    Public ReadOnly Property getId() As Integer
        Get
            Return divId
        End Get
    End Property

End Structure
```

```
    Friend WithEvents cmdDV As System.Windows.Forms.Button
    Friend WithEvents lblAssociated As System.Windows.Forms.Label
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox

' Code phát sinh form được bỏ qua

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        Dim strConnection As String = SQL_CONNECTION_STRING

        ' Hiển thị thông báo là đang kết nối vào SQL Server. Thông báo này chỉ cần
        ' thiết cho lần thực hiện kết nối đầu tiên. Sau khi đã xác định được MSDE hay
        ' SQL Server, thông báo này không cần đến nữa.
        Dim frmStatusMessage As New frmStatus()

        frmStatusMessage.Show("Connecting to SQL Server")

        ' Thực hiện kết nối vào SQL Server cục bộ trước, nếu không thành công
        ' thì chuyển qua MSDE cục bộ
        Dim IsConnecting As Boolean = True
        While IsConnecting

            Try
                ' Class SqlConnection cho phép trao đổi với SQL Server. Constructor
                ' chấp nhận một chuỗi kết nối làm một tham số. Chuỗi kết nối này sử
                ' dụng cơ chế an toàn tích hợp, nghĩa là người lập trình phải có quyền
                ' đăng nhập vào SQL Server, hoặc là một thành viên của nhóm
                ' Administrators trên máy tính cục bộ. Không cần password hoặc
                ' user id trong chuỗi kết nối này.
                Dim northwindConnection As New SqlConnection(strConnection)

                ' SqlDataAdapter được sử dụng để thiết lập một DataSet
                Dim ProductAdapter As New SqlDataAdapter( _
                  "SELECT * " _
                  & "FROM products", _
                  northwindConnection)

                ' Thiết lập DataSet chứa dữ liệu của bảng Products. Bởi vì DataSet có thể
                ' chứa nhiều tập kết quả, tốt nhất là đặt tên cho tập kết quả khi thiết lập
```

```
        ' DataSet. Trong trường hợp này ta đặt tên là "Products"
        ProductAdapter.Fill(dsProducts, PRODUCT_TABLE_NAME)

        ' Tạo DataView; sử dụng Constructor để chỉ định sắp xếp, điều kiện lọc
        dvProducts = New DataView(dsProducts.Tables("products"), "", _
                                  DEFAULT_SORT, DataViewRowState.OriginalRows)

        ' Dữ liệu đã được lấy thành công, thoát vòng lặp
        IsConnecting = False

      ' Xử lý khi thực hiện kết nối xảy ra lỗi
      Catch exc As Exception
        If strConnection = SQL_CONNECTION_STRING Then
          ' Không thể kết nối vào SQL Server, chuyển qua MSDE
          strConnection = MSDE_CONNECTION_STRING
          frmStatusMessage.Show("Connecting to MSDE")
        Else
          ' Không thể kết nối vào SQL Server hoặc MSDE
          frmStatusMessage.Close()
          MsgBox("To run this sample, you must have SQL " & _
          "or MSDE with the Northwind database installed. ", _
           CAPTION_ICON_BUTTON, CAPTION_TITLE)
          ' Thoát chương trình

          End

        End If
      End Try
    End While

    frmStatusMessage.Close()
    GroupBox1.Visible = False
End Sub

Private Sub cmdArray_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdArray.Click
    ' Kết vào một mảng đơn giản chứa các chuỗi là tên các màu
    Dim myColors() As String = {"AQUA", "BLACK", "BLUE", "GREEN", "RED",
"WHITE", "YELLOW"}
    ComboBox1.DataSource = myColors
```

```
    ComboBox1.SelectedIndex = 0
    GroupBox1.Visible = False
    lblSource.Text = "Array"
End Sub

Private Sub cmdArrayList_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdArrayList.Click
    ' Kết vào một arraylist đơn giản chứa tên các hình cơ bản
    Dim myShapes As New ArrayList()
    With myShapes
        .Add("CIRCLE")
        .Add("OCTAGON")
        .Add("RECTANGLE")
        .Add("SQUARE")
        .Add("TRAPEZOID")
        .Add("TRIANGLE")
    End With

    ComboBox1.DataSource = myShapes
    ComboBox1.SelectedIndex = 0
    GroupBox1.Visible = False
    lblSource.Text = "ArrayList"
End Sub

Private Sub cmdArrayListA_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdArrayListA.Click
    ' Kết vào một arraylist chứa các phần tử là cấu trúc đã định nghĩa
    Dim myDivisions As New ArrayList()

    ' Cho các phần tử có cấu trúc vào arraylist
    With myDivisions
        .Add(New Divisions("CENTRAL", 1))
        .Add(New Divisions("EAST", 2))
        .Add(New Divisions("NORTH", 3))
        .Add(New Divisions("SOUTH", 4))
    End With

    ' Kết arraylist vào ComboBox
    With ComboBox1
        .DataSource = myDivisions
```

```
        .DisplayMember = "getName"
        .ValueMember = "getId"
    End With

    ComboBox1.SelectedIndex = 0
    lblAssociated.Text = CType(ComboBox1.SelectedValue, String)
    GroupBox1.Visible = True
    lblSource.Text = "ArrayList-Advanced"
End Sub

Private Sub cmdDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdDS.Click
' Kết bảng Products trong database Northwind đã được đưa vào dataset dsProducts.
    With ComboBox1
        .DataSource = dsProducts.Tables("PRODUCTS")
        .DisplayMember = "ProductName"
        .ValueMember = "ProductID"
    End With
    ComboBox1.SelectedIndex = 0
    lblAssociated.Text = CType(ComboBox1.SelectedValue, String)
    GroupBox1.Visible = True
    lblSource.Text = "DataSet"
End Sub

Private Sub cmdDV_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdDV.Click
    ' Kết vào dataview đã được sắp xếp
    With ComboBox1
        .DataSource = dvProducts
        .DisplayMember = "ProductName"
        .ValueMember = "ProductID"
    End With

    ComboBox1.SelectedIndex = 0
    lblAssociated.Text = CType(ComboBox1.SelectedValue, String)
    GroupBox1.Visible = True
    lblSource.Text = "DataView"

End Sub
```

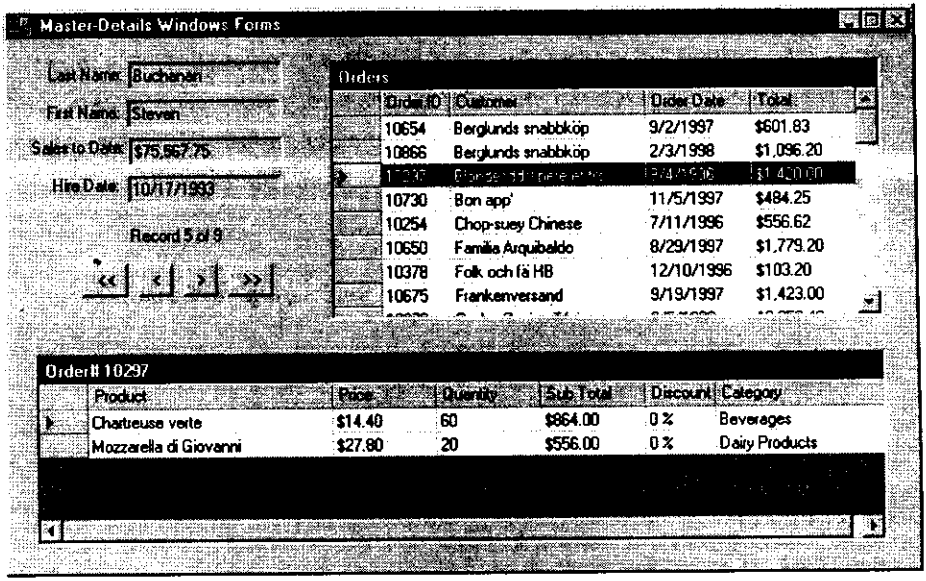
```
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        ' Hiển thị giá trị kết hợp với phần tử được chọn trong ComboBox, nếu giá trị này
        ' tồn tại. Để xác định có tồn tại, kiểm tra groupbox có hiển thị (visible) hay không.
        ' Thuộc tính này được gán là False trong lúc kết dữ liệu nếu không có giá trị kết
        ' hợp, và gán là True nếu có.
        If GroupBox1.Visible = True Then
            lblAssociated.Text = CType(ComboBox1.SelectedValue, String)
        End If
    End Sub

End Class
```

frmStatus.vb

```
Option Strict On
Public Class frmStatus
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Public Overloads Sub Show(ByVal Message As String)
        lblStatus.Text = Message
        Me.Show()
        Application.DoEvents()
    End Sub
End Class
```

46. Tạo ứng dụng Master-Details Windows Forms

Chương trình này chỉ cách tạo ứng dụng Master-Details. Đây là kiểu form kết dữ liệu phổ biến và hữu dụng cho việc duyệt qua các record và xem các cấp độ chi tiết dữ liệu khác nhau có liên quan đến mỗi record.



Các chức năng chính:

- Biểu diễn cách một DataSet có thể chứa bốn DataTable và kết vào các control khác nhau và được đồng bộ với nhau bằng đối tượng BindingContext của Form.
- Hiển thị dữ liệu trong hai control DataGrid khác nhau, một cho bảng **Orders** (bảng chính - Master), và một cho bảng **Order Details** (bảng các chi tiết - Details).
- Hiển thị dữ liệu trong một DataGrid, sử dụng các control phụ của DataGrid để đi xâu vào **Order Details**.
- Duyệt qua các record bằng cách nhấn vào các nút hoặc dùng các phím mũi tên và các phím Home/End.
- Cách áp dụng các định dạng riêng vào các control có kết dữ liệu.

Chú ý:

Chương trình sử dụng database Northwind trong SQL Server hoặc MSDE.

Project: **Master-Details Windows Forms**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient
```

```
Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    ' Khởi tạo các hằng số để kết nối vào database và hiển thị thông báo lỗi
    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL " & _
        "or MSDE with the Northwind database installed. "
    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Protected DidPreviouslyConnect As Boolean = False
    Private dsEmployeeOrders As DataSet
    Private dtEmployee As DataTable
    Private dtOrders As DataTable
    Private dtOrderDetails As DataTable
    Private dtSales As DataTable
    Private dvOrders As DataView
    Private dvOrderDetails As DataView
    Private dvSales As DataView
    Private sda As SqlDataAdapter
    Protected strConn As String = SQL_CONNECTION_STRING

    ' Xử lý nút <<
    Private Sub btnFirst_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnFirst.Click

        ' Di chuyển đến record đầu tiên
        FirstRecord()
    End Sub
```

```
' Xử lý nút >>
Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLast.Click

    ' Di chuyển đến record cuối cùng
    LastRecord()
End Sub

' Xử lý nút >
Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNext.Click

    ' Di chuyển đến record kế tiếp
    NextRecord()
End Sub

' Xử lý nút <
Private Sub btnPrevious_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrevious.Click
    ' Di chuyển lùi về một record
    PreviousRecord()
End Sub

' Xử lý sự kiện PositionChanged của BindingContext. Sự kiện này phát sinh tương
' ứng khi thay đổi Position trong các sự kiện Click của các nút duyệt dữ liệu.
Protected Sub dtEmployee_PositionChanged(ByVal sender As Object, ByVal e As
System.EventArgs)
    BindOrdersGrid()
    BindOrderDetailsGrid()
    ShowTotalSales()
    ShowCurrentRecordNumber()
End Sub

' Xử lý sự kiện KeyDown trên Form. Để sự kiện này phát sinh, thuộc tính
' KeyPreview trên Form phải được đặt là True. Giá trị mặc nhiên là False.
Private Sub frmMain_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles MyBase.KeyDown
    ' Cho phép người sử dụng duyệt qua các record sử dụng các phím mũi tên
    If e.KeyCode = Keys.Right Then NextRecord()
    If e.KeyCode = Keys.Left Then PreviousRecord()
```

```
    If e.KeyCode = Keys.Home Then FirstRecord()
    If e.KeyCode = Keys.End Then LastRecord()
End Sub

' Xử lý Form load
Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    CreateDataSet()
    InitializeBindings()
    BindOrdersGrid()
    BindOrderDetailsGrid()
    ShowCurrentRecordNumber()
End Sub

' Xử lý sự kiện nhấn chuột trên DataGrid. Sự kiện này phát sinh khi khung của
' DataGrid được nhấn chuột – không phải là các ô. Cần xử lý thêm cho sự
' kiện CurrentCellChanged
Private Sub grdOrders_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles grdOrders.Click
    ' Kết dữ liệu cho Order Details dựa trên record được chọn trong grdOrders.
    BindOrderDetailsGrid()
End Sub

' Xử lý sự kiện CurrentCellChanged, phát sinh khi người sử dụng
' chuyển qua một cell khác trên grid.
Private Sub grdOrders_CurrentCellChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles grdOrders.CurrentCellChanged
    ' Làm nổi bật toàn bộ dòng được chọn
    grdOrders.Select(grdOrders.CurrentCell.RowNumber)
    ' Kết dữ liệu cho Order Details dựa trên record được chọn trong grdOrders.
    BindOrderDetailsGrid()
End Sub

' Kết và định dạng lưới Order Details dựa vào record đang được chọn trên lưới Orders
Sub BindOrderDetailsGrid()
    ' Lấy OrderID hiện hành bằng cách dùng CurrentRowIndex của DataGrid
    Dim strCurrentOrderID As String = _
        dvOrders(grdOrders.CurrentRowIndex)("OrderID").ToString
    ' Lọc dữ liệu trên OrderDetails dựa trên OrderID đang được chọn
    dvOrderDetails.RowFilter = "OrderID = " & strCurrentOrderID
```

```
With grdOrderDetails
    .CaptionText = "Order# " & strCurrentOrderID
    .DataSource = dvOrderDetails
End With

' Trước tiên là làm sạch collection TableStyles
grdOrderDetails.TableStyles.Clear()
Dim grdTableStyle1 As New DataGridTableStyle()
With grdTableStyle1
  ' Luôn luôn quy định MappingName, ngay cả với DataView chỉ có một Table.
  ' Nếu không, không có lỗi xảy ra nhưng định dạng không được áp dụng.
    .MappingName = dvOrderDetails.Table.TableName
End With

' Sử dụng column style sẽ làm cho việc tự động phát sinh các cột mất tác dụng.
' Khi các đối tượng Column Style được sử dụng, các cột dữ liệu mà ta muốn hiển
' thị phải có một đối tượng column style kết với nó.
Dim grdColStyle1 As New DataGridTextBoxColumn()
With grdColStyle1
    .MappingName = "ProductName"
    .HeaderText = "Product"
    .Width = 175
End With

Dim grdColStyle2 As New DataGridTextBoxColumn()
With grdColStyle2
    .MappingName = "UnitPrice"
    .HeaderText = "Price"
      ' Định dạng dữ liệu dạng tiền tệ
    .Format = "c"
    .Width = 75
End With

Dim grdColStyle3 As New DataGridTextBoxColumn()
With grdColStyle3
    .MappingName = "Quantity"
    .HeaderText = "Quantity"
    .Width = 75
End With
```

```
    Dim grdColStyle4 As New DataGridTextBoxColumn()
    With grdColStyle4
      .MappingName = "SubTotal"
      .HeaderText = "Sub Total"
      ' Định dạng dữ liệu dạng tiền tệ
      .Format = "c"
      .Width = 75
    End With

    Dim grdColStyle5 As New DataGridTextBoxColumn()
    With grdColStyle5
      .MappingName = "Discount"
      .HeaderText = "Discount"
      ' Định dạng dữ liệu để hiển thị ở dạng phần trăm.
      .Format = "P0"
      .Width = 50
    End With

    Dim grdColStyle6 As New DataGridTextBoxColumn()
    With grdColStyle6
      .MappingName = "CategoryName"
      .HeaderText = "Category"
      .Width = 125
    End With

    ' Áp dụng các Column Style vừa tạo
    grdTableStyle1.GridColumnStyles.AddRange _
      (New DataGridColumnStyle() {grdColStyle1, grdColStyle2, _
        grdColStyle3, grdColStyle4, grdColStyle5, grdColStyle6})

    grdOrderDetails.TableStyles.Add(grdTableStyle1)

End Sub

' Kết dữ liệu và định dạng lưới Orders dựa vào Employee hiện hành
Sub BindOrdersGrid()
  ' Lọc dữ liệu dựa trên giá trị của thuộc tính Tag. Thuộc tính này chứa EmployeeID
  dvOrders.RowFilter = "EmployeeID = " & txtLastName.Tag.ToString
```

```
    With grdOrders
        .CaptionText = "Orders"
        .DataSource = dvOrders
    End With

    grdOrders.TableStyles.Clear()

    Dim grdTableStyle1 As New DataGridTableStyle()
    With grdTableStyle1
        .MappingName = dvOrders.Table.TableName
    End With

    Dim grdColStyle1 As New DataGridTextBoxColumn()
    With grdColStyle1
        .MappingName = "OrderID"
        .HeaderText = "Order ID"
        .Width = 50
    End With

    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
        .MappingName = "CompanyName"
        .HeaderText = "Customer"
        .Width = 140
    End With

    Dim grdColStyle3 As New DataGridTextBoxColumn()
    With grdColStyle3
        .MappingName = "OrderDate"
        ' Định dạng dữ liệu là ngày tháng. Loại bỏ thời gian ra khỏi giá trị
        .Format = "d"
        .HeaderText = "Order Date"
        .Width = 75
    End With

    Dim grdColStyle4 As New DataGridTextBoxColumn()
    With grdColStyle4
        .MappingName = "Total"
        .HeaderText = "Total"
        .Format = "c"
```

```
        .Width = 75
    End With

    grdTableStyle1.GridColumnStyles.AddRange _
        (New DataGridColumnStyle() {grdColStyle1, grdColStyle2, _
            grdColStyle3, grdColStyle4})
    grdOrders.TableStyles.Add(grdTableStyle1)
End Sub

' Tạo DataSet sử dụng trong chương trình này. Nó chứa bốn bảng, bao gồm
' Employee, Các Order của Employee, Sales to Date, và Order Details.
Sub CreateDataSet()
    Dim frmStatusMessage As New frmStatus()
    If Not DidPreviouslyConnect Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim IsConnecting As Boolean = True

    While IsConnecting
        Try
            Dim scnnNW As New SqlConnection(strConn)

            Dim strSQL As String = _
                "SELECT EmployeeID, LastName, FirstName, HireDate " & _
                "FROM Employees"

            ' Đối tượng SqlCommand được sử dụng để thực thi các lệnh SQL
            Dim scmd As New SqlCommand(strSQL, scnnNW)

            ' SqlDataAdapter sử dụng đối tượng SqlCommand để đưa dữ liệu vào DataSet
            sda = New SqlDataAdapter(scmd)

            ' SqlCommandBuilder tự động phát sinh các lệnh SQL cần thiết để
            ' cập nhật database sau này.
            Dim scb As New SqlCommandBuilder(sda)

            ' Các lệnh được phát sinh bởi SqlCommandBuilder được dựa trên tập
            ' CommandText hiện hành của đối tượng SqlCommand. Ta có thể gọi
```

```
' GetUpdateCommand để phát sinh lệnh Update dựa trên giá trị hiện hành
' của thuộc tính CommandText.
scb.GetUpdateCommand()

  ' Tạo mới một DataSet và cho vào DataTable đầu tiên
dsEmployeeOrders = New DataSet()
sda.Fill(dsEmployeeOrders, "Employee")

' Lệnh SQL để lấy các Order của Employee
scmd.CommandText = _
  "SELECT od.OrderID, SUM(CONVERT(money, (od.UnitPrice * " & _
  "  od.Quantity) * (1 - od.Discount) / 100) * 100) " & _
  "  AS Total, o.EmployeeID, o.OrderDate, " & _
  "  c.CompanyName" & vbCrLf & _
  "FROM [Order Details] od " & _
  "  INNER JOIN Orders o " & _
  "  ON od.OrderID = o.OrderID" & vbCrLf & _
  "  INNER JOIN Customers c " & _
  "  ON o.CustomerID = c.CustomerID" & vbCrLf & _
  "GROUP BY od.OrderID, o.EmployeeID, o.OrderDate, c.CompanyName"

  ' Đưa bảng thứ hai vào DataSet
sda.Fill(dsEmployeeOrders, "Orders")

' Lệnh SQL lấy Employee Sales-To-Date
scmd.CommandText = _
  "SELECT e.employeeid, sum(UnitPrice * Quantity) as " & _
  "  'SalesToDate' " & _
  "FROM [order details] od " & _
  "  INNER JOIN orders o " & _
  "  ON o.orderid = od.orderid " & _
  "  INNER JOIN employees e " & _
  "  ON e.employeeid = o.employeeid" & vbCrLf & _
  "GROUP BY e.employeeid"

  ' Đưa bảng thứ ba vào DataSet
sda.Fill(dsEmployeeOrders, "Sales")
```

```
        ' Lệnh SQL lấy Order Details
        scmd.CommandText = _
          "SELECT od.OrderID, od.UnitPrice, od.Quantity, od.Discount, " & _
          "     p.ProductName, c.CategoryName, " & _
          "     (od.UnitPrice * od.Quantity) As SubTotal " & _
          "FROM [order details] od " & _
          "  INNER JOIN Products p ON od.ProductID = p.ProductID " & _
          "  INNER JOIN Categories c ON c.CategoryID = p.CategoryID " & _
          "ORDER BY od.OrderID"

        ' Đưa bảng thứ tư vào DataSet
        sda.Fill(dsEmployeeOrders, "OrderDetails")

        ' Gán các biến cho các DataTable để sử dụng sau này
        dtEmployee = dsEmployeeOrders.Tables(0)
        dtOrders = dsEmployeeOrders.Tables(1)
        dtSales = dsEmployeeOrders.Tables(2)
        dtOrderDetails = dsEmployeeOrders.Tables(3)

        ' Quy định các DataView cho các DataGird và TextBox SalesToDate
        dvOrders = dtOrders.DefaultView
        dvOrderDetails = dtOrderDetails.DefaultView
        dvSales = dtSales.DefaultView

        ' Để thấy được kiểu giao diện Master-Details khác, cả dữ liệu Master và
        ' dữ liệu Details chứa trong cùng một DataGrid, bỏ ghi chú các dòng code
        ' bên dưới, các dòng code này quy định một mối quan hệ bảng cha-con.

          ' dsEmployeeOrders.Relations.Add("Order_OrderDetails", _
          '    dtOrders.Columns("OrderID"), dtOrderDetails.Columns("OrderID"))

        ' Dữ liệu đã được lấy thành công, thoát vòng lặp và đóng form Status
        IsConnecting = False
        DidPreviouslyConnect = True
        frmStatusMessage.Close()

      Catch expSql As SqlException
        MsgBox(expSql.Message, MsgBoxStyle.Critical, Me.Text)
        Exit Sub
```

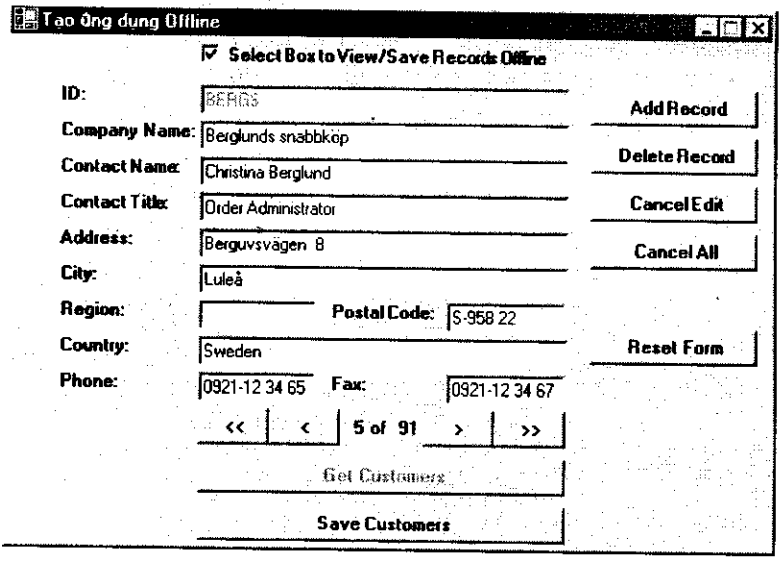
```
        Catch exp As Exception
            If strConn = SQL_CONNECTION_STRING Then
                strConn = MSDE_CONNECTION_STRING
                frmStatusMessage.Show("Connecting to MSDE")
            Else
                frmStatusMessage.Close()
                MsgBox(CONNECTION_ERROR_MSG, MsgBoxStyle.Critical, Me.Text)
                End
            End If
        End Try
        frmStatusMessage.Close()
    End While
End Sub

' Xử lý định dạng cho TextBox Hire Date
Protected Sub DateToString(ByVal sender As Object, ByVal e As
ConvertEventArgs)
    ' Có thể sử dụng một trong hai cách sau để chuyển định dạng ngày:
    ' e.Value = CType(e.Value, DateTime).ToString("d")
    e.Value = CType(e.Value, DateTime).ToShortDateString
End Sub

' Di chuyển Position của BindingContext về record đầu tiên.
Public Sub FirstRecord()
    ' Vị trí của binding context điều khiển Position (vị trí) của record hiện hành,
    ' Record đầu tiên là record 0 (không phải 1)
    Me.BindingContext(dtEmployee).Position = 0
End Sub

' Quy định các kết dữ liệu cho các control
Private Sub InitializeBindings()
    txtLastName.DataBindings.Add("Text", dtEmployee, "LastName")
    ' Sử dụng thuộc tính Tag ta có thể kết một giá trị ẩn cần khi duyệt qua các
    ' record. Trong chương trình này, các số EmployeeID tương ứng với các số
    ' dòng trong DataTable, do đó chỉ cần dùng thuộc tính Position như trong
    ' thủ tục ShowCurrentRecordNumber.
    txtLastName.DataBindings.Add("Tag", dtEmployee, "EmployeeID")
    txtFirstName.DataBindings.Add("Text", dtEmployee, "FirstName")
```

```
    ' Đối tượng Binding kết một thuộc tính của một control vào một trường của
    ' data source. Sau khi tạo đối tượng Binding, cho nó vào đối tượng
    ' ControlBindingCollection thấy được bởi tất cả các control Windows Forms
    ' thông qua thuộc tính DataBindings.
    Dim dbnHireDate As New Binding("Text", dtEmployee, "HireDate")
    txtHireDate.DataBindings.Add(dbnHireDate)
     ' Nếu có sử dụng các thủ tục phân tích/định dạng, thì thêm chúng
    AddHandler dbnHireDate.Format, AddressOf DateToString
    AddHandler dbnHireDate.Parse, AddressOf StringToDate

    Dim dbnSalesToDate As New Binding("Text", dtSales, "SalesToDate")
    txtSalesToDate.DataBindings.Add(dbnSalesToDate)
    AddHandler dbnSalesToDate.Format, AddressOf MoneyToString
    AddHandler dbnSalesToDate.Parse, AddressOf StringToMoney
    AddHandler Me.BindingContext(dtEmployee).PositionChanged, _
      AddressOf dtEmployee_PositionChanged
End Sub

' Di chuyển BindingContext Position đến record cuối cùng
Public Sub LastRecord()
  Me.BindingContext(dtEmployee).Position = dtEmployee.Rows.Count - 1
End Sub

' Xử lý Format cho TextBox SalesToDate
Protected Sub MoneyToString(ByVal sender As Object, ByVal e As
ConvertEventArgs)
  e.Value = CType(e.Value, Decimal).ToString("c")
End Sub

' Di chuyển BindingContext Position đến record kết tiếp
Public Sub NextRecord()
  Me.BindingContext(dtEmployee).Position += 1
End Sub

' Di chuyển BindingContext Position lùi về một record
Public Sub PreviousRecord()
  Me.BindingContext(dtEmployee).Position -= 1
End Sub
```

```
    ' Hiển thị chỉ số record hiện hành và tổng số record
    Protected Sub ShowCurrentRecordNumber()
        lblRecordNumber.Text = "Record " & _
            Me.BindingContext(dtEmployee).Position + 1 & " of " & _
                dtEmployee.Rows.Count
    End Sub

    ' Cập nhật giá trị của Sales To Date cho mỗi employee khi người sử dụng
    ' duyệt qua các record.
    Protected Sub ShowTotalSales()
        ' Lọc dvSales dựa vào giá trị của thuộc tính Tag
        dvSales.RowFilter = "EmployeeID = " & txtLastName.Tag.ToString
    End Sub

    ' Xử lý sự kiện Parse cho TextBox Hire Date
    Protected Sub StringToDate(ByVal sender As Object, ByVal e As
    ConvertEventArgs)
        Try
            e.Value = CDate(e.Value)
        Catch exp As Exception
            MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
        End Try
    End Sub

    ' Xử lý sự kiện Parse cho TextBox Sales To Date
    Protected Sub StringToMoney(ByVal sender As Object, ByVal e As
    ConvertEventArgs)
        Try
            e.Value = CType(e.Value, Double)
        Catch exp As Exception
            MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
        End Try
    End Sub
End Class
```

47. Tạo một ứng dụng Offline

Chương trình này biểu diễn cách sử dụng các phương thức của đối tượng DataSet để cho phép một ứng dụng làm việc với cơ sở dữ liệu ở chế độ có kết nối (online) hoặc không kết nối thường trực (offline).



Các chức năng chính:

- Chuyển dữ liệu và thông tin trạng thái các record của một DataSet vào một file XML.
- Đọc dữ liệu và thông tin trạng thái các record từ một file XML vào một DataSet.
- Cập nhật các thay đổi đã thực hiện trên DataSet lên cơ sở dữ liệu trên SQL Server.
- SqlCommandBuilder được sử dụng để tạo các lệnh Insert, Update và Delete cho SQL Adapter.

Project: **Tao ung dung Offline**

CustomersDataSet.xsd

```
<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema id="CustomersDataSet" xmlns=""
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
  microsoft-com:xml-msdata">
  <xsd:element name="Customers">
```

```
      <xsd:complexType>
        <xsd:all>
          <xsd:element name="CustomerID" type="xsd:string" />
          <xsd:element name="CompanyName" type="xsd:string" />
          <xsd:element name="ContactName" minOccurs="0" type="xsd:string" />
          <xsd:element name="ContactTitle" minOccurs="0" type="xsd:string" />
          <xsd:element name="Address" minOccurs="0" type="xsd:string" />
          <xsd:element name="City" minOccurs="0" type="xsd:string" />
          <xsd:element name="Region" minOccurs="0" type="xsd:string" />
          <xsd:element name="PostalCode" minOccurs="0" type="xsd:string" />
          <xsd:element name="Country" minOccurs="0" type="xsd:string" />
          <xsd:element name="Phone" minOccurs="0" type="xsd:string" />
          <xsd:element name="Fax" minOccurs="0" type="xsd:string" />
        </xsd:all>
      </xsd:complexType>
      <xsd:key name="Constraint1" msdata:PrimaryKey="true">
        <xsd:selector xpath="." />
        <xsd:field xpath="CustomerID" />
      </xsd:key>
    </xsd:element>
    <xsd:element name="CustomersDataSet" msdata:IsDataSet="true">
      <xsd:complexType>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element ref="Customers" />
        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' dsCustomers sẽ chứa tất cả customer ở USA
    Protected dsCustomers As New CustomersDataSet()
```

```
    Protected dsCustomerChanges As New CustomersDataSet()
    Protected daCustomers As SqlDataAdapter

    ' Sử dụng để tham chiếu đến bảng chứa thông tin customer
    Protected Const CUSTOMER_TABLE_NAME As String = "Customers"

    Protected Const SQL_CONNECTION_STRING As String = _
      "Server=localhost;" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
      "Server=(local)\NetSDK;" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI"

    Dim DidPreviouslyConnect As Boolean = False
    Dim IsAdding As Boolean = False

' Code phát sinh form được bỏ qua

    Private Sub bnAdd_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles bnAdd.Click
      Try
        ' Kết thúc các soạn thảo hiện hành
        BindingContext(dsCustomers, "Customers").EndCurrentEdit()
        BindingContext(dsCustomers, "Customers").AddNew()
      Catch exc As Exception
        MsgBox(exc.Message, MsgBoxStyle.Critical, Me.Text)
      End Try

      ' Thêm một record mới. Thủ tục UpdateViewState sẽ enable
      ' các nút dựa trên giá trị này
      IsAdding = True
      UpdateViewState()

    End Sub
```

```
Private Sub bnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnCancel.Click
    ' Trả lại các giá trị gốc
    BindingContext(dsCustomers, "Customers").CancelCurrentEdit()
    IsAdding = False
    UpdateViewState()
End Sub

Private Sub bnCancelAll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnCancelAll.Click
    ' Trả lại các giá trị gốc cho DataSet
    dsCustomers.Customers.RejectChanges()
End Sub

Private Sub bnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnDelete.Click
    ' Sử dụng phương thức Delete thay cho phương thức RemoveAt khi DataSet
    ' sẽ cập nhật dữ liệu nguồn
    dsCustomers.Customers(BindingContext(dsCustomers, _
                          "Customers").Position).Delete()
    UpdateViewState()
End Sub

Private Sub bnFirst_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnFirst.Click
    BindingContext(dsCustomers, "Customers").Position = 0
    UpdateViewState()
End Sub

Private Sub bnGetCustomers_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnGetCustomers.Click
    ' Dữ liệu có thể lấy từ một file XML hoặc từ Server
    If chkWorkOffline.Checked Then
        dsCustomers = GetCustomersFromXML()
    Else
        dsCustomers = GetCustomersFromServer()
    End If
    BindData()
    UpdateViewState()
End Sub
```

```
Private Sub bnLast_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnLast.Click
    BindingContext(dsCustomers, "Customers").Position = _
                        BindingContext(dsCustomers, "Customers").Count
    UpdateViewState()
End Sub

Private Sub bnNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnNext.Click
    BindingContext(dsCustomers, "Customers").Position += 1
    UpdateViewState()
End Sub

Private Sub bnPrev_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnPrev.Click
    BindingContext(dsCustomers, "Customers").Position -= 1
    UpdateViewState()
End Sub

Private Sub bnReset_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnReset.Click
    ' Xóa dữ liệu trong bảng Customers của DataSet
    dsCustomers.Customers.Clear()

    ' Cập nhật trạng thái các control để cho phép lấy dữ liệu
    UpdateViewState()
End Sub

Private Sub bnSaveCustomers_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles bnSaveCustomers.Click
    ' Đưa các thay đổi vào DataSet
    BindingContext(dsCustomers, "Customers").EndCurrentEdit()
    ' Có thể thực hiện cập nhật vào một file XML hoặc lên Server
    If chkWorkOffline.Checked Then

        ' Gửi DataSet để được ghi vào một file XML
        SaveCustomersInXML(dsCustomers)

    Else
```

```
        ' Lấy các record được thay đổi trong DataSet dsCustomers
        dsCustomerChanges = CType(dsCustomers.GetChanges, _
                                  CustomersDataSet)

        ' Cập nhật dữ liệu nguồn nếu đã có thay đổi và trộn các thay đổi
        ' vào DataSet hiện hành
        If Not (dsCustomerChanges Is Nothing) Then
          dsCustomerChanges = SaveCustomersOnServer(dsCustomerChanges)

          ' Trộn các thay đổi vào DataSet nếu có thay đổi
          If Not (dsCustomerChanges Is Nothing) Then
            dsCustomers.Merge(dsCustomerChanges)
          End If
        End If
      End If
    End Sub

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
      ' Sao chép bảng Customers vào bảng CustomersHT để tránh lỗi khóa ngoại
      SetupTable()
      UpdateViewState()
    End Sub

    Sub BindData()

      ' Xóa tất cả các kết dữ liệu
      tbCustomerID.DataBindings.Clear()
      tbCompanyName.DataBindings.Clear()
      tbContactName.DataBindings.Clear()
      tbContactTitle.DataBindings.Clear()
      tbAddress.DataBindings.Clear()
      tbCity.DataBindings.Clear()
      tbRegion.DataBindings.Clear()
      tbPostalCode.DataBindings.Clear()
      tbCountry.DataBindings.Clear()
      tbPhone.DataBindings.Clear()
      tbFax.DataBindings.Clear()
```

```
    ' Kết thuộc tính của mỗi control vào một field của dsCustomers để dữ liệu hiển
    ' thị trên control đó. DataBinding sẽ quản lý việc hiển thị dữ liệu khi người sử
    ' dụng duyệt, thêm, xóa, sửa dữ liệu trong DataSet.
    tbCustomerID.DataBindings.Add("Text", dsCustomers, _
                                  "Customers.CustomerID")
    tbCompanyName.DataBindings.Add("Text", dsCustomers, _
                                   "Customers.CompanyName")
    tbContactName.DataBindings.Add("Text", dsCustomers, _
                                   "Customers.ContactName")
    tbContactTitle.DataBindings.Add("Text", dsCustomers, _
                                    "Customers.ContactTitle")
    tbAddress.DataBindings.Add("Text", dsCustomers, _
                               "Customers.Address")
    tbCity.DataBindings.Add("Text", dsCustomers, "Customers.City")
    tbRegion.DataBindings.Add("Text", dsCustomers, "Customers.Region")
    tbPostalCode.DataBindings.Add("Text", dsCustomers, "Customers.PostalCode")
    tbCountry.DataBindings.Add("Text", dsCustomers, "Customers.Country")
    tbPhone.DataBindings.Add("Text", dsCustomers, "Customers.Phone")
    tbFax.DataBindings.Add("Text", dsCustomers, "Customers.Fax")
End Sub

Function GetCustomersFromServer() As CustomersDataSet
    ' Nhận các record từ bảng Customers IT trong database Northwind
    Static ConnectionString As String = SQL_CONNECTION_STRING

    Dim ds As New CustomersDataSet()
    Dim frmStatusMessage As New frmStatus()
    If Not DidPreviouslyConnect Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    ' Thực hiện kết nối vào SQL Server hoặc MSDE cục bộ
    Dim IsConnecting As Boolean = True

    While IsConnecting

        Try
            Dim cnNorthwind As New SqlConnection(ConnectionString)
```

```
            ' SqlDataAdapter được sử dụng để thiết lập một DataSet
            daCustomers = New SqlDataAdapter( _
              "SELECT CustomerID, CompanyName, ContactName, ContactTitle, " _
               & "Address, City, Region, PostalCode, Country, Phone, Fax " _
              & "FROM CustomersHT", _
              cnNorthwind)

            ' Thiết lập DataSet với thông tin từ bảng Customers.
            daCustomers.Fill(ds, "Customers")

            ' Dữ liệu đã được lấy, thoát khỏi vòng lặp
            IsConnecting = False
            DidPreviouslyConnect = True

        Catch exc As Exception
            If ConnectionString = SQL_CONNECTION_STRING Then
                ' Không thể kết nối vào SQL Server, chuyển qua MSDE
                ConnectionString = MSDE_CONNECTION_STRING
                frmStatusMessage.Show("Connecting to MSDE")
            Else
                ' Không thể thực hiện kết nối
                frmStatusMessage.Close()
                MessageBox.Show("To run this sample, you must have SQL " & _
                  "or MSDE with the Northwind database installed.", _
                  "SQL Server/MSDE not found", _
                  MessageBoxButtons.OK, _
                  MessageBoxIcon.Exclamation)
                End
            End If
        End Try
    End While

    frmStatusMessage.Close()

    daCustomers.Dispose()

    Return ds
End Function
```

```
Function GetCustomersFromXML() As CustomersDataSet
    Dim ds As New CustomersDataSet()

    ' Đọc một file XML vào DataSet. Sử dụng cơ chế DiffGram để thông tin RowState
    ' có thể được sử dụng để cập nhật dữ liệu nguồn khi gọi phương thức Update
    ' của SQL Data Adapter. Cơ chế DiffGram cũng phải được sử dụng khi gọi
    ' phương thức WriteXML của DataSet.
    Try
        ds.ReadXml("../dsCustomers.xml", XmlReadMode.DiffGram)
        ds.Tables(0).TableName = "Customers"
    Catch FileNotFound As FileNotFoundException
        ' Dữ liệu XML phải được ghi vào trước khi có thể đọc nó.
        MsgBox("No XML file available.  View data online and save to offline.",
MsgBoxStyle.Critical)
    End Try

    Return ds
End Function

Function SaveCustomersOnServer(ByVal dsCustomerChanges As
CustomersDataSet) As CustomersDataSet
    ' Cập nhật bảng CustomersHT với các thay đổi từ DataSet
    Static ConnectionString As String = SQL_CONNECTION_STRING

    Dim frmStatusMessage As New frmStatus()
    If Not DidPreviouslyConnect Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim IsConnecting As Boolean = True
    While IsConnecting

        Try
            Dim cnNorthwind As New SqlConnection(ConnectionString)

            daCustomers = New SqlDataAdapter( _
                "SELECT CustomerID, CompanyName, ContactName, ContactTitle, " _
                 & "Address, City, Region, PostalCode, Country, Phone, Fax " _
                & "FROM CustomersHT", _
                cnNorthwind)
```

```
        ' SqlCommandBuilder xem xét câu lệnh Select và xây dựng lệnh SQL để
        ' thực hiện insert, update và delete.
        Dim oCommandBuilder As New SqlCommandBuilder(daCustomers)

        daCustomers.InsertCommand = oCommandBuilder.GetInsertCommand
        daCustomers.UpdateCommand = oCommandBuilder.GetUpdateCommand
        daCustomers.DeleteCommand = oCommandBuilder.GetDeleteCommand

        ' Cập nhật dữ liệu nguồn với các dòng có thay đổi và trả về
        ' các dòng có thay đổi
        If Not (dsCustomerChanges Is Nothing) Then
            ' Phải chỉ cụ thể bảng nào để cập nhật
            daCustomers.Update(dsCustomerChanges.Tables(0))
            Return dsCustomerChanges
        Else
            Return Nothing
        End If

        ' Dữ liệu đã được cập nhật thành công, thoát khỏi vòng lặp
        IsConnecting = False
        DidPreviouslyConnect = True

    Catch sqlExc As SqlException
        Select Case sqlExc.Number
            Case 17
                ' SQL Server không tồn tại
                If ConnectionString = SQL_CONNECTION_STRING Then
                    ConnectionString = MSDE_CONNECTION_STRING
                    frmStatusMessage.Show("Connecting to MSDE")
                Else
                    frmStatusMessage.Close()
                    MessageBox.Show("To run this sample, you must have SQL " & _
                        "or MSDE with the Northwind database installed.", _
                        "SQL Server/MSDE not found", _
                        MessageBoxButtons.OK, _
                        MessageBoxIcon.Exclamation)
                    End
                End If
```

```
            Case 2627
                ' Lỗi khóa chính
                MsgBox("2 Records were added with the same primary key." & _
                       " Changes canceled.", MsgBoxStyle.Critical, Me.Text)
                dsCustomers.RejectChanges()
                Exit Function
            End Select
        Catch exc As Exception
            MsgBox(exc.Message, MsgBoxStyle.Critical, Me.Text)
        End Try
    End While

    frmStatusMessage.Close()
End Function

Sub SetupTable()
    ' Tạo bảng CustomersHT và insert các record từ bảng Customers
    Static ConnectionString As String = SQL_CONNECTION_STRING

    Dim ds As New CustomersDataSet()

    Dim frmStatusMessage As New frmStatus()
    If Not DidPreviouslyConnect Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim IsConnecting As Boolean = True
    While IsConnecting

        Try
            Dim cnNorthwind As New SqlConnection(ConnectionString)
            cnNorthwind.Open()

            ' Bảng sử dụng cho chương trình này phải được tạo
            ' Đối tượng SqlCommand dùng để thực hiện các lệnh SQL
            Dim cmInitSQL As New SqlCommand()

            ' Kết command vào kết nối database
            cmInitSQL.Connection = cnNorthwind
```

```
        ' Chọn kiểu lệnh là Text
        cmInitSQL.CommandType = CommandType.Text
        ' Kiểm tra bảng CustomersHT có tồn tại chưa
        cmInitSQL.CommandText = "SELECT count(*) FROM dbo.sysobjects
WHERE id = object_id('CustomersHT')"

        ' Thực thi lệnh
        Dim recordCount As Integer = CType(cmInitSQL.ExecuteScalar(), Integer)
        If recordCount > 0 Then
            ' Nếu bảng đã tồn tại thì thoát, nếu chưa có thì tạo nó
            cmInitSQL.Dispose()
            cnNorthwind.Close()

            IsConnecting = False
            DidPreviouslyConnect = True
            Return
        End If

        ' Tạo bảng CustomersHT
        cmInitSQL.CommandText = "CREATE TABLE [CustomersHT] ( " & _
                    "[CustomerID] [nchar] (5) NOT NULL PRIMARY KEY ," & _
                    "[CompanyName] [nvarchar] (40) NOT NULL , " & _
                    "[ContactName] [nvarchar] (30) NULL , " & _
                    "[ContactTitle] [nvarchar] (30) NULL , " & _
                    "[Address] [nvarchar] (60)  NULL , " & _
                    "[City] [nvarchar] (15) NULL , " & _
                    "[Region] [nvarchar] (15) NULL , " & _
                    "[PostalCode] [nvarchar] (10) NULL , " & _
                    "[Country] [nvarchar] (15) NULL , " & _
                    "[Phone] [nvarchar] (24) NULL , " & _
                    "[Fax] [nvarchar] (24) NULL )"

        ' Thực hiện câu lệnh trên
        cmInitSQL.ExecuteNonQuery()
```

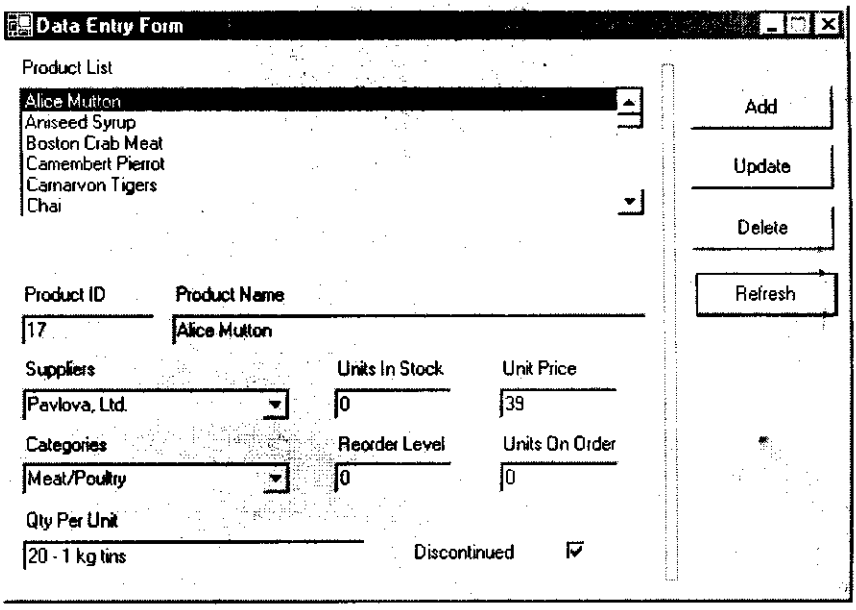
```
            ' Insert dữ liệu từ bảng Customers vào bảng CustomersHT mới
            cmInitSQL.CommandText = "INSERT INTO CustomersHT " & _
                                    "(CustomerID, CompanyName, ContactName," & _
                                    "ContactTitle, Address, City, Region," & _
                                    "PostalCode, Country, Phone, Fax) " & _
                                    "SELECT CustomerID, CompanyName, ContactName," & _
                                    "ContactTitle, Address, City, Region," & _
                                    "PostalCode, Country, Phone, Fax FROM Customers"

            ' Thực hiện câu lệnh
            cmInitSQL.ExecuteNonQuery()

            cmInitSQL.Dispose()
            cnNorthwind.Close()

            ' Dữ liệu đã cập nhật thành công, thoát vòng lặp
            IsConnecting = False
            DidPreviouslyConnect = True

        Catch e As SqlException
            If ConnectionString = SQL_CONNECTION_STRING Then
                ConnectionString = MSDE_CONNECTION_STRING
                frmStatusMessage.Show("Connecting to MSDE")
            Else
                frmStatusMessage.Close()
                MsgBox("You must have SQL Server ot MSDE with " & _
                    "the Northwind database installed", MsgBoxStyle.Critical, _
                    "SQL Server/MSDE not found")
                ' Thoát chương trình nếu không kết nối được
                End

            End If
        Catch e As Exception
            MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
        End Try
    End While
End Sub
```

```
Sub UpdateViewState()
    ' Hiển thị vị trí của record để người sử dụng biết đang làm việc trên record
    ' nào và có bao nhiêu record.
    lblPosition.Text = (((BindingContext(dsCustomers, "Customers").Position +
1).ToString + " of ")  + BindingContext(dsCustomers, _
                                        "Customers").Count.ToString)

    ' Nếu DataSet có dữ liệu thì disable các nút Read và enable các nút Update
    If dsCustomers.Customers.Count <> 0 Then
        bnGetCustomers.Enabled = False
        bnAdd.Enabled = True
        bnDelete.Enabled = True
        bnCancel.Enabled = True
        bnCancelAll.Enabled = True
        bnSaveCustomers.Enabled = True
        bnReset.Enabled = True
    Else
        bnGetCustomers.Enabled = True

        bnAdd.Enabled = False
        bnDelete.Enabled = False

        bnCancel.Enabled = False
        bnCancelAll.Enabled = False
        bnSaveCustomers.Enabled = False
        bnReset.Enabled = False
    End If

    ' Cho phép người sử dụng thay đổi CustomerID khi thêm một record.
    ' Không cho phép xóa trong khi thêm một record.
    If IsAdding Then
        tbCustomerID.Enabled = True
        bnDelete.Enabled = False
    Else
        tbCustomerID.Enabled = False
    End If
    IsAdding = False
End Sub
```

```
Function SaveCustomersInXML(ByVal dsCustomers As CustomersDataSet) As
CustomersDataSet
    ' Ghi DataSet vào file XML cục bộ. Sử dụng cơ chế DiffGram để lưu thông tin
    ' RowState hiện hành của tất cả dữ liệu để SQL Data Adapter có thể thực
    ' hiện các thao tác thích hợp khi gọi phương thức Update. DiffGram cũng
    ' phải được sử dụng khi đọc dữ liệu từ file XML.
    dsCustomers.WriteXml("../dsCustomers.xml", XmlWriteMode.DiffGram)
    Return dsCustomers
End Function
End Class
```

48. Tạo một Form nhập liệu

Chương trình này biểu diễn cách tạo một form nhập dữ liệu để thêm, cập nhật và xóa các record trong một bảng.



Các chức năng chính:

- ADO .NET được sử dụng để thêm (Insert), xóa (delete), và sửa (update) các record trong bảng Products của database Northwind. Một class ListItem được tạo để lưu một giá trị và một khóa chính kết với giá trị đó để sử dụng trong các ComboBox trên form.

Project: **Tao mot form nhap lieu**

ListItem.vb

```
Option Strict On
Option Explicit On

' Class này được sử dụng để lưu nội dung và khóa chính của một record mà ta
' muốn hiển thị trong ListBox

Public Class ListItem
    Private mValue As String   ' Lưu nội dung của record
    Private mID As Integer     ' Lưu khóa chính của record

    Public Sub New(ByVal strValue As String, ByVal intID As Integer)
        mValue = strValue
        mID = intID
    End Sub

    Public Sub New()
        mValue = ""
        mID = 0
    End Sub

    Property ID() As Integer
        Get
            Return mID
        End Get
        Set(ByVal Value As Integer)
            mID = Value
        End Set
    End Property

    Property Value() As String
        Get
            Return mValue
        End Get
        Set(ByVal Value As String)
            mValue = Value
        End Set
    End Property
```

```
    ' Khi một ListBox hiển thị một phần tử trong tập hợp các phần tử, nó gọi phương thức
    ' ToString để lấy giá trị để hiển thị, do đó ta cần override phương thức này để
    ' class này trả về mValue.

    Public Overrides Function ToString() As String
        Return mValue
    End Function

End Class
```

frmMain.vb

```
Option Strict On
Option Explicit On
Imports System.Data
Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI;Connect Timeout=5"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\VSDotNet;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI;Connect Timeout=5"

    Private ConnectionString As String = SQL_CONNECTION_STRING
    Private HasConnected As Boolean = False
    Private Mode As String = "Update"

    ' Code phát sinh form được bỏ qua

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnAdd.Click
        ClearForm()
        Mode = "Add"
```

```
    btnDelete.Enabled = False
    btnAdd.Enabled = False
End Sub

Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
    DeleteProduct()
End Sub

Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdate.Click
    ' Kiểm tra chế độ để quyết định hành động tương ứng
    If Mode = "Add" Then
        AddProduct()
    Else
        UpdateProduct()
    End If

End Sub

Private Sub frmDataEntry_Load(ByVal sender As System.Object, _
                    ByVal e As System.EventArgs) _
                    Handles MyBase.Load
    ' Thiết lập tất cả danh sách và ComboBox được sử dụng trên form
    PopulateCategoryCombo()
    PopulateSupplierCombo()
    PopulateProductList()

End Sub

Private Sub lstProducts_SelectedIndexChanged(ByVal sender As System.Object, _
                            ByVal e As System.EventArgs) _
                            Handles lstProducts.SelectedIndexChanged
    PopulateForm()
    btnDelete.Enabled = True
    btnAdd.Enabled = True
    Mode = "Update"

End Sub
```

```
Private Sub AddProduct()
    ' Thủ tục này thêm một record mới vào database khi người sử dụng nhấn vào
    ' nút Update và đang ở chế độ ADD (mode="ADD")

    ' Kiểm tra các giá trị trên form
    If Not IsValidForm() Then
        Exit Sub
    End If

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer
    Dim ProductName As String = txtProductName.Text

    Try
        ' Xây dựng câu lệnh Insert để thêm một sản phẩm mới vào bảng Products
        strSQL = "INSERT Products VALUES (" & _
        PrepareStr(ProductName) & "," & _
        CType(cbSuppliers.Items(cbSuppliers.SelectedIndex), ListItem).ID & "," & _
        CType(cbCategories.Items(cbCategories.SelectedIndex), ListItem).ID & "," & _
        PrepareStr(txtQtyPerUnit.Text) & "," & _
        txtUnitPrice.Text & "," & _
        txtUnitsInStock.Text & "," & _
        txtUnitsOnOrder.Text & "," & _
            txtReorderLevel.Text & "," & _
            CType(IIf(chkDiscontinued.Checked, "1", "0"), String) & ")"


        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        cmSQL.ExecuteNonQuery()

        ' Đóng và giải phóng các đối tượng
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()
```

```
        ' Cập nhật lại danh sách Product List
        PopulateProductList()
        FindProductByName(lstProducts, ProductName)

    Catch Exp As SqlException
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "SQL Error")

    Catch Exp As Exception
        MsgBox(Exp.Message, MsgBoxStyle.Critical, "General Error")
    End Try
End Sub

Private Sub ClearForm()

    ' Xóa nội dung trên form
    txtProductID.Text = ""
    txtProductName.Text = ""
    txtQtyPerUnit.Text = ""
    txtUnitPrice.Text = "0.00"
    txtUnitsInStock.Text = "0"
    txtUnitsOnOrder.Text = "0"
    txtReorderLevel.Text = "0"
    chkDiscontinued.Checked = False
    cbSuppliers.SelectedIndex = -1
    cbCategories.SelectedIndex = -1

End Sub

Private Sub DeleteProduct()
    ' Thủ tục này sử dụng để xóa một record product ra khỏi database khi người
    ' sử dụng nhấn vào nút Delete

    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer
```

```
    Try
        ' Xây dựng câu lệnh Delete để xóa sản phẩm hiện hành ra khỏi bảng
        strSQL = "DELETE FROM Products " & _
            "WHERE ProductID = " & CInt(txtProductID.Text)

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        intRowsAffected = cmSQL.ExecuteNonQuery()

        If intRowsAffected <> 1 Then
            MsgBox("Delete Failed. Product ID " & txtProductID.Text & _
                " not found.", MsgBoxStyle.Critical, "Delete")
        End If

        ' Đóng và giải phóng các đối tượng
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()

        PopulateProductList()

    Catch e As SqlException
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL Error")

    Catch e As Exception
        MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
    End Try
End Sub

Private Sub FindItemByID(ByVal cbCombo As ComboBox, ByVal strID As String)
    ' Thủ tục này sử dụng để tìm một Item trong một ComboBox và chọn nó
    Dim bFound As Boolean
    Dim ListItem As ListItem
    Dim ItemCount As Integer

    ListItem = New ListItem()
```

```
    ItemCount = 0
    While Not bFound Or ItemCount <= cbCombo.Items.Count - 1
      ListItem = CType(cbCombo.Items(ItemCount), ListItem)
      If ListItem.ID = CInt(strID) Then
        cbCombo.SelectedIndex = ItemCount
        bFound = True
      End If
      ItemCount += 1
    End While

    If Not bFound Then
      cbCombo.SelectedIndex = -1
    End If

  End Sub

  Private Sub FindProductByName(ByVal lbList As ListBox, ByVal strValue As
  String)
    ' Thủ tục này sử dụng để tìm một Product trong Product List Box và chọn nó
    Dim blsFound As Boolean
    Dim ListItem As ListItem
    Dim ItemCount As Integer

    ListItem = New ListItem()

    ItemCount = 0
    ' Duyệt qua danh sách cho đến khi tìm thấy
    While Not blsFound And ItemCount <= lbList.Items.Count - 1
      ListItem = CType(lbList.Items(ItemCount), ListItem)
      If ListItem.Value = strValue Then
        lbList.SelectedIndex = ItemCount
        blsFound = True
      End If
      ItemCount += 1
    End While

    If Not blsFound Then
      lbList.SelectedIndex = 0
    End If
  End Sub
```

```
Private Function IsValidForm() As Boolean
    ' Kiểm tra để bảo đảm các field có giá trị hợp lệ
  If txtProductName.Text = "" Or _
    txtQtyPerUnit.Text = "" Or _
    txtUnitPrice.Text = "" Or _
    txtUnitsInStock.Text = "" Or _
    txtUnitsOnOrder.Text = "" Or _
    txtReorderLevel.Text = "" Or _
    cbCategories.SelectedIndex = -1 Or _
    cbSuppliers.SelectedIndex = -1 Or _
    Not IsNumeric(txtUnitPrice.Text) Or _
    Not IsNumeric(txtUnitsInStock.Text) Or _
    Not IsNumeric(txtUnitsOnOrder.Text) Or _
    Not IsNumeric(txtReorderLevel.Text) Then

    MsgBox("Please enter a valid value for each field.", _
        MsgBoxStyle.Exclamation, Me.Text)
    Return False
  Else
    Return True
  End If
End Function

Private Sub PopulateCategoryCombo()
  ' Thủ tục này thiết lập một ComboBox trên form với một danh sách các Categories

  Dim cnSQL As SqlConnection
  Dim cmSQL As SqlCommand
  Dim drSQL As SqlDataReader
  Dim strSQL As String
  Dim objListItem As ListItem

  Dim frmStatusMessage As New frmStatus()
  If Not HasConnected Then
    frmStatusMessage.Show("Connecting to SQL Server")
  End If

  Dim IsConnecting As Boolean = True
```

```
While IsConnecting

  Try
      ' Xây dựng lệnh Select để lấy Category Name từ bảng Categories
    strSQL = "SELECT CategoryID, CategoryName FROM Categories"

    cnSQL = New SqlConnection(ConnectionString)
    cnSQL.Open()

    cmSQL = New SqlCommand(strSQL, cnSQL)
    drSQL = cmSQL.ExecuteReader()

    cbCategories.Items.Clear()
      ' Duyệt qua tập kết quả và cho các Category Name vào ComboBox
    Do While drSQL.Read()
      objListItem = New ListItem(drSQL.Item("CategoryName").ToString(), _
                    CInt(drSQL.Item("CategoryID")))
      cbCategories.Items.Add(objListItem)
    Loop

    IsConnecting = False
    HasConnected = True

    drSQL.Close()
    cnSQL.Close()
    cmSQL.Dispose()
    cnSQL.Dispose()

  Catch e As SqlException
    If ConnectionString = SQL_CONNECTION_STRING Then
      ConnectionString = MSDE_CONNECTION_STRING
      frmStatusMessage.Show("Connecting to MSDE")
    Else
      frmStatusMessage.Close()
      MsgBox("To run this sample you must have SQL Server ot MSDE.", _
          MsgBoxStyle.Critical, _
          "SQL Server/MSDE not found")
      End
    End If
```

```
        Catch e As Exception
            MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
        End Try
        frmStatusMessage.Close()
    End While
End Sub

Private Sub PopulateForm()
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
    Dim objListItem As ListItem
    Dim strID As String

    Try
        ' Lấy khóa chính từ ListBox
        objListItem = CType(lstProducts.SelectedItem, ListItem)

        ' Xây dựng câu lệnh Select để lấy thông tin sản phẩm từ bảng Products
        strSQL = "SELECT ProductID, " & _
                 "    ProductName, " & _
                 "    QuantityPerUnit, " & _
                 "    UnitPrice, " & _
                 "    UnitsInStock, " & _
                 "    UnitsOnOrder, " & _
                 "    ReorderLevel, " & _
                 "    Discontinued, " & _
                 "    SupplierID, " & _
                 "    CategoryID " & _
                 "FROM Products " & _
                 "WHERE ProductID = " & objListItem.ID

        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        drSQL = cmSQL.ExecuteReader()
```

```
        If drSQL.Read() Then
            ' Thiết lập form với dữ liệu
            txtProductID.Text = drSQL.Item("ProductID").ToString()
            txtProductName.Text() = drSQL.Item("ProductName").ToString()
            txtQtyPerUnit.Text() = drSQL.Item("QuantityPerUnit").ToString()
            txtUnitPrice.Text() = drSQL.Item("UnitPrice").ToString()
            txtUnitsInStock.Text() = drSQL.Item("UnitsInStock").ToString()
            txtUnitsOnOrder.Text() = drSQL.Item("UnitsOnOrder").ToString()
            txtReorderLevel.Text() = drSQL.Item("ReorderLevel").ToString()
            chkDiscontinued.Checked = CType(drSQL.Item("Discontinued"), Boolean)
            strID = drSQL.Item("SupplierID").ToString()
            FindItemByID(cbSuppliers, strID)
            strID = drSQL.Item("CategoryID").ToString()
            FindItemByID(cbCategories, strID)
        End If

        drSQL.Close()
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()

    Catch e As SqlException
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL Error")
    Catch e As Exception
        MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
    End Try
End Sub

Private Sub PopulateProductList()
    ' Thủ tục này thiết lập ListBox trên form với danh sách các sản phẩm
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim drSQL As SqlDataReader
    Dim strSQL As String
    Dim objListItem As ListItem

    Try
        ' Xây dựng câu lệnh Select để lấy các ProductName từ bảng Products
        strSQL = "SELECT ProductName, ProductID FROM Products"
```

```
        cnSQL = New SqlConnection(ConnectionString)
        cnSQL.Open()

        cmSQL = New SqlCommand(strSQL, cnSQL)
        drSQL = cmSQL.ExecuteReader()

        lstProducts.Items.Clear()

          ' Duyệt qua tập dữ liệu kết quả bằng class DataReader. DataReader được
          ' sử dụng vì chỉ cần duyệt tới sẽ có hiệu quả hơn.
        Do While drSQL.Read()
          objListItem = New ListItem(drSQL.Item("ProductName").ToString(), _
                        CInt(drSQL.Item("ProductID")))
          lstProducts.Items.Add(objListItem)
        Loop

        If lstProducts.Items.Count > 0 Then
          lstProducts.SetSelected(0, True)
        End If

        drSQL.Close()
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()

      Catch e As SqlException
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL Error")

      Catch e As Exception
        MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
      End Try
    End Sub

    Private Sub PopulateSupplierCombo()
      ' Thủ tục này thiết lập ComboBox với danh sách các Supplier
      Dim cnSQL As SqlConnection
      Dim cmSQL As SqlCommand
      Dim drSQL As SqlDataReader
      Dim strSQL As String
      Dim objListItem As ListItem
```

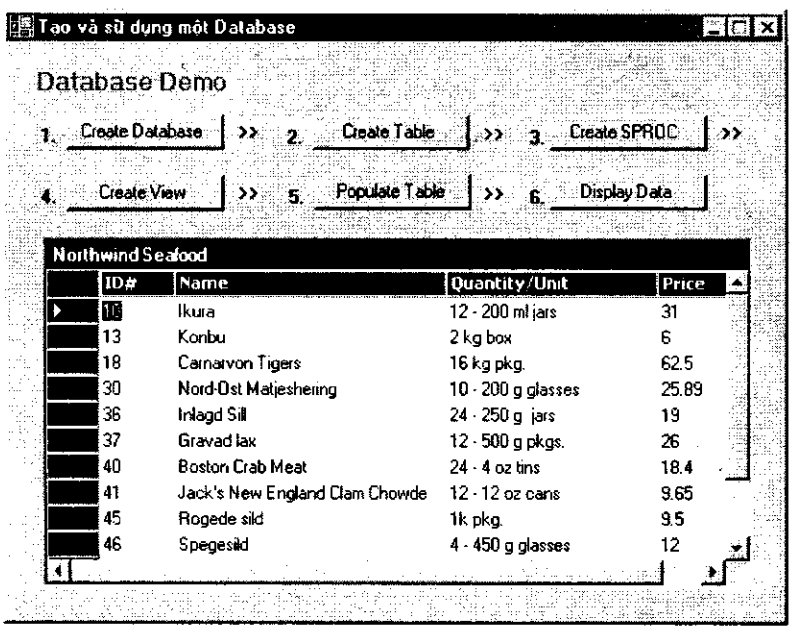
```
    Try
      strSQL = "SELECT SupplierID, CompanyName FROM Suppliers"

      cnSQL = New SqlConnection(ConnectionString)
      cnSQL.Open()

      cmSQL = New SqlCommand(strSQL, cnSQL)
      drSQL = cmSQL.ExecuteReader()

      cbSuppliers.Items.Clear()
      Do While drSQL.Read()
        objListItem = New ListItem(drSQL.Item("CompanyName").ToString(), _
                      CInt(drSQL.Item("SupplierID")))
        cbSuppliers.Items.Add(objListItem)
      Loop

      drSQL.Close()
      cnSQL.Close()
      cmSQL.Dispose()
      cnSQL.Dispose()

    Catch e As SqlException
      MsgBox(e.Message, MsgBoxStyle.Critical, "SQL Error")

    Catch e As Exception
      MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
    End Try
  End Sub

  Private Function PrepareStr(ByVal strValue As String) As String
    ' Hàm này nhận vào một chuỗi và tạo một chuỗi có thể sử dụng được trong
    ' lệnh SQL bằng cách bọc nó trong dấu ngoặc đơn ' ' và xử lý các khoảng trắng.
    If strValue.Trim() = "" Then
      Return "NULL"
    Else
      Return "'" & strValue.Trim() & "'"
    End If
  End Function
```

```
Private Sub UpdateProduct()
' Thủ tục này được sử dụng để cập nhật một record đã tồn tại với các giá trị trên form
    Dim cnSQL As SqlConnection
    Dim cmSQL As SqlCommand
    Dim strSQL As String
    Dim intRowsAffected As Integer

     ' Kiểm tra các giá trị trên form
    If Not IsValidForm() Then
      Exit Sub
    End If

    Try
       ' Xây dựng lệnh Update để cập nhật bảng Products
      strSQL = "UPDATE Products SET" & _
            " ProductName = " & PrepareStr(txtProductName.Text) & _
            " ,QuantityPerUnit = " & PrepareStr(txtQtyPerUnit.Text) & _
            " ,UnitPrice = " & txtUnitPrice.Text & _
            " ,UnitsInStock = " & txtUnitsInStock.Text & _
            " ,UnitsOnOrder = " & txtUnitsOnOrder.Text & _
            " ,ReorderLevel = " & txtReorderLevel.Text & _
            " ,SupplierID = " & _
     CType(cbSuppliers.Items(cbSuppliers.SelectedIndex), ListItem).ID & _
            " ,CategoryID = " & _
     CType(cbCategories.Items(cbCategories.SelectedIndex), ListItem).ID & _
            " ,Discontinued = " & CType(IIf(chkDiscontinued.Checked, "1", "0"), _
                                        String) & " " & _
            "WHERE ProductID = " & CInt(txtProductID.Text)

      cnSQL = New SqlConnection(ConnectionString)
      cnSQL.Open()

      cmSQL = New SqlCommand(strSQL, cnSQL)
      intRowsAffected = cmSQL.ExecuteNonQuery()

      If intRowsAffected <> 1 Then
        MsgBox("Update Failed.", MsgBoxStyle.Critical, "Update")
      End If
```

```
        cnSQL.Close()
        cmSQL.Dispose()
        cnSQL.Dispose()

    Catch e As SqlException
        MsgBox(e.Message, MsgBoxStyle.Critical, "SQL Error")

    Catch e As Exception
        MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
    End Try
  End Sub

  Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnRefresh.Click
    PopulateCategoryCombo()
    PopulateSupplierCombo()
    PopulateProductList()
  End Sub
End Class
```

49. Tạo và sử dụng một Database

Chương trình này cho biết cách thực hiện các lệnh SQL từ code sử dụng ADO.NET.



Các chức năng chính:

Chương trình này tạo một database, một bảng, một stored procedure, và một view. Sau đó các lệnh SQL được thực hiện để thiết lập bảng với dữ liệu lấy từ database Northwind. Sau đó truy vấn bảng này để lấy dữ liệu đưa vào một DataSet và hiển thị trên một DataGrid.

Để chạy ứng dụng nhấn F5. Nếu muốn chạy lại ứng dụng sau khi đã tạo database, thì cần đóng tất cả các ứng dụng đang duy trì kết nối đến database này.

Project: **Tao mot Database**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=;" & _
        "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=;" & _
        "Integrated Security=SSPI"

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL or MSDE. "

    Protected bolDidPreviouslyConnect As Boolean = False
    Protected bolDidCreateTable As Boolean = False
    Protected connectionString As String = SQL_CONNECTION_STRING

    ' Code phát sinh form được bỏ qua
```

```
        MessageBox.Show("Table 'NW_Seafood' successfully created.", _
          "Table Creation Status", _
          MessageBoxButtons.OK, MessageBoxIcon.Information)
      Catch sqlExc As SqlException
        MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
          MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try
    End Sub

    ' Xử lý cho nút Create View.
    Private Sub btnCreateView_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCreateView.Click
      Dim northwindConnection As New SqlConnection(connectionString)
      Dim strSQL As String = _
        "USE HowToDemo" & vbCrLf & _
        "IF EXISTS (" & _
        "SELECT * " & _
        "FROM HowToDemo.dbo.sysobjects " & _
        "WHERE Name = 'GetSeafood' " & _
        "AND TYPE = 'v')" & vbCrLf & _
        "BEGIN" & vbCrLf & _
        "DROP VIEW GetSeafood" & vbCrLf & _
        "END"

      Dim cmd As New SqlCommand(strSQL, northwindConnection)
      Try
        northwindConnection.Open()
        cmd.ExecuteNonQuery()
      Catch sqlExc As SqlException
        MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
          MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try

      Try
        cmd.CommandText = _
          "CREATE VIEW GetSeafood AS " & _
          "SELECT * " & _
          "FROM NW_Seafood"

        cmd.ExecuteNonQuery()
```

```
        northwindConnection.Close()

        lblArrow4.Visible = True
        lblStep5.Enabled = True
        btnPopulate.Enabled = True

        MessageBox.Show("View 'GetSeafood' successfully " & _
          "created.", "SQL View Creation Status", _
          MessageBoxButtons.OK, MessageBoxIcon.Information)

    Catch sqlExc As SqlException
        MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
          MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub

' Xử lý cho nút Display. Thủ tục này lấy thông tin các sản phẩm từ
' bảng NW_Seafood đưa vào một DataSet để hiển thị trên một DataGrid.
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisplay.Click

    If IsNothing(dgSeafood.DataSource) Then

        Dim strSQL As String = _
          "USE HowToDemo" & vbCrLf & _
          "SELECT * " & _
          "FROM GetSeafood"

        Try
          Dim northwindConnection As New SqlConnection(connectionString)
          Dim cmd As New SqlCommand(strSQL, northwindConnection)
          Dim da As New SqlDataAdapter(cmd)
          Dim dsSeafood As New DataSet()
          da.Fill(dsSeafood, "Seafood")
            ' Quy định tiêu đề cho DataGrid, kết nó vào DataSet, và hiển thị nó.
          With dgSeafood
            .CaptionText = "Northwind Seafood"
            .DataSource = dsSeafood.Tables(0)
            .Visible = True
          End With
```

Các chức năng chính:

Chương trình này tạo một database, một bảng, một stored procedure, và một view. Sau đó các lệnh SQL được thực hiện để thiết lập bảng với dữ liệu lấy từ database Northwind. Sau đó truy vấn bảng này để lấy dữ liệu đưa vào một DataSet và hiển thị trên một DataGrid.

Để chạy ứng dụng nhấn F5. Nếu muốn chạy lại ứng dụng sau khi đã tạo database, thì cần đóng tất cả các ứng dụng đang duy trì kết nối đến database này.

Project: **Tao mot Database**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=;" & _
        "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=;" & _
        "Integrated Security=SSPI"

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL or MSDE. "

    Protected bolDidPreviouslyConnect As Boolean = False
    Protected bolDidCreateTable As Boolean = False
    Protected connectionString As String = SQL_CONNECTION_STRING

' Code phát sinh form được bỏ qua
```

```
' Thủ tục này thực hiện một lệnh SQL xóa database (nếu đã tồn tại) và tạo nó
Private Sub CreateTable()
  Dim strSQL As String = _
    "IF EXISTS (" & _
    "SELECT * " & _
    "FROM master..sysdatabases " & _
    "WHERE Name = 'HowToDemo')" & vbCrLf & _
    "DROP DATABASE HowToDemo" & vbCrLf & _
    "CREATE DATABASE HowToDemo"

  Dim frmStatusMessage As New frmStatus()
  If Not bolDidPreviouslyConnect Then
    frmStatusMessage.Show("Connecting to SQL Server")
  End If

  Dim bolIsConnecting As Boolean = True
  While bolIsConnecting
    Try
      Dim northwindConnection As New SqlConnection(connectionString)

      ' Đối tượng SqlCommand được sử dụng để thực hiện các lệnh SQL
      Dim cmd As New SqlCommand(strSQL, northwindConnection)

      ' Mở kết nối, thực hiện lệnh, và đóng kết nối. Sử dụng ExecuteNonQuery
      ' sẽ hiệu quả hơn khi không cần trả về dữ liệu
      northwindConnection.Open()
      cmd.ExecuteNonQuery()
      northwindConnection.Close()

      ' Dữ liệu đã truyền đi, thoát vòng lặp
      bolIsConnecting = False
      bolDidPreviouslyConnect = True
      bolDidCreateTable = True
      frmStatusMessage.Close()

      ' Hiển thị các control cho bước kế tiếp
      lblArrow1.Visible = True
      lblStep2.Enabled = True
      btnCreateTable.Enabled = True
```

```
            MessageBox.Show("Database 'HowToDemo' successfully created!", _
              "Database Creation Status", MessageBoxButtons.OK, _
              MessageBoxIcon.Information)

          Catch sqlExc As SqlException
            MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
              MessageBoxButtons.OK, MessageBoxIcon.Error)
            Exit While
          Catch exc As Exception
            If connectionString = SQL_CONNECTION_STRING Then
                ' Không thể kết nối vào SQL Server, chuyển qua MSDE
              connectionString = MSDE_CONNECTION_STRING
              frmStatusMessage.Show("Connecting to MSDE")
            Else
                ' Không thể kết nối vào SQL hay MSDE
              frmStatusMessage.Close()
              MessageBox.Show(CONNECTION_ERROR_MSG, _
                  "Connection Failed!", MessageBoxButtons.OK, _
                  MessageBoxIcon.Error)
              End
            End If
          End Try
        End While
        frmStatusMessage.Close()
    End Sub

    ' Thiết lập giao diện để người sử dụng làm việc đúng theo thứ tự
    Private Sub ResetUI()
        lblArrow1.Visible = False
        lblStep2.Enabled = False
        btnCreateTable.Enabled = False
        lblArrow2.Visible = False
        lblStep3.Enabled = False
        btnCreateSP.Enabled = False
        lblArrow3.Visible = False
        lblStep4.Enabled = False
        btnCreateView.Enabled = False
        lblArrow4.Visible = False
        lblStep5.Enabled = False
```

```
        btnPopulate.Enabled = False
        lblArrow5.Visible = False
        lblStep6.Enabled = False
        btnDisplay.Enabled = False
        With dgSeafood
            .Visible = False
            .DataSource = Nothing
            .TableStyles.Clear()
            .Visible = False
        End With
    End Sub

    ' Xử lý cho nút Create Database
    Private Sub btnCreateDB_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCreateDB.Click
        ' Nếu nút "Create Table" đang cho phép, có nghĩa là người sử dụng
        ' muốn tạo lại database.
        If btnCreateTable.Enabled Then
            Dim dr As DialogResult = MessageBox.Show("Recreating the database " & _
                "will undo the other database object creation steps you have " & _
                "made so far. Do you wish to proceed?", "Database Re-creation " & _
                "Warning", MessageBoxButtons.YesNo, MessageBoxIcon.Warning)
            If dr = DialogResult.Yes Then
                ResetUI()
                CreateTable()
            End If
        Else
            CreateTable()
        End If
    End Sub

    ' Xử lý cho nút Create Procedure. Thủ tục này thực hiện hai lệnh SQL, một lệnh
    ' xóa Procedure nếu nó đã tồn tại và lệnh còn lại tạo Procedure. SQL Server
    ' không cho phép kết hợp hai lệnh này cùng lúc trong code.
    Private Sub btnCreateSP_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCreateSP.Click

        Dim northwindConnection As New SqlConnection(connectionString)
```

```
Dim strSQL As String = _
   "USE HowToDemo" & vbCrLf & _
   "IF EXISTS (" & _
   "SELECT * " & _
   "FROM HowToDemo.dbo.sysobjects " & _
   "WHERE Name = 'AddSeafood' " & _
   "AND TYPE = 'p')" & vbCrLf & _
   "BEGIN" & vbCrLf & _
   "DROP PROCEDURE AddSeafood" & vbCrLf & _
   "END"

Dim cmd As New SqlCommand(strSQL, northwindConnection)
Try
   northwindConnection.Open()
   cmd.ExecuteNonQuery()

Catch sqlExc As SqlException
   MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
      MessageBoxButtons.OK, MessageBoxIcon.Error)
End Try

Try
   cmd.CommandText = _
      "CREATE PROCEDURE AddSeafood AS" & vbCrLf & _
      "INSERT INTO NW_Seafood" & vbCrLf & _
      "(ProductID, ProductName, QuantityPerUnit, UnitPrice)" & _
      "SELECT ProductID, ProductName, QuantityPerUnit, UnitPrice " & _
      "FROM Northwind.dbo.Products " & _
      "WHERE CategoryID = 8"

   cmd.ExecuteNonQuery()
   northwindConnection.Close()

   lblArrow3.Visible = True
   lblStep4.Enabled = True
   btnCreateView.Enabled = True
   MessageBox.Show("Stored Procedure 'AddSeafood' successfully " & _
      "created.", "SPROC Creation Status", _
      MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
    Catch sqlExc As SqlException
      MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
  End Sub

  ' Xử lý cho nút Create Table
  Private Sub btnCreateTable_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnCreateTable.Click

    Dim strSQL As String = _
      "USE HowToDemo" & vbCrLf & _
      "IF EXISTS (" & _
      "SELECT * " & _
      "FROM HowToDemo.dbo.sysobjects " & _
      "WHERE Name = 'NW_Seafood' " & _
      "AND TYPE = 'u')" & vbCrLf & _
      "BEGIN" & vbCrLf & _
      "DROP TABLE HowToDemo.dbo.NW_Seafood" & vbCrLf & _
      "END" & vbCrLf & _
      "CREATE TABLE NW_Seafood (" & _
      "ProductID Int NOT NULL," & _
      "ProductName NVarChar(40) NOT NULL," & _
      "QuantityPerUnit NVarChar(20) NOT NULL," & _
      "UnitPrice Money NOT NULL," & _
      "CONSTRAINT [PK_Product] PRIMARY KEY CLUSTERED" & _
      "(ProductID))"

    Try
      Dim northwindConnection As New SqlConnection(connectionString)
      Dim cmd As New SqlCommand(strSQL, northwindConnection)

      northwindConnection.Open()
      cmd.ExecuteNonQuery()
      northwindConnection.Close()
      lblArrow2.Visible = True
      lblStep3.Enabled = True
      btnCreateSP.Enabled = True
```

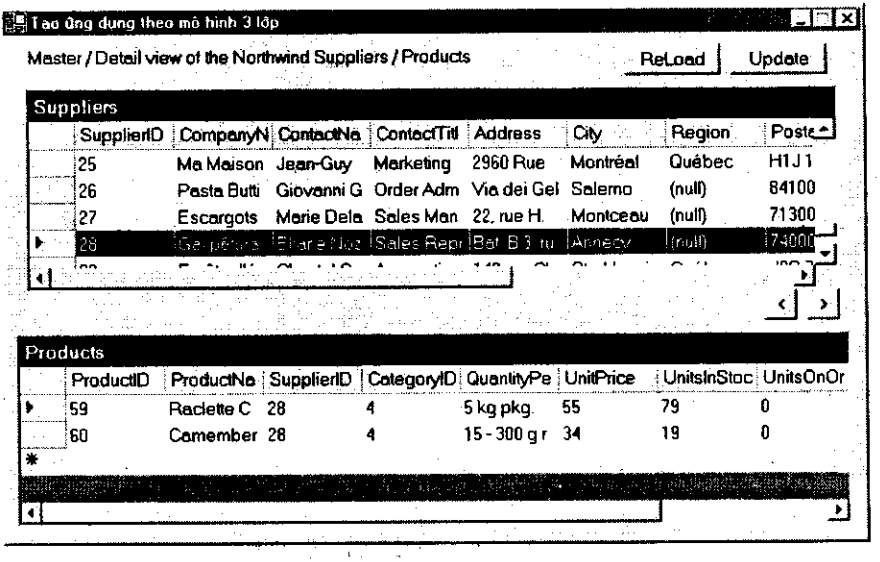
```
    ' Sử dụng đối tượng TableStyle để định dạng DataGrid
  Dim dgTableStyle1 As New DataGridTableStyle()
  With dgTableStyle1
    .AlternatingBackColor = Color.Lavender
    .BackColor = Color.WhiteSmoke
    .ForeColor = Color.MidnightBlue
    .GridLineColor = Color.Gainsboro
    .GridLineStyle = System.Windows.Forms.DataGridLineStyle.None
    .HeaderBackColor = Color.MidnightBlue
    .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
    .HeaderForeColor = Color.WhiteSmoke
    .LinkColor = Color.Teal
      ' Đừng quên quy định thuộc tính MappingName
    .MappingName = "Seafood"
    .SelectionBackColor = Color.CadetBlue
    .SelectionForeColor = Color.WhiteSmoke
  End With
    ' Sử dụng các đối tượng ColumnStyle để định dạng cho từng cột
  Dim grdColStyle1 As New DataGridTextBoxColumn()
  With grdColStyle1
    .HeaderText = "ID#"
    .MappingName = "ProductID"
    .Width = 50
  End With

  Dim grdColStyle2 As New DataGridTextBoxColumn()
  With grdColStyle2
    .HeaderText = "Name"
    .MappingName = "ProductName"
    .Width = 180
  End With

  Dim grdColStyle3 As New DataGridTextBoxColumn()
  With grdColStyle3
    .HeaderText = "Quantity/Unit"
    .MappingName = "QuantityPerUnit"
    .Width = 140
  End With
```

```
        Dim grdColStyle4 As New DataGridTextBoxColumn()
        With grdColStyle4
          .HeaderText = "Price"
          .MappingName = "UnitPrice"
          .Width = 70
        End With

        dgTableStyle1.GridColumnStyles.AddRange _
          (New DataGridColumnStyle() _
          {grdColStyle1, grdColStyle2, grdColStyle3, grdColStyle4})
        dgSeafood.TableStyles.Add(dgTableStyle1)

      Catch sqlExc As SqlException
        MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
          MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try
    End If
  End Sub

  ' Xử lý cho nút Populate. Thủ tục này thực hiện stored procedure lấy thông tin
  ' sản phẩm từ bảng Products của database Northwind đưa vào bảng NW_Seafood.
  Private Sub btnPopulate_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnPopulate.Click

    Dim strSQL As String = "EXECUTE HowToDemo.dbo.AddSeafood"

    Try
      Dim northwindConnection As New SqlConnection(connectionString)
      Dim cmd As New SqlCommand(strSQL, northwindConnection)
      northwindConnection.Open()
      cmd.ExecuteNonQuery()
      northwindConnection.Close()
      lblArrow5.Visible = True
      lblStep6.Enabled = True
      btnDisplay.Enabled = True

      MessageBox.Show("Table successfully populated.", _
        "Data Addition Status", _
        MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
        Catch sqlExc As SqlException
            MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
                MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        ResetUI()
    End Sub
End Class
```

50. Tạo ứng dụng theo mô hình 3 lớp (3-Tier)

Chương trình này cho biết cách xây dựng một ứng dụng dữ liệu theo mô hình 3 lớp. Sử dụng SQL Server hoặc MSDE làm lớp Back end (sử dụng database Nortwind), Lớp truy xuất dữ liệu – lớp giữa, và form kết xuất dữ liệu – front end. Front end là một form kết dữ liệu thông dụng và dùng để duyệt qua các record và hiển thị nhiều cấp dữ liệu chi tiết có quan hệ đến mỗi record.



Các chức năng chính:

- Cách tạo lớp truy xuất dữ liệu ở lớp giữa, lớp này sẽ tạo một đối tượng DataSet được sử dụng trong front-end.

- Cách các sự kiện được sử dụng trong back-end để cung cấp các thông tin trạng thái cho front-end.
- Ví dụ kết dữ liệu ở front-end.
- Cách một DataSet chứa hai DataTable và kết vào nhiều control có đồng bộ lẫn nhau.

Project: **Tao ung dung theo mo hinh 3 lop**

DataAccessLayer.vb

```
Option Strict On
Imports System.Data.SqlClient

' Class này cung cấp truy xuất đến database cho front-end. Nó truyền một DataSet
' chứa thông tin Supplier và Product cho front end.
Public Class DataAccessLayer

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL or MSDE."

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Protected DidPreviouslyConnect As Boolean = False
    ' Tạo hai DataAdapter cho hai bảng
    Private sdaSuppliers As SqlDataAdapter
    Private sdaProducts As SqlDataAdapter

    Protected strConn As String = SQL_CONNECTION_STRING

    ' Khai báo các sự kiện dùng để cung cấp thông tin cho front-end
    Public Event ConnectionStatusChange(ByVal status As String)
    Public Event ConnectionFailure(ByVal reason As String)
    Public Event ConnectionCompleted(ByVal success As Boolean)
```

```
' Tạo DataSet chứa hai bảng dữ liệu Supplier và Product
Public Function CreateDataSet() As DataSet

  Dim dsSupplierProducts As DataSet

  ' Phát sinh sự kiện cho biết là người sử dụng thực hiện kết nối
  If Not DidPreviouslyConnect Then
    RaiseEvent ConnectionStatusChange("Connecting to SQL Server")
  End If

  ' Thực hiện kết nối vào SQL Server (hoặc MSDE) cục bộ
  Dim IsConnecting As Boolean = True
  While IsConnecting
    Try
      Dim scnnNW As New SqlConnection(strConn)

      Dim strSQL As String = _
        "SELECT * FROM Suppliers"

      Dim scmd As New SqlCommand(strSQL, scnnNW)

        ' SqlDataAdapter dùng đối tượng SqlCommand để đưa dữ liệu vào DataSet
      sdaSuppliers = New SqlDataAdapter(scmd)

        ' Đối tượng SqlCommandBuilder tự động phát sinh các lệnh SQL cần
        ' thiết để cập nhật database sau này
      Dim scb As New SqlCommandBuilder(sdaSuppliers)
        ' Tạo mới một DataSet và đưa bảng đầu tiên vào
      dsSupplierProducts = New DataSet()
      Me.sdaSuppliers.Fill(dsSupplierProducts, "Supplier")

        ' Tạo mới một SqlDataAdapter, lần này truyền lệnh SQL vào
      sdaProducts = New SqlDataAdapter("SELECT * FROM Products", _
                                       scnnNW)

      Dim scbProducts As New SqlCommandBuilder(sdaProducts)

        ' Đưa bảng thứ hai vào DataSet
      sdaProducts.Fill(dsSupplierProducts, "Product")
```

```
            ' Để thấy được giao diện Master-Details trên cùng một DataGrid, bỏ ghi
            ' chú các dòng code sau, chúng thiết lập một mối quan hệ cha con.

            'dsSupplierProducts.Relations.Add("Supplier_Products", _
            '    dsSupplierProducts.Tables("Supplier").Columns("SupplierID"), _
            '    dsSupplierProducts.Tables("Product").Columns("SupplierID"))

            ' Dữ liệu đã được lấy thành công, thoát vòng lặp

          IsConnecting = False
          DidPreviouslyConnect = True

        Catch exp As Exception
          If strConn = SQL_CONNECTION_STRING Then
              ' Không thể kết nối vào SQL Server. Chuyển qua MSDE
            strConn = MSDE_CONNECTION_STRING
            RaiseEvent ConnectionStatusChange("Connecting to MSDE")
          Else
              ' Không thể kết nối vào SQL hay MSDE
            RaiseEvent ConnectionFailure(CONNECTION_ERROR_MSG)
            End
          End If
        End Try
      End While
      ' Thông báo thành công
      RaiseEvent ConnectionCompleted(True)

      ' Trả về DataSet có dữ liệu
      Return dsSupplierProducts
    End Function

    ' Thủ tục này nhận vào một DataSet và cập nhật các thay đổi
    ' vào database Northwind
    Public Sub UpdateDataSet(ByVal inDS As DataSet)
      ' Nếu DataSet truyền vào là Nothing thì thoát thủ tục
      If inDS Is Nothing Then
        Exit Sub
      End If
```

```
Try
    ' Trước tiên kiểm tra các DataAdapter đã được tạo,
    ' nếu chưa thì gọi CreateDataSet để tạo chúng.
  If (Me.sdaProducts Is Nothing Or Me.sdaSuppliers Is Nothing) Then
    CreateDataSet()
  End If

    ' Cập nhật DataSet. Cần phải thực hiện mọi thứ theo đúng thứ tự, trừ
    ' khi tắt thuộc tính EnforceConstraints

    ' Ta có hai cách để cập nhật dữ liệu

    ' Cách đầu tiên là tắt thuộc tính EnforceConstraints
  inDS.EnforceConstraints = False
  Me.sdaProducts.Update(inDS, "Product")
  Me.sdaSuppliers.Update(inDS, "Supplier")

    ' Ta cũng có thể thực hiện cập nhật theo thứ tự để không xảy ra lỗi ràng buộc.
    'If Not inDS.GetChanges(DataRowState.Deleted) Is Nothing Then
    '   Me.sdaProducts.Update(inDS.GetChanges(DataRowState.Deleted), _
    '                         "Products")
    '   Me.sdaSuppliers.Update(inDS.GetChanges(DataRowState.Deleted), _
    '                         "Suppliers")
    'End If

    'If Not inDS.GetChanges(DataRowState.Detached) Is Nothing Then
    '   Me.sdaProducts.Update(inDS.GetChanges(DataRowState.Detached), _
    '                         "Products")
    '   Me.sdaSuppliers.Update(inDS.GetChanges(DataRowState.Detached), _
    '                         "Suppliers")
    'End If

    'If Not inDS.GetChanges(DataRowState.Added) Is Nothing Then
    '   Me.sdaSuppliers.Update(inDS.GetChanges(DataRowState.Added), _
    '                         "Suppliers")
    '   Me.sdaProducts.Update(inDS.GetChanges(DataRowState.Added), _
    '                         "Products")
    'End If
```

```
        'If Not inDS.GetChanges(DataRowState.Modified) Is Nothing Then
        '    Me.sdaSuppliers.Update(inDS.GetChanges(DataRowState.Modified), _
        '                           "Suppliers")
        '    Me.sdaProducts.Update(inDS.GetChanges(DataRowState.Modified), _
        '                          "Products")
        'End If

    Catch exc As Exception
        ' Thông báo cho front-end biết có lỗi xảy ra
      RaiseEvent ConnectionFailure("Unable to update the data source.")
    End Try
  End Sub

End Class
```

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class frmMain
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
  Protected DidPreviouslyConnect As Boolean = False
  Private dsSupplierProducts As DataSet
  Private dtSupplier As DataTable
  Private dtProduct As DataTable
  Private dvSupplier As DataView
  Private dvProduct As DataView
  Protected WithEvents m_DAL As DataAccessLayer
  Dim frmStatusMessage As New frmStatus()

  ' Xử lý nút >
  Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnNext.Click
      ' Di chuyển đến record kế tiếp
    NextRecord()
  End Sub
```

```
' Xử lý nút <
Private Sub btnPrevious_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrevious.Click
    ' Di chuyển lùi về một record
    PreviousRecord()
End Sub

' Refresh dữ liệu bằng cách gọi lại form_load. Hành động này lấy dữ liệu từ
' database mà không thực hiện thay đổi
Private Sub btnRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefresh.Click
    ' Gọi thủ tục load form
    frmMain_Load(Me, New System.EventArgs())
End Sub

' Nút này gửi các lệnh Update đến đối tượng DataAccessLayer, và load lại dữ liệu
Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdate.Click
    Dim m_DAL As New DataAccessLayer()
    m_DAL.UpdateDataSet(dsSupplierProducts.GetChanges())
    frmMain_Load(Me, New System.EventArgs())
End Sub

' Xử lý sự kiện PositionChanged
Protected Sub dtSupplier_PositionChanged(ByVal sender As Object, ByVal e As
System.EventArgs)
    BindSuppliersGrid()
    BindProductsGrid()
End Sub

' Xử lý sự kiện KeyDown cho form. Thuộc tính KeyPreview phải có giá trị True.
Private Sub frmMain_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles MyBase.KeyDown
    ' Cho phép người sử dụng duyệt qua các record bằng cách sử dụng bàn phím
    If e.KeyCode = Keys.Right Then NextRecord()
    If e.KeyCode = Keys.Left Then PreviousRecord()
End Sub
```

```
Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    frmStatusMessage = New frmStatus()
    GetDataSet()
    BindSuppliersGrid()
    BindProductsGrid()
End Sub

' Thủ tục này bảo đảm nếu một dòng mới được thêm vào, một SupplierID đúng
' được tự động thêm vào. Chú ý: Cũng cần có thủ tục cho sự kiện
' grdProducts.CurrentCellChanged
Private Sub grdProducts_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles grdProducts.Click
    ' Nếu là dòng mới, ta phải tự động đưa giá trị vào trường SupplierID
    ' (vì đây là một form Master-Details)
    If TypeOf (grdProducts.Item(grdProducts.CurrentRowIndex, 2)) Is DBNull Then
        ' Đây là dòng mới, do đó cho vào một SupplierID đúng
        grdProducts.Item(grdProducts.CurrentRowIndex, 2) = _
            grdSuppliers.Item(grdSuppliers.CurrentRowIndex, 0)
    End If
End Sub

Private Sub grdProducts_CurrentCellChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles grdProducts.CurrentCellChanged
    grdProducts.Select(grdProducts.CurrentCell.RowNumber)

    If TypeOf (grdProducts.Item(grdProducts.CurrentRowIndex, 2)) Is DBNull Then
        grdProducts.Item(grdProducts.CurrentRowIndex, 2) = _
            grdSuppliers.Item(grdSuppliers.CurrentRowIndex, 0)
    End If
End Sub

' Xử lý sự kiện Click trên DataGrid, phát sinh khi người sử dụng nhấn vào khung --
' không phải là dòng hay cột. Do đó, cần xử lý thêm sự kiện CurrentCellChanged
Private Sub grdSuppliers_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles grdSuppliers.Click
    ' Kết dữ liệu cho lưới Products dựa trên record được chọn trong grdSuppliers
    BindProductsGrid()
End Sub
```

```
Private Sub grdSuppliers_CurrentCellChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles grdSuppliers.CurrentCellChanged
    ' Làm nổi bật dòng được chọn
    grdSuppliers.Select(grdSuppliers.CurrentCell.RowNumber)
    BindProductsGrid()
End Sub

' Đóng form frmStatusMessage khi kết nối hoàn tất
Private Sub m_DAL_ConnectionCompleted(ByVal success As Boolean) Handles
m_DAL.ConnectionCompleted
    frmStatusMessage.Close()
End Sub

' Thông báo cho người sử dụng nếu DAL có lỗi kết nối
Private Sub m_DAL_ConnectionFailure(ByVal reason As String) Handles
m_DAL.ConnectionFailure
    MsgBox(reason, MsgBoxStyle.Critical, Me.Text)
End Sub

' Hiển thị thông báo
Private Sub m_DAL_ConnectionStatusChange(ByVal status As String) Handles
m_DAL.ConnectionStatusChange
    frmStatusMessage.Show(status)
End Sub

' Kết và định dạng lưới Products dựa trên record được chọn trên lưới Suppliers
Sub BindProductsGrid()
    ' Lấy SupplierID hiện hành bằng cách dùng thuộc tính CurrentRowIndex
    ' của DataGrid, sau đó truy xuất đến cột "SupplierID" để lấy SupplierID
    Dim strCurrentSupplierID As String = _
        dvSupplier(grdSuppliers.CurrentRowIndex)("SupplierID").ToString

    If strCurrentSupplierID = "" Then
        strCurrentSupplierID = "-1"
    End If
    ' Lọc các Products để hiển thị dựa trên SupplierID
    dvProduct.RowFilter = "SupplierID = " & strCurrentSupplierID
```

```
    ' Đặt tiêu đề và kết DataSource cho DataGrid
    With grdProducts
      .CaptionText = "Products"
      .DataSource = dvProduct
    End With
  End Sub

  ' Kết dữ liệu vào lưới Suppliers
  Sub BindSuppliersGrid()

    ' Quy định tiêu đề và kết DataSource
    With grdSuppliers
      .CaptionText = "Suppliers"
      .DataSource = dvSupplier
    End With
  End Sub

  ' Tạo DataSet chứa hai bảng Supplier và Products, và các DataView.
  ' DataSet được lấy từ đối tượng DataAccessLayer
  Sub GetDataSet()
    ' Thông báo cho người sử dụng biết dữ liệu lấy từ DAL
    frmStatusMessage.Show("Retrieving Data From Data Access Layer")
    ' Tạo một instance của DataAccessLayer và truy cập dữ liệu
    If m_DAL Is Nothing Then
      m_DAL = New DataAccessLayer()
    End If

    ' Gọi CreateDataSet để đưa dữ liệu vào DataSet cục bộ
    dsSupplierProducts = m_DAL.CreateDataSet()

    ' Quy định các biến DataTable để sử dụng sau này
    dtSupplier = dsSupplierProducts.Tables("Supplier")
    dtProduct = dsSupplierProducts.Tables("Product")

    ' Quy định các DataView là các DataView mặc nhiên của các DataGrid
    dvSupplier = dtSupplier.DefaultView
    dvProduct = dtProduct.DefaultView

  End Sub
```

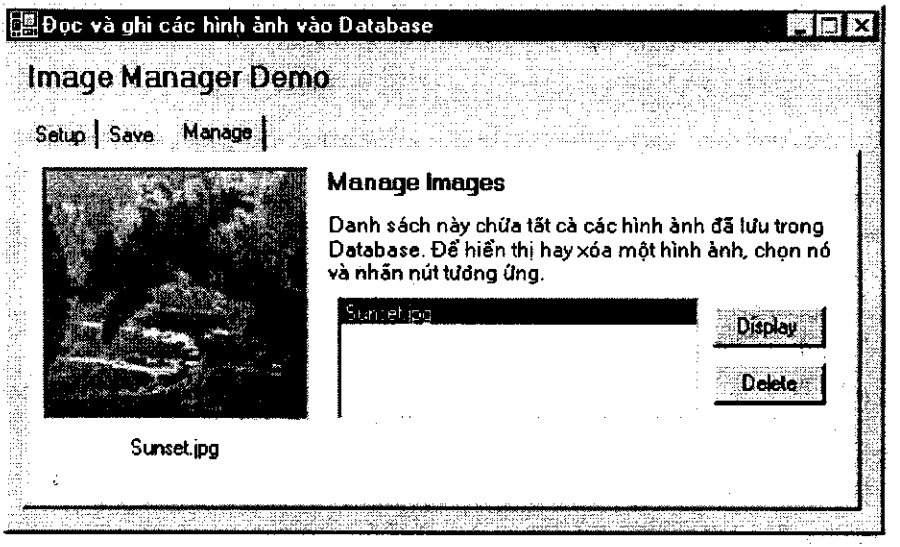
```
    ' Di chuyển Position của Supplier đến record kế tiếp
    Public Sub NextRecord()
        grdSuppliers.UnSelect(grdSuppliers.CurrentRowIndex)
        grdSuppliers.CurrentRowIndex += 1
        grdSuppliers.Select(grdSuppliers.CurrentRowIndex)
    End Sub

    ' Di chuyển lùi về một record
    Public Sub PreviousRecord()
        grdSuppliers.UnSelect(grdSuppliers.CurrentRowIndex)
        If grdSuppliers.CurrentRowIndex > 0 Then
            grdSuppliers.CurrentRowIndex -= 1
        End If
        grdSuppliers.Select(grdSuppliers.CurrentRowIndex)
    End Sub

End Class
```

frmStatus.vb

```
Option Strict On
Public Class frmStatus
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua
    Public Overloads Sub Show(ByVal Message As String)
        lblStatus.Text = Message
        Me.Show()
        Application.DoEvents()
    End Sub
End Class
```

51. Đọc và ghi các hình ảnh vào Database

Chương trình này chỉ cách đọc và ghi các hình ảnh vào một database.



Các chức năng chính:

- Chương trình này sử dụng một TabControl để chia làm ba vùng làm việc chính. Tab Setup cho phép tạo một bảng mới trong database Northwind để lưu các hình ảnh. Tab Save cho phép chọn và xem các hình ảnh, sau đó lưu nó với kiểu dữ liệu nhị phân vào database. Tab Manage cho phép xem và xóa các hình ảnh đã lưu trong database.

Project: **Doc va ghi cac hinh anh vao Database**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.Text
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form
    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"
```

```
    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL or MSDE."

    Protected da As SqlDataAdapter
    Protected cbd As SqlCommandBuilder
    Protected dsPictures As DataSet
    Protected didPreviouslyConnect As Boolean = False
    Protected connectionString As String = SQL_CONNECTION_STRING

' Code phát sinh form được bỏ qua

    ' Xử lý nút Browse, cho phép người sử dụng chọn một hình ảnh, hiển thị nó
    ' trong một Picture Box để lưu vào database.
    Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnBrowse.Click
        ' Sử dụng một OpenFileDialog để cho phép người sử dụng chọn hình ảnh.
        With OpenFileDialog1
            .InitialDirectory = "C:\"
            .Filter = "All Files|*.*|Bitmaps|*.bmp|GIFs|*.gif|JPEGs|*.jpg"
            .FilterIndex = 2
        End With

        ' Khi người sử dụng nhấn nút Open, hiển thị hình ảnh trong PictureBox và
        ' hiển thị đường dẫn đầy đủ.
        If OpenFileDialog1.ShowDialog() = DialogResult.OK Then
            With PictureBox1
                .Image = Image.FromFile(OpenFileDialog1.FileName)
                .SizeMode = PictureBoxSizeMode.CenterImage
                .BorderStyle = BorderStyle.Fixed3D
            End With
            lblFilePath.Text = OpenFileDialog1.FileName
        End If
    End Sub
```

```
' Xử lý nút Delete, cho phép người sử dụng xóa một hình ảnh lưu trong database
Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
    ' Khi không có phần tử nào được chọn, SelectedIndex = -1
    If lstPictures.SelectedIndex < 0 Then
        MessageBox.Show("There are no images in the database to delete.", _
            "Empty Database!", MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else
        ' Sử dụng SqlCommandBuilder để lấy lệnh delete và cập nhật DataSet
        ' sau khi xóa một dòng tương ứng với hình được chọn trong DataSet
        dsPictures.Tables(0).Rows(lstPictures.SelectedIndex).Delete()
        da.UpdateCommand = cbd.GetDeleteCommand
        da.Update(dsPictures)

        ' Xóa hình ảnh và tên file
        lblFileName.Text = ""
        PictureBox2.Image = Nothing
    End If
End Sub

' Xử lý nút Display, cho phép người sử dụng hiển thị một hình ảnh được
' lưu trong database
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisplay.Click
    If lstPictures.SelectedIndex < 0 Then
        MessageBox.Show("There are no images in the database to display.", _
            "Empty Database!", MessageBoxButtons.OK, MessageBoxIcon.Information)
    Else
        ' Kiểu dữ liệu Image trong SQL Server là kiểu dữ liệu nhị phân. Do đó, để
        ' phát sinh một hình ảnh từ nó ta phải tạo đối tượng Stream chứa dữ liệu nhị
        ' phân. Sau đó có thể tạo hình ảnh bằng cách gọi Image.FromStream().
        Dim arrPicture() As Byte = _
        CType(dsPictures.Tables(0).Rows(lstPictures.SelectedIndex)("Picture"), _
        Byte())

        Dim ms As New MemoryStream(arrPicture)

        With PictureBox2
            .Image = Image.FromStream(ms)
            .SizeMode = PictureBoxSizeMode.CenterImage
```

```
            .BorderStyle = BorderStyle.Fixed3D
        End With

        lblFileName.Text = _
         dsPictures.Tables(0).Rows(lstPictures.SelectedIndex)("FileName").ToString

         ' Đóng đối tượng Stream để giải phóng tài nguyên
        ms.Close()
    End If
End Sub

' Xử lý nút Create Table, cho phép xóa (nếu đã tồn tại) và tạo một bảng
' Picture trong database Northwind
Private Sub btnModifyDB_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnModifyDB.Click

    Dim strSQL As String = _
        "IF EXISTS (" & _
        "SELECT * " & _
        "FROM northwind.dbo.sysobjects " & _
        "WHERE NAME = 'Picture' " & _
        "AND TYPE = 'u')" & vbCrLf & _
        "BEGIN" & vbCrLf & _
        "DROP TABLE northwind.dbo.picture" & vbCrLf & _
        "END" & vbCrLf & _
        "CREATE TABLE Picture (" & _
        "PictureID Int IDENTITY(1,1) NOT NULL," & _
        "[FileName] Varchar(255) NOT NULL," & _
        "Picture Image NOT NULL," & _
        "CONSTRAINT [PK_Picture] PRIMARY KEY CLUSTERED" & _
        "(PictureID))"

    Dim frmStatusMessage As New frmStatus()

    If Not didPreviouslyConnect Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim isConnecting As Boolean = True
```

```
    While isConnecting
      Try
        Dim northwindConnection As New SqlConnection(connectionString)

        Dim cmd As New SqlCommand(strSQL, northwindConnection)

        northwindConnection.Open()
        cmd.ExecuteNonQuery()
        northwindConnection.Close()

        isConnecting = False
        didPreviouslyConnect = True
        frmStatusMessage.Close()
        MessageBox.Show("Table successfully created.", "Table Creation Status", _
          MessageBoxButtons.OK, MessageBoxIcon.Information)

      Catch sqlExc As SqlException
        MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
          MessageBoxButtons.OK, MessageBoxIcon.Error)
        Exit While
      Catch exc As Exception
        If connectionString = SQL_CONNECTION_STRING Then
          connectionString = MSDE_CONNECTION_STRING
          frmStatusMessage.Show("Connecting to MSDE")
        Else
          frmStatusMessage.Close()
          MessageBox.Show(CONNECTION_ERROR_MSG, _
              "Connection Failed!", MessageBoxButtons.OK, _
              MessageBoxIcon.Error)
          End
        End If
      End Try
    End While
    frmStatusMessage.Close()
  End Sub
```

```
' Xử lý nút Save, cho phép người sử dụng lưu hình ảnh được hiển thị
' trong PictureBox vào database.
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click

    ' CHUẨN BỊ DỮ LIỆU ĐỂ TRUYỀN CHO DATABASE
    ' Ta chỉ cần lưu tên file, không cần lưu đường dẫn đầy đủ. Do đó, Split đường
    ' dẫn, tạo một mảng chuỗi. Sau đó đảo ngược mảng để ta truyền phần tử đầu
    ' tiên vào SQL Parameter
    Dim arrFilename() As String = Split(lblFilePath.Text, "\")
    arrFilename.Reverse(arrFilename)

    ' Kiểu dữ liệu Image của SQL Server là kiểu dữ liệu nhị phân. Do đó, để lưu nó
    ' vào database ta cần chuyển hình ảnh sang mảng các byte. Có thể sử dụng
    ' đối tượng FileStream để mở file hình ảnh và đọc nó vào một stream, nhưng
    ' MemoryStream với phương thức Image.Save sẽ dễ dàng hơn.
    Dim ms As New MemoryStream()
    PictureBox1.Image.Save(ms, PictureBox1.Image.RawFormat)
    Dim arrImage() As Byte = ms.GetBuffer

    ' Đóng Stream để giải phóng tài nguyên
    ms.Close()

    Dim frmStatusMessage As New frmStatus()
    If Not didPreviouslyConnect Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim isConnecting As Boolean = True
    While isConnecting

        Try
            Dim northwindConnection As New SqlConnection(connectionString)
            Dim strSQL As String = _
                "INSERT INTO Picture (Filename, Picture)" & _
                "VALUES (@Filename, @Picture)"

            Dim cmd As New SqlCommand(strSQL, northwindConnection)
```

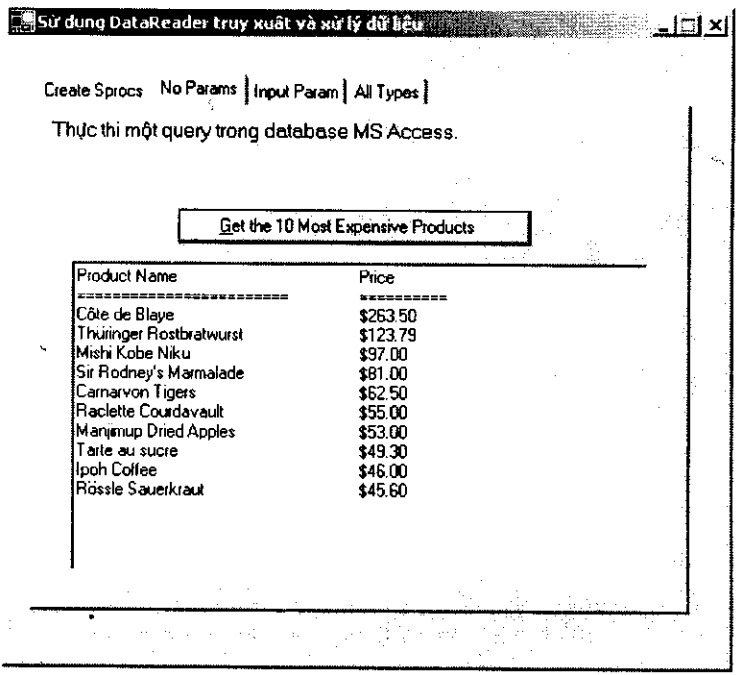
```
        With cmd
            ' Thêm các tham số cần thiết cho lệnh SQL. PictureID là một trường
            ' identity (trong MS Access, là trường AutoNumber), do đó chỉ cần
            ' truyền các giá trị cho hai trường còn lại.
          .Parameters.Add(New SqlParameter("@Filename", _
            SqlDbType.NVarChar, 50)).Value = arrFilename(0)
          .Parameters.Add(New SqlParameter("@Picture", _
            SqlDbType.Image)).Value = arrImage
        End With

        northwindConnection.Open()
        cmd.ExecuteNonQuery()
        northwindConnection.Close()

        isConnecting = False
        didPreviouslyConnect = True
        frmStatusMessage.Close()
        MessageBox.Show(arrFilename(0) & " saved to the database.", _
          "Image Save Status", MessageBoxButtons.OK, _
          MessageBoxIcon.Information)

      Catch sqlExc As SqlException
        MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
          MessageBoxButtons.OK, MessageBoxIcon.Error)
        Exit While
      Catch exc As Exception
        If connectionString = SQL_CONNECTION_STRING Then
          connectionString = MSDE_CONNECTION_STRING
          frmStatusMessage.Show("Connecting to MSDE")
        Else
          frmStatusMessage.Close()
          MessageBox.Show(CONNECTION_ERROR_MSG, _
              "Connection Failed!", MessageBoxButtons.OK, _
              MessageBoxIcon.Error)
          End
        End If
      End Try
    End While
  End Sub
```

```
' Xử lý sự kiện SelectedIndexChanged cho TabControl, để DataSet được tạo và
' kết vào ListBox khi người sử dụng chọn tab Manage.
Private Sub TabControl1_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TabControl1.SelectedIndexChanged

    If TabControl1.SelectedIndex = 2 Then
        Dim frmStatusMessage As New frmStatus()
        If Not didPreviouslyConnect Then
            frmStatusMessage.Show("Connecting to SQL Server")
        End If

        Dim isConnecting As Boolean = True
        While isConnecting
            Try
                Dim northwindConnection As New SqlConnection(connectionString)
                Dim cmd As New SqlCommand("SELECT * " & _
                                "FROM Picture", _
                                northwindConnection)
                da = New SqlDataAdapter(cmd)

                ' SqlCommandBuilder được sử dụng để xóa các hình ảnh khỏi
                ' database trong thủ tục btnDelete
                cbd = New SqlCommandBuilder(da)
                dsPictures = New DataSet()
                da.Fill(dsPictures)

                isConnecting = False
                didPreviouslyConnect = True
                frmStatusMessage.Close()

                ' Hiển thị các tên file
                With lstPictures
                    .DataSource = dsPictures.Tables(0)
                    .DisplayMember = "FileName"
                End With

            Catch sqlExc As SqlException
                MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
                    MessageBoxButtons.OK, MessageBoxIcon.Error)
                Exit While
```

```
            Catch exc As Exception
              If connectionString = SQL_CONNECTION_STRING Then
                connectionString = MSDE_CONNECTION_STRING
                frmStatusMessage.Show("Connecting to MSDE")
              Else
                frmStatusMessage.Close()
                MessageBox.Show(CONNECTION_ERROR_MSG, _
                    "Connection Failed!", MessageBoxButtons.OK, _
                    MessageBoxIcon.Error)
              End If
            End Try
          End While
        End If
    End Sub
End Class
```

52. Truy xuất và xử lý dữ liệu sử dụng DataReader

Chương trình này biểu diễn nhiều cách sử dụng DataReader để truy xuất dữ liệu và sử dụng nó để đưa dữ liệu vào ComboBox, TextBox và các control khác.



Các chức năng chính:

- Chương trình tạo các Stored procedure trong database Northwind của SQL Server bằng cách thực thi các lệnh SQL chuẩn thông qua các class trong namespace System.Data.SqlClient.
- Thực hiện một câu truy vấn (query) có trong database Northwind phiên bản Microsoft Access. Sử dụng các class trong namespace System.Data.OleDb.
- Thực hiện một stored procedure có yêu cầu tham số truyền vào và đưa tập kết quả vào một DataGrid.
- Thực hiện một stored procedure có yêu cầu tham số truyền vào, có kết quả xuất ra và có giá trị trả về.
- So sánh cách sử dụng DataSet và DataReader để đưa dữ liệu vào một ComboBox.

Project: **Su dung DataReader truy xuat va xu ly du lieu**

Category.vb

```
' Category là class tự tạo được sử dụng để kết dữ liệu vào một ComboBox.
' Chú ý các thuộc tính để kết dữ liệu phải ReadOnly.
Class Category
    Dim _ID As Integer
    Dim _Name As String

    Sub New(ByVal intID As Integer, ByVal strName As String)
        _ID = intID
        _Name = strName
    End Sub

    Public ReadOnly Property ID() As Integer
        Get
            Return _ID
        End Get
    End Property

    Public ReadOnly Property Name() As String
        Get
            Return _Name
        End Get
    End Property
End Class
```

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.Data.OleDb
Imports System.Text

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const ACCESS_CONNECTION_STRING As String = _
    "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=..\Northwind.mdb"

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL " & _
        "or MSDE with the Northwind database installed."

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    Private DidPreviouslyConnect As Boolean = False
    Private HasCreatedSprocs As Boolean = False
    Private sdr As SqlDataReader
    Private strConn As String

' Code phát sinh form được bỏ qua

    ' Xử lý nút Create Sprocs. Sử dụng các class trong namespace
    ' System.Data.SqlClient để thực thi các lệnh SQL, các lệnh này xóa các stored
    ' procedure (nếu nó đã tồn tại) và tạo lại chúng.
    Private Sub btnCreateSprocs_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCreateSprocs.Click

        strConn = SQL_CONNECTION_STRING
```

```
Dim frmStatusMessage As New frmStatus()
If Not DidPreviouslyConnect Then
  frmStatusMessage.Show("Connecting to SQL Server")
End If

Dim IsConnecting As Boolean = True
While IsConnecting
  Try

    Dim scnnNorthwind As New SqlConnection(strConn)

      ' Lệnh này xóa stored procedure "GetProducts" nếu nó đã tồn tại
    Dim strSQL As String = _
      "IF EXISTS (" & _
      "SELECT * " & _
      "FROM northwind.dbo.sysobjects " & _
      "WHERE Name = 'GetProducts' " & _
      "AND TYPE = 'p')" & vbCrLf & _
      "DROP PROCEDURE GetProducts"

    Dim scmd As New SqlCommand(strSQL, scnnNorthwind)
    Try
        ' Mở kết nối
      scnnNorthwind.Open()
        ' Kết nối thành công do đó thoát vòng lặp
      IsConnecting = False
      DidPreviouslyConnect = True
      frmStatusMessage.Close()

        ' Thực hiệnn lệnh SQL
      scmd.ExecuteNonQuery()

  Catch expSql As SqlException
      MsgBox(expSql.ToString, MsgBoxStyle.Critical, Me.Text)
      frmStatusMessage.Close()
      Exit Sub
    End Try
```

```
    ' Tạo stored procedure "GetProducts"
  scmd.CommandText = _
    "CREATE PROCEDURE GetProducts " & vbCrLf & _
    "@CategoryID Int " & vbCrLf & _
    "AS " & vbCrLf & _
    "SELECT ProductID, ProductName, UnitPrice, UnitsInStock " & _
    "FROM Northwind.dbo.Products " & _
    "WHERE CategoryID = @CategoryID"

  Try
    scmd.ExecuteNonQuery()

  Catch expSql As SqlException
    MsgBox(expSql.ToString, MsgBoxStyle.Critical, Me.Text)
    frmStatusMessage.Close()
    Exit Sub

  End Try

    ' Xóa stored procedure "GetProductCountAndAveragePrice"
  scmd.CommandText = _
    "IF EXISTS (" & _
    "SELECT * " & _
    "FROM northwind.dbo.sysobjects " & _
    "WHERE Name = 'GetProductCountAndAveragePrice' " & _
    "AND TYPE = 'p')" & vbCrLf & _
    "DROP PROCEDURE GetProductCountAndAveragePrice"

  Try
    scmd.ExecuteNonQuery()

  Catch expSql As SqlException
    MsgBox(expSql.ToString, MsgBoxStyle.Critical, Me.Text)
    frmStatusMessage.Close()
    Exit Sub

  End Try
```

```
            ' Tạo stored procedure "GetProductCountAndAveragePrice"
          scmd.CommandText = _
            "CREATE PROCEDURE GetProductCountAndAveragePrice " & _
            vbCrLf & _
            "@CategoryID Int, " & vbCrLf & _
            "@AveragePrice Int OUT" & vbCrLf & _
            "AS " & vbCrLf & _
            "DECLARE @SumProdPrices Money " & vbCrLf & _
            "SELECT @AveragePrice = SUM(UnitPrice)/COUNT(ProductID) " & _
            "FROM Northwind.dbo.Products " & _
            "WHERE CategoryID = @CategoryID" & vbCrLf & _
            "RETURN " & _
            "(SELECT COUNT(ProductID) " & _
            "FROM Northwind.dbo.Products " & _
            "WHERE CategoryID = @CategoryID)"
          Try
            scmd.ExecuteNonQuery()
          Catch expSql As SqlException
            MsgBox(expSql.ToString, MsgBoxStyle.Critical, Me.Text)
            frmStatusMessage.Close()
            Exit Sub
          Finally
              ' Đảm bảo kết nối luôn được đóng lại cho dù có lỗi xảy ra
            scnnNorthwind.Close()
          End Try

        Catch exp As Exception
          If strConn = SQL_CONNECTION_STRING Then
            strConn = MSDE_CONNECTION_STRING
            frmStatusMessage.Show("Connecting to MSDE")
          Else
            frmStatusMessage.Close()
            MsgBox("To run this sample, you must have SQL " & _
              "or MSDE with the Northwind database installed.", _
              MsgBoxStyle.Critical, Me.Text)
            End
          End If
        End Try
      End While
```

```
    frmStatusMessage.Close()

    MsgBox("The stored procedures were successfully added to the " & _
      "Northwind database.", MsgBoxStyle.Information, Me.Text)

    HasCreatedSprocs = True
  End Sub

  ' Xử lý cho nút " Top Ten Products " trên tab No Params. Sử dụng database
  ' Northwind phiên bản Access. Thủ tục này gọi một câu truy vấn (query) không
  ' có tham số.
  Private Sub btnGet10MostExpProds_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnGet10MostExpProds.Click

    Dim ocnnNorthwind As New
OleDbConnection(ACCESS_CONNECTION_STRING)
    Dim ocmd As New OleDbCommand("[Ten Most Expensive Products]",
ocnnNorthwind)
    Dim odr As OleDbDataReader

    ocmd.CommandType = CommandType.StoredProcedure

    ' Mở kết nối và sau đó thực hiện lệnh, ta cũng truyền vào một giá trị để đóng
    ' kết nối khi hoàn tất hoặc có thể dùng ocnnNorthwind.Close() để đóng kết nối
    ocnnNorthwind.Open()

    Try
      odr = ocmd.ExecuteReader(CommandBehavior.CloseConnection)
    Catch exp As Exception
      MsgBox(exp.ToString, MsgBoxStyle.Critical, Me.Text)
    End Try

    ' Sử dụng StringBuilder để ghép các chuỗi được hiển thị trong TextBox.
    ' Class này được tối ưu việc xử lý chuỗi và được ưa thích hơn so với cách cũ &=
    Dim sb As New StringBuilder()
    sb.Append("Product Name")
    sb.Append(vbTab)
    sb.Append(vbTab)
    sb.Append(vbTab)
    sb.Append("Price")
```

```
        sb.Append(vbCrLf)
        sb.Append("========================")
        sb.Append(vbTab)
        sb.Append("==========")
        sb.Append(vbCrLf)

        ' Duyệt qua nội dung của đối tượng OleDbDataReader
        While odr.Read
            sb.Append(odr.GetString(0))
            sb.Append(vbTab)
            sb.Append(vbTab)

            If odr.GetString(0).Length < 20 Then
                sb.Append(vbTab)
            End If

            sb.Append(odr.GetDecimal(1).ToString("c"))
            sb.Append(vbCrLf)
        End While

        ' Đóng DataReader
        odr.Close()

        ' Hiển thị kết quả
        txtTenMostExpProds.Text = sb.ToString
    End Sub

    ' Thủ tục này xử lý nút "Get Product Count and Avg Price" trên tab "All Types".
    ' Stored procedure ở đây yêu cầu một tham số truyền vào và trả về một kết quả và
    ' một giá trị trả về. Sau khi stored procedure được thực hiện, kết quả được hiển
    ' thị trên một Label.
    Private Sub btnGetProductCountAndAvgPrice_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    btnGetProductCountAndAvgPrice.Click
        strConn = SQL_CONNECTION_STRING

        Dim scnnNorthwind As New SqlConnection(strConn)
        Dim scmd As New SqlCommand("GetProductCountAndAveragePrice",
    scnnNorthwind)
        Dim sda As New SqlDataAdapter(scmd)
```

```
    scmd.CommandType = CommandType.StoredProcedure

     ' Thêm tham số cho stored procedure. Quy định giá trị của tham số là giá trị
     ' được chọn trong ComboBox. Quy định Direction cho tham số output. Cuối cùng,
     ' thêm một tham số để lấy giá trị trả về: Giá trị trả về từ stored procedure bằng
     ' lệnh RETURN, nếu không có, SQL Server trả về mã thực hiện thành công hoặc
     ' có lỗi.
    With scmd.Parameters
      .Add(New SqlParameter("@CategoryID", SqlDbType.Int)).Value = _
        cboCategoriesAllTypes.SelectedIndex + 1
  ' Mặc nhiên giá trị của Direction là "Input", chỉ cần thay đổi cho hai tham số kế tiếp
      .Add(New SqlParameter("@AveragePrice", _
        SqlDbType.Money)).Direction = ParameterDirection.Output
      .Add(New SqlParameter("ReturnValue", _
        SqlDbType.Int)).Direction = ParameterDirection.ReturnValue
    End With

    Try
      scnnNorthwind.Open()
      scmd.ExecuteNonQuery()

    Catch expSQL As SqlException
      MsgBox(expSQL.ToString, MsgBoxStyle.Critical, Me.Text)
      Exit Sub
    Finally
      scnnNorthwind.Close()
    End Try

     ' Hiển thị kết quả
    lblProductCountAndAvgPrice.Text = "There are " & _
      scmd.Parameters("ReturnValue").Value.ToString & " products " & _
      "in the " & cboCategoriesAllTypes.Text & " category, at an average " & _
      "price of " & CType(scmd.Parameters("@AveragePrice").Value, _
        Double).ToString("c")

  End Sub
```

```
' Xử lý cho nút "Get Products" trên tab "Input Param". Stored procedure
' "Get Products" yêu cầu một tham số truyền vào. SP này được thực hiện và
' một DataGrid sử dụng để hiển thị kết quả.
Private Sub btnGetProducts_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetProducts.Click

    strConn = SQL_CONNECTION_STRING

    ' Xem chỉ dẫn trong frmMain_Load
    Dim scnnNorthwind As New SqlConnection(strConn)
    Dim scmd As New SqlCommand("GetProducts", scnnNorthwind)
    Dim sda As New SqlDataAdapter(scmd)
    Dim dsProducts As New DataSet()

    ' Thêm các tham số được yêu cầu cho SP. Đây là cách dài dòng để thêm một
    ' SqlParameter và quy định các thuộc tính. Xem thủ tục
    ' btnGetProductCountAndAvgPrice_Click để thấy cách ngắn hơn.
    Dim sparCatID As New SqlParameter()
    With sparCatID
        .ParameterName = "@CategoryID"
        .SqlDbType = SqlDbType.Int
        .Value = cboCategoriesInputParam.SelectedValue
    End With

    With scmd
        .Parameters.Add(sparCatID)_
        .CommandType = CommandType.StoredProcedure
    End With
    Try
        ' Đưa dữ liệu vào DataSet
        sda.Fill(dsProducts, "Products")
        ' Kết DataGrid vào bảng trong DataSet. Thao tác này sẽ hiển thị kết quả
        grdProducts.DataSource = dsProducts.Tables(0)

    Catch expSQL As SqlException
        MsgBox(expSQL.ToString, MsgBoxStyle.Critical, Me.Text)
        Exit Sub
    End Try
```

```
    ' Quy định trực tiếp các thuộc tính của DataGrid, nhưng chỉ các thuộc tính
    ' không thuộc các thuộc tính của DataGridTableStyle
    With grdProducts
        .BackColor = Color.GhostWhite
        .BackgroundColor = Color.Lavender
        .BorderStyle = BorderStyle.None
        .CaptionBackColor = Color.RoyalBlue
        .CaptionFont = New Font("Tahoma", 10.0!, FontStyle.Bold)
        .CaptionForeColor = Color.Bisque
        .CaptionText = "Northwind Products"
        .Font = New Font("Tahoma", 8.0!)
        .ParentRowsBackColor = Color.Lavender
        .ParentRowsForeColor = Color.MidnightBlue
        ' Xóa các đối tượng TableStyle đã tạo trước đó, nếu không sẽ gây ra lỗi
        .TableStyles.Clear()
    End With

    ' Quy định các định dạng
    Dim grdTableStyle1 As New DataGridTableStyle()
    With grdTableStyle1
        .AlternatingBackColor = Color.GhostWhite
        .BackColor = Color.GhostWhite
        .ForeColor = Color.MidnightBlue
        .GridLineColor = Color.RoyalBlue
        .HeaderBackColor = Color.MidnightBlue
        .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
        .HeaderForeColor = Color.Lavender
        .SelectionBackColor = Color.Teal
        .SelectionForeColor = Color.PaleGreen

        ' Đừng quên quy định thuộc tính MappingName
        .MappingName = dsProducts.Tables(0).TableName
        .PreferredColumnWidth = 125
        .PreferredRowHeight = 15
    End With
```

```
    ' Định dạng các cột mà ta muốn xuất hiện trên DataGrid.
    Dim grdColStyle1 As New DataGridTextBoxColumn()
    With grdColStyle1
        .HeaderText = "ID"
        .MappingName = "ProductID"
        .Width = 50
    End With

    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
        .HeaderText = "Name"
        .MappingName = "ProductName"
    End With

    Dim grdColStyle3 As New DataGridTextBoxColumn()

    With grdColStyle3
        .HeaderText = "Price"
        .MappingName = "UnitPrice"
        .Format = "c"
        .Width = 75
        .ReadOnly = True
    End With

    Dim grdColStyle4 As New DataGridTextBoxColumn()
    With grdColStyle4
        .HeaderText = "# In Stock"
        .MappingName = "UnitsInStock"
        .Width = 75
        .Alignment = HorizontalAlignment.Center
    End With

    grdTableStyle1.GridColumnStyles.AddRange _
        (New DataGridColumnStyle() _
        {grdColStyle1, grdColStyle2, grdColStyle3, grdColStyle4})
    grdProducts.TableStyles.Add(grdTableStyle1)

End Sub
```

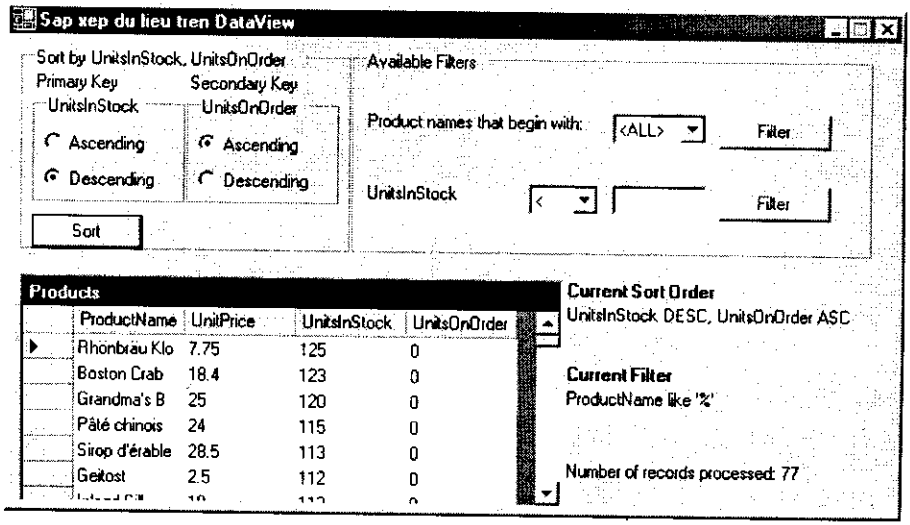
```
    ' Thủ tục này xử lý sự kiện SelectedIndexChanged của TabControl. Nó bảo đảm
    ' là người sử dụng không thể chạy các phần khác nếu chưa tạo các SP. Nếu đã
    ' tạo rồi, các lệnh SQL sẽ được thực hiện để đưa dữ liệu vào ComboBox Categories.
    Private Sub tabApp_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tabApp.SelectedIndexChanged
        If Not HasCreatedSprocs AndAlso tabApp.SelectedTab.TabIndex > 0 Then
            MsgBox("You must first create the required stored procedures.", _
                MsgBoxStyle.Exclamation, Me.Text)
            tabApp.SelectedIndex = 0

        ElseIf HasCreatedSprocs AndAlso cboCategoriesInputParam.Items.Count = 0
Then
            ' Đưa dữ liệu vào các ComboBox chỉ khi chúng trống và các SP đã được tạo
            strConn = SQL_CONNECTION_STRING

            Dim strSQL As String = _
                "SELECT CategoryID, CategoryName " & _
                "FROM Categories"

            Dim scnnNorthwind As New SqlConnection(strConn)
            Dim scmd As New SqlCommand(strSQL, scnnNorthwind)

            ' Sử dụng DATASET
            'Dim dsCategories As DataSet
            'Dim sda As New SqlDataAdapter(scmd)

            ' ArrayList này để sử dụng với cboCategoriesInputParam.
            ' Ghi chú dòng này nếu sử dụng DATASET.
            Dim arlCategories As New ArrayList()

            ' Sử dụng DATASET
            'dsCategories = New DataSet()

            Try
                ' Sử dụng DATASET
                'sda.Fill(dsCategories)
                '
                'With cboCategoriesInputParam
                '    .DataSource = dsCategories.Tables(0)
```

```
        '    .DisplayMember = "CategoryName"
        '    .ValueMember = "CategoryID"
        'End With
        '
        'With cboCategoriesAllTypes
        '    .DataSource = dsCategories.Tables(0)
        '    .DisplayMember = "CategoryName"
        '    .ValueMember = "CategoryID"
        'End With
        '

        scnnNorthwind.Open()
        sdr = scmd.ExecuteReader(CommandBehavior.CloseConnection)

        ' Duyệt qua tập hợp các phần tử của DataReader. Phương thức Read trả
        ' về giá trị True nếu như còn các dòng khác để đọc.
        While sdr.Read
          AddCategoryForDataSource(arlCategories)
          AddTextForSimpleFill()
        End While

        ' Thực hiện kết dữ liệu thực sự
        With cboCategoriesInputParam
          .DataSource = arlCategories
          .ValueMember = "ID"
          .DisplayMember = "Name"
        End With

        ' Đóng DataReader
        sdr.Close()

        ' Nếu không gán giá trị này, ComboBox sẽ trống
        cboCategoriesAllTypes.SelectedIndex = 0

      Catch expSql As SqlException
        MsgBox(expSql.ToString, MsgBoxStyle.Critical, Me.Text)
      End Try
    End If
  End Sub
```

```
    ' Thủ tục này được sử dụng bên trong vòng lặp While...End While duyệt qua
    ' DataReader. Nó đưa dữ liệu vào ArrayLists là các instance của class Category,
    ' giá trị được lấy từ DataReader.
    Private Sub AddCategoryForDataSource(ByRef arl As ArrayList)
        With arl
            .Add(New Category(sdr.GetInt32(0), sdr.GetString(1)))
        End With
    End Sub

    ' Thủ tục này được sử dụng bên trong vòng lặp While...End While duyệt qua
    ' DataReader. Nó đưa dữ liệu vào Items Collection của ComboBox,
    ' giá trị được lấy từ DataReader.
    Private Sub AddTextForSimpleFill()
        ' Duyệt qua Items collection của DataReader. Phương thức Read trả về True
        ' nếu còn các dòng khác để đọc.
        With cboCategoriesAllTypes
            .Items.Add(sdr("CategoryName"))
        End With
    End Sub
End Class
```

53. Sắp xếp và lọc dữ liệu trên DataView

Chương trình chỉ cách sử dụng một dataview để sắp xếp và lọc dữ liệu của bảng Products trong database Northwind.



Các chức năng chính:

- Khi form chính được load, bảng Products được truy xuất và đưa vào một DataSet bằng một lệnh SQL. Sau đó, một DataView được xây dựng để lọc và sắp xếp dữ liệu. Người sử dụng có thể chọn điều kiện lọc và điều kiện sắp xếp trên dataview mà không ảnh hưởng đến dữ liệu nguồn. Các dòng có thể được lọc bằng cách chọn các Product có ký tự đầu tiên do người sử dụng chọn. Các dòng còn có thể được sắp xếp theo hai cấp, mỗi cấp có thể theo thứ tự tăng hoặc giảm.

Project: **Sap xep va Loc du lieu tren DataView**

frmMain.vb

```
Option Strict On
Imports System.data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Friend WithEvents grdProducts As System.Windows.Forms.DataGrid
    Friend WithEvents rbtUnitsInStockA As System.Windows.Forms.RadioButton
    Friend WithEvents rbtUnitsInStockD As System.Windows.Forms.RadioButton
    Friend WithEvents rbtUnitsOnOrderA As System.Windows.Forms.RadioButton
    Friend WithEvents rbtUnitsOnOrderD As System.Windows.Forms.RadioButton
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents cboProducts As System.Windows.Forms.ComboBox
    Friend WithEvents cboCompare As System.Windows.Forms.ComboBox
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents txtUnits As System.Windows.Forms.TextBox
    Friend WithEvents cmdUnitsInStock As System.Windows.Forms.Button
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents cmdProductNames As System.Windows.Forms.Button
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents lblFilter As System.Windows.Forms.Label
    Friend WithEvents lblSort As System.Windows.Forms.Label
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents Label5 As System.Windows.Forms.Label
    Protected dsProducts As New DataSet()

    Protected dvProducts As DataView
```

```
    Protected Const SQL_CONNECTION_STRING As String = _
      "Server=localhost;" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
      "Server=(local)\NetSDK;" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI"

    Protected Const PRODUCT_TABLE_NAME As String = "Products"
    Protected Const STATUS_MESSAGE As String = "Number of records processed: "
    Protected Const CAPTION_MESSAGE_ORIG As String = "Products"
    Protected Const NO_RECORDS_FOUND_MESSAGE As String = _
                    "No records were found that match the filter criteria."
    Protected Const CAPTION_TITLE As String = "Sort and Filter with a DataView"
    Protected Const DEFAULT_FILTER As String = "ProductName like '%'"
    Protected Const DEFAULT_SORT As String = _
                    "UnitsInStock ASC, UnitsOnOrder ASC"
    Protected Const NO_RECORDS_TO_SORT_MESSAGE As String = _
                    "There are no records to sort."
    Protected Const CAPTION_ICON_BUTTON As MsgBoxStyle = _
          CType(MsgBoxStyle.Information + MsgBoxStyle.OKOnly, MsgBoxStyle)

' Code phát sinh form được bỏ qua

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load

      Dim strConnection As String = SQL_CONNECTION_STRING
      Dim frmStatusMessage As New frmStatus()
      frmStatusMessage.Show("Connecting to SQL Server")

      Dim IsConnecting As Boolean = True
      While IsConnecting
        Try

          Dim northwindConnection As New SqlConnection(strConnection)
```

```
        ' SqlDataAdapter được sử dụng để thiết lập một DataSet
        Dim ProductAdapter As New SqlDataAdapter( _
          "SELECT ProductName, UnitPrice, UnitsInStock, UnitsOnOrder " _
          & "FROM products", _
          northwindConnection)

        ' Thiết lập DataSet với thông tin trong bảng Products.
        ProductAdapter.Fill(dsProducts, PRODUCT_TABLE_NAME)

        ' Tạo DataView; sử dụng bộ khởi tạo để chỉ định kiểu sắp xếp, điều kiện lọc
        dvProducts = New DataView(dsProducts.Tables("products"), _
                                  DEFAULT_FILTER, DEFAULT_SORT, _
                                  DataViewRowState.OriginalRows)

        ' Dữ liệu đã truy xuất thành công, gán tín hiệu thoát vòng lặp
        IsConnecting = False

        ' Xử lý khi kết nối có lỗi
      Catch exc As Exception
        If strConnection = SQL_CONNECTION_STRING Then
          strConnection = MSDE_CONNECTION_STRING
          frmStatusMessage.Show("Connecting to MSDE")
        Else
          frmStatusMessage.Close()
          MsgBox("To run this sample, you must have SQL " & _
          "or MSDE with the Northwind database installed.", _
           CAPTION_ICON_BUTTON, CAPTION_TITLE)

          End

        End If
      End Try
    End While

    frmStatusMessage.Close()

     ' Kết DataGrid vào DataView đã tạo ở trên
    grdProducts.DataSource = dvProducts
```

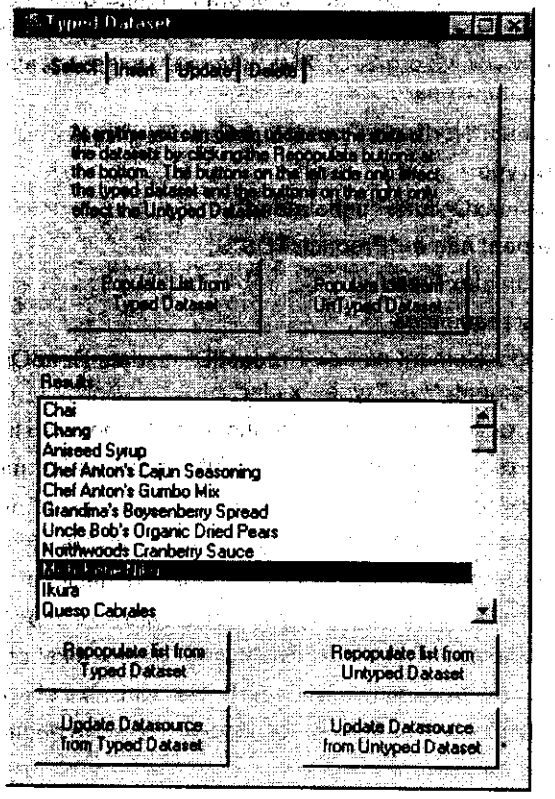
```
    ' Thiết lập ComboBox để lọc ProductName.
    ' Cho phép người sử dụng chọn ký tự đầu tiên của các sản phẩm muốn xem
    cboProducts.Items.AddRange(New Object() {"<ALL>", "A", "B", "C", "D", "E", _
      "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", _
      "W", "X", "Y", "Z"})
    cboProducts.Text = "<ALL>"

    ' Thiết lập ComboBox để lọc trên UnitsInStock
    ' Cho phép sử dụng các phép so sánh thông thường
    cboCompare.Items.AddRange(New Object() {"<", "<=", "=", ">=", ">"})
    cboCompare.Text = "<"

    ' Hiển thị thông tin trạng thái
    lblRecords.Text = STATUS_MESSAGE & _
        dsProducts.Tables(PRODUCT_TABLE_NAME).Rows.Count.ToString
    grdProducts.CaptionText = CAPTION_MESSAGE_ORIG
    lblSort.Text = DEFAULT_SORT
    lblFilter.Text = DEFAULT_FILTER
End Sub

Private Sub cmdProductNames_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdProductNames.Click
    Dim strFilter As String
    ' Xử lý lọc các dòng dựa trên ký tự đầu tiên được chọn
    ' Nếu là <All> được chọn, hiển thị tất cả các dòng, nếu  không phải thì chỉ hiển
    ' thị các dòng có bắt đầu bằng ký tự được chọn.
    If cboProducts.Text = "<ALL>" Then
      strFilter = "ProductName like '%'"
    Else
      strFilter = "ProductName like '" & cboProducts.Text & "%'"
    End If
    dvProducts.RowFilter = strFilter

    ' Hiển thị DataGrid đã được sắp xếp và được lọc
    grdProducts.DataSource = dvProducts

    ' Hiển thị số dòng có trên DataGrid
    lblRecords.Text = STATUS_MESSAGE & dvProducts.Count.ToString
    lblFilter.Text = strFilter
```

```
    ' Hiển thị một thông báo nếu không tìm thấy record nào
    If dvProducts.Count = 0 Then
        MsgBox(NO_RECORDS_FOUND_MESSAGE, _
                    CAPTION_ICON_BUTTON, CAPTION_TITLE)
    End If
End Sub

Private Sub cmdUnitsInStock_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUnitsInStock.Click
    Dim strFilter As String
    ' Xử lý lọc các record dựa trên số UnitsInStock
    ' Sử dụng số được nhập vào và phép toán so sánh được chọn để lọc

    ' Kiểm tra giá trị nhập vào
    If txtUnits.Text.Trim.Length = 0 Then
        MsgBox(NO_RECORDS_FOUND_MESSAGE, _
                CAPTION_ICON_BUTTON, CAPTION_TITLE)
        Exit Sub
    ElseIf Not IsNumeric(txtUnits.Text.Trim.ToString) Then
        MsgBox(NO_RECORDS_FOUND_MESSAGE, _
                CAPTION_ICON_BUTTON, CAPTION_TITLE)
        Exit Sub
    End If

    strFilter = "UnitsInStock " & cboCompare.Text & " " & txtUnits.Text
    ' Lọc theo UnitsInStock
    dvProducts.RowFilter = strFilter
    ' Hiển thị dữ liệu đã được xử lý
    grdProducts.DataSource = dvProducts
    ' Hiển thị số dòng có trong DataView
    lblRecords.Text = STATUS_MESSAGE & dvProducts.Count.ToString
    lblFilter.Text = strFilter

    If dvProducts.Count = 0 Then
        MsgBox(NO_RECORDS_FOUND_MESSAGE, _
                CAPTION_ICON_BUTTON, CAPTION_TITLE)
    End If
End Sub
```

```
Private Sub cmdSort_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSort.Click

    ' Áp dụng điều kiện sắp xếp vào DataView

    Dim strSort As String

    ' Chỉ sắp xếp khi nào DataView có dữ liệu
    If dvProducts.Count = 0 Then
        MsgBox(NO_RECORDS_TO_SORT_MESSAGE, _
                    CAPTION_ICON_BUTTON, CAPTION_TITLE)
        Exit Sub
    End If

    ' Xử lý các điều kiện sắp xếp được chọn cho DataView
    ' Xây dựng chuỗi điều kiện sắp xếp theo hai cột.
    ' Cột chính là cột UnitsInStock, cột thứ hai là cột UnitsOnOrder
    If rbtUnitsInStockA.Checked = True Then
        strSort = "UnitsInStock ASC"
    Else
        strSort = "UnitsInStock DESC"
    End If

    If rbtUnitsOnOrderA.Checked = True Then
        strSort = strSort & ", UnitsOnOrder ASC"
    Else
        strSort = strSort & ", UnitsOnOrder DESC"
    End If

    ' Áp dụng điều kiện sắp xếp vào DataView
    dvProducts.Sort = strSort

    ' Hiển thị dữ liệu lên DataGrid
    grdProducts.DataSource = dvProducts
    lblRecords.Text = STATUS_MESSAGE & dvProducts.Count.ToString
    lblSort.Text = strSort
End Sub

End Class
```

54. DataSet có cấu trúc trước (Typed DataSet)

Chương trình này cho thấy sự khác nhau giữa untyped dataset (dataset không được cấu trúc trước) và typed dataset khi thực hiện thêm, cập nhật, xóa và truy vấn dữ liệu. Một typed dataset cho người lập trình khả năng tham chiếu đến các bảng và các trường bằng cách sử dụng tính năng IntelliSense vào lúc thiết kế, lập trình. Điều này cho phép phát hiện lỗi vào lúc thiết kế thay vì lúc chạy.



Các chức năng chính:

- Tương phản giữa Insert với một Typed Dataset và với một Untyped Dataset.
- Tương phản giữa Update với một Typed Dataset và với một Untyped Dataset.
- Tương phản giữa Delete với một Typed Dataset và với một Untyped Dataset.
- Tương phản giữa Truy vấn với một Typed Dataset và với một Untyped Dataset.
- Đưa các thay đổi trong các dataset vào database.

Project: **Typed DataSet**

Northwind.xsd

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema id="Nothwind" targetNamespace="http://tempuri.org/dsMain.xsd"
  elementFormDefault="qualified" attributeFormDefault="qualified"
  xmlns="http://tempuri.org/dsMain.xsd" xmlns:mstns="http://tempuri.org/dsMain.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
  microsoft-com:xml-msdata">
   <xs:element name="Northwind" msdata:IsDataSet="true">
     <xs:complexType>
       <xs:choice maxOccurs="unbounded">
         <xs:element name="ProductsTDS">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="ProductID" msdata:ReadOnly="true"
  msdata:AutoIncrement="true" type="xs:int" />
               <xs:element name="ProductName" type="xs:string" />
               <xs:element name="SupplierID" type="xs:int" minOccurs="0" />
               <xs:element name="CategoryID" type="xs:int" minOccurs="0" />
             <xs:element name="QuantityPerUnit" type="xs:string" minOccurs="0" />
              <xs:element name="UnitPrice" type="xs:decimal" minOccurs="0" />
               <xs:element name="UnitsInStock" type="xs:short" minOccurs="0" />
               <xs:element name="UnitsOnOrder" type="xs:short" minOccurs="0" />
               <xs:element name="ReorderLevel" type="xs:short" minOccurs="0" />
               <xs:element name="Discontinued" type="xs:boolean" />
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:choice>
     </xs:complexType>
     <xs:unique name="NorthwindKey1" msdata:PrimaryKey="true">
       <xs:selector xpath=".//mstns:ProductsTDS" />
       <xs:field xpath="mstns:ProductID" />
     </xs:unique>
   </xs:element>
</xs:schema>
```

frmMain.vb

```
Option Strict On
Imports System.Data
Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI;Connect Timeout=5"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NETSDK;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI;Connect Timeout=5"

    Private ConnectionString As String = SQL_CONNECTION_STRING
    Private HasConnected As Boolean = False

    ' Khai báo một typed dataset
    Private tdsNorthwind As Northwind

    ' Khai báo một dataset thông thường
    Private dsNorthwind As DataSet

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

        InitDataSets()

    End Sub
```

```
' Code phát sinh form được bỏ qua

Private Sub btnDeleteDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeleteDS.Click
    ' Xóa dòng được chọn trong bảng Products, sử dụng typed DataSet
    If lstResults.SelectedIndex >= 0 Then
        dsNorthwind.Tables("ProductsDS").Rows(lstResults.SelectedIndex).Delete()
    Else
        MsgBox("No item selected.", MsgBoxStyle.Information, Me.Text)
    End If
    PopulateListFromDS()
End Sub

Private Sub btnDeleteTDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeleteTDS.Click
    ' Xóa dòng được chọn trong bảng Products
    If lstResults.SelectedIndex >= 0 Then
        tdsNorthwind.ProductsTDS(lstResults.SelectedIndex).Delete()
    Else
        MsgBox("No item selected.", MsgBoxStyle.Information, Me.Text)
    End If
    PopulateListFromTDS()
End Sub

Private Sub btnInsertTDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnInsertTDS.Click
    ' Insert một dòng mới vào typed dataset và thiết lập các giá trị cho nó
    Dim Row As Northwind.ProductsTDSRow

    Row = tdsNorthwind.ProductsTDS.NewProductsTDSRow()

    Row.ProductName = "Typed Dataset Inserted Record"
    Row.Discontinued = True
    Row.QuantityPerUnit = ""
    Row.UnitPrice = 50
    Row.UnitsInStock = 20
    Row.UnitsOnOrder = 0
    Row.ReorderLevel = 25

    tdsNorthwind.ProductsTDS.AddProductsTDSRow(Row)
```

```
    PopulateListFromTDS()
    lstResults.SelectedIndex = lstResults.Items.Count - 1

End Sub

Private Sub btnInsertDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnInsertDS.Click
    ' Insert một dòng mới vào untyped dataset và thiết lập các giá trị cho nó
    Dim Row As DataRow

    Row = dsNorthwind.Tables("ProductsDS").NewRow()

    Row("ProductName") = "Typed Dataset Inserted Record"
    Row("Discontinued") = True
    Row("QuantityPerUnit") = ""
    Row("UnitPrice") = 50
    Row("UnitsInStock") = 20
    Row("UnitsOnOrder") = 0
    Row("ReorderLevel") = 25

    dsNorthwind.Tables("ProductsDS").Rows.Add(Row)

    PopulateListFromDS()

    lstResults.SelectedIndex = lstResults.Items.Count - 1
End Sub

Private Sub btnSelectDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSelectDS.Click
    PopulateListFromDS()
End Sub

Private Sub btnSelectTDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSelectTDS.Click
    PopulateListFromTDS()
End Sub
```

```
Private Sub btnUpdateDataDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdateDataDS.Click
    ' Cập nhật vào dữ liệu nguồn với các thay đổi đã được thực hiện
    ' trên untyped dataset

  Try
      ' Mở kết nối
    Dim con As New SqlConnection(ConnectionString)
    con.Open()

      ' Tạo một DataAdapter
    Dim daProductsDS As New SqlDataAdapter("Select * from ProductsDS", con)

    Dim oCommandBuilder As New SqlCommandBuilder(daProductsDS)

    daProductsDS.DeleteCommand = oCommandBuilder.GetDeleteCommand
    daProductsDS.InsertCommand = oCommandBuilder.GetInsertCommand
    daProductsDS.UpdateCommand = oCommandBuilder.GetUpdateCommand

      ' Gọi phương thức Update để cập nhật dữ liệu
    daProductsDS.Update(dsNorthwind, "ProductsDS")
  Catch exp As Exception
    MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
  End Try
End Sub

Private Sub btnUpdateDataTDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdateDataTDS.Click
    ' Cập nhật dữ liệu nguồn với các thay đổi đã thực hiện trên typed dataset

  Try
      ' Mở kết nối
    Dim con As New SqlConnection(ConnectionString)
    con.Open()

      ' Tạo DataAdapter
    Dim daProductsTDS As New SqlDataAdapter( _
                          "Select * from ProductsTDS", con)
    Dim oCommandBuilder As New SqlCommandBuilder(daProductsTDS)
    daProductsTDS.DeleteCommand = oCommandBuilder.GetDeleteCommand
```

```
        daProductsTDS.InsertCommand = oCommandBuilder.GetInsertCommand
        daProductsTDS.UpdateCommand = oCommandBuilder.GetUpdateCommand

        ' Thực hiện cập nhật
        daProductsTDS.Update(tdsNorthwind, "ProductsTDS")
      Catch exp As Exception
        MsgBox(exp.Message, MsgBoxStyle.Critical, Me.Text)
      End Try
    End Sub

    Private Sub btnUpdateDS_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnUpdateDS.Click
      ' Cập nhật record được chọn trong bảng Products
      If lstResults.SelectedIndex >= 0 Then
        Dim ProductName As String = _
            CStr(dsNorthwind.Tables("ProductsDS").Rows( _
                                  lstResults.SelectedIndex)( "ProductName"))
        dsNorthwind.Tables("ProductsDS").Rows(lstResults.SelectedIndex)( _
                                    "ProductName") = ProductName & " - Updated"
      Else
        MsgBox("No item selected.", MsgBoxStyle.Information, Me.Text)
      End If

      PopulateListFromDS()
    End Sub

    Private Sub btnUpdateTDS_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnUpdateTDS.Click
      ' Cập nhật record được chọn trong bảng Products, sử dụng typed DataSet
      If lstResults.SelectedIndex >= 0 Then
        Dim ProductName As String = _
              tdsNorthwind.ProductsTDS(lstResults.SelectedIndex).ProductName
        tdsNorthwind.ProductsTDS(0).ProductName = ProductName & " - Updated"
      Else
        MsgBox("No item selected.", MsgBoxStyle.Information, Me.Text)
      End If
      PopulateListFromTDS()
    End Sub
```

```
Private Sub lstUpdateFromDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lstUpdateFromDS.Click
    PopulateListFromDS()
End Sub

Private Sub lstUpdateFromTDS_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles lstUpdateFromTDS.Click
    PopulateListFromTDS()
End Sub

Private Sub InitDataSets()

    Dim frmStatusMessage As New frmStatus()
    If Not HasConnected Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim IsConnecting As Boolean = True
    While IsConnecting
        Try
            Dim con As New SqlConnection(ConnectionString)
            con.Open()

            IsConnecting = False
            HasConnected = True
            frmStatusMessage.Close()

            ' Vì đây là lần kết nối đầu tiên nên phải tạo một bảng cho chương trình này
            ' Sử dụng một Instance của SqlCommand để thực hiện các lệnh SQL
            Dim cmInitSQL As New SqlCommand()

            ' Kết Command vào SqlConnection
            cmInitSQL.Connection = con

            ' Quy định kiểu lệnh là Text
            cmInitSQL.CommandType = CommandType.Text
```

```
        ' Xóa bảng ProductsDS nếu đã tồn tại
        cmInitSQL.CommandText = "IF EXISTS (SELECT * FROM dbo.sysobjects
WHERE id = object_id(N'[dbo].[ProductsDS]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1) " & _
                        "DROP TABLE [dbo].[ProductsDS] "

        ' Thực hiện câu lệnh
        cmInitSQL.ExecuteNonQuery()

        ' Xóa bảng ProductsTDS nếu đã tồn tại
        cmInitSQL.CommandText = "IF EXISTS (SELECT * FROM dbo.sysobjects
WHERE id = object_id(N'[dbo].[ProductsTDS]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1) " & _
                        "DROP TABLE [dbo].[ProductsTDS] "

        ' Thực hiện câu lệnh
        cmInitSQL.ExecuteNonQuery()

        ' Tạo bảng ProductsDS
        cmInitSQL.CommandText = "CREATE TABLE [dbo].[ProductsDS] ( " & _
                        "[ProductID] [int] IDENTITY (1, 1) PRIMARY KEY ," & _
                        "[ProductName] [nvarchar] (40) NOT NULL , " & _
                        "[SupplierID] [int] NULL , " & _
                        "[CategoryID] [int] NULL , " & _
                        "[QuantityPerUnit] [nvarchar] (20) NULL , " & _
                        "[UnitPrice] [money] NULL , " & _
                        "[UnitsInStock] [smallint] NULL , " & _
                        "[UnitsOnOrder] [smallint] NULL , " & _
                        "[ReorderLevel] [smallint] NULL , " & _
                        "[Discontinued] [bit] NOT NULL )"

        ' Thực hiện câu lệnh
        cmInitSQL.ExecuteNonQuery()
```

```
    ' Tạo bảng ProductsTDS
cmInitSQL.CommandText = "CREATE TABLE [dbo].[ProductsTDS] ( " & _
                "[ProductID] [int] IDENTITY (1, 1) PRIMARY KEY ," & _
                "[ProductName] [nvarchar] (40) NOT NULL , " & _
                "[SupplierID] [int] NULL , " & _
                "[CategoryID] [int] NULL , " & _
                "[QuantityPerUnit] [nvarchar] (20) NULL , " & _
                "[UnitPrice] [money] NULL , " & _
                "[UnitsInStock] [smallint] NULL , " & _
                "[UnitsOnOrder] [smallint] NULL , " & _
                "[ReorderLevel] [smallint] NULL , " & _
                "[Discontinued] [bit] NOT NULL )"

    ' Thực hiện câu lệnh
cmInitSQL.ExecuteNonQuery()

    ' Insert dữ liệu vào bảng ProductsDS, dữ liệu lấy từ bảng Products
cmInitSQL.CommandText = "INSERT INTO ProductsDS " & _
                "(ProductName,SupplierID,CategoryID," & _
                "QuantityPerUnit,UnitPrice,UnitsInStock," & _
                "UnitsOnOrder,ReorderLevel, Discontinued) " & _
                "SELECT ProductName,SupplierID,CategoryID," & _
                "QuantityPerUnit,UnitPrice,UnitsInStock," & _
                "UnitsOnOrder,ReorderLevel, Discontinued FROM Products"

cmInitSQL.ExecuteNonQuery()

    ' Insert dữ liệu vào bảng bảng ProductsDS, dữ liệu lấy từ bảng Products
cmInitSQL.CommandText = "INSERT INTO ProductsTDS " & _
                "(ProductName,SupplierID,CategoryID," & _
                "QuantityPerUnit,UnitPrice,UnitsInStock," & _
                "UnitsOnOrder,ReorderLevel, Discontinued) " & _
                "SELECT ProductName,SupplierID,CategoryID," & _
                "QuantityPerUnit,UnitPrice,UnitsInStock," & _
                "UnitsOnOrder,ReorderLevel, Discontinued FROM Products"

cmInitSQL.ExecuteNonQuery()
cmInitSQL.Dispose()
```

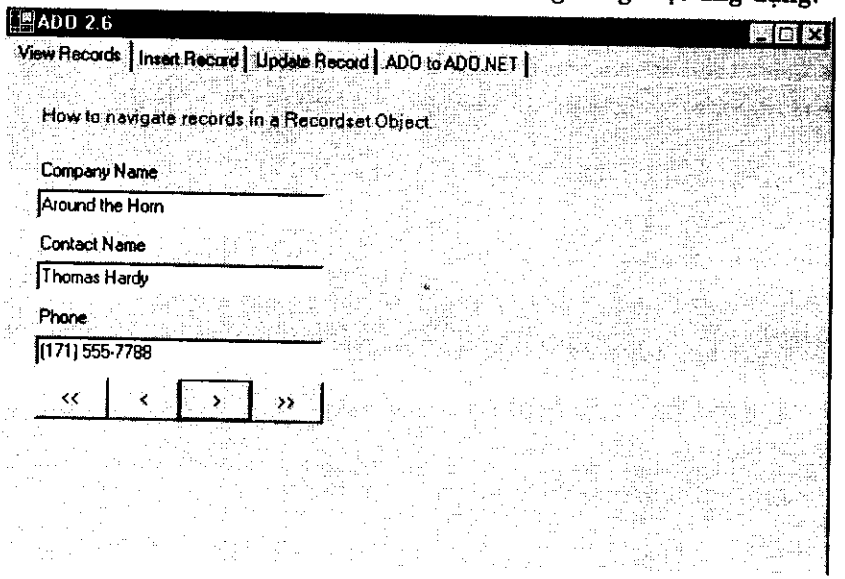
```
    ' Tạo các đối tượng Command để lấy dữ liệu đưa vào các DataSet
    Dim cmdProductsDS As New SqlCommand( _
                                "SELECT * FROM ProductsDS", con)
    Dim cmdProductsTDS As New SqlCommand( _
                                "SELECT * FROM ProductsTDS", con)

    ' Tạo mới các instance của dataset và typed dataset
    tdsNorthwind = New Northwind()
    dsNorthwind = New DataSet()

    ' Sử dụng SqlDataAdapter để đưa dữ liệu vào các dataset
    Dim daLocal As SqlDataAdapter = New SqlDataAdapter()

    ' Chọn câu lệnh SQL cho Typed Dataset
    daLocal.SelectCommand = cmdProductsTDS

    ' Đưa dữ liệu vào typed dataset
    daLocal.Fill(tdsNorthwind, "ProductsTDS")

    ' Chọn câu lệnh cho Untyped Dataset
    daLocal.SelectCommand = cmdProductsDS

    ' Đưa dữ liệu vào Dataset
    daLocal.Fill(dsNorthwind, "ProductsDS")

    con.Close()

Catch e As SqlException
    If ConnectionString = SQL_CONNECTION_STRING Then
        ConnectionString = MSDE_CONNECTION_STRING
        frmStatusMessage.Show("Connecting to MSDE")
    Else
        frmStatusMessage.Close()
        MsgBox("To run this sample you must have SQL Server ot MSDE.", _
                MsgBoxStyle.Critical, _
                "SQL Server/MSDE not found")
        End
    End If
```

```
        Catch e As Exception
            MsgBox(e.Message, MsgBoxStyle.Critical, "General Error")
        End Try
    End While
End Sub

Private Sub PopulateListFromDS()
    ' Một bảng trong một dataset được dùng để thiết lập một ListBox
    ' với ProductName lấy từ bảng Products.
    ' Với Dataset, tên của bảng và tên các cột phải được biết chính xác lúc thiết kế.
    ' Nếu chúng bị sai hoặc sai kiểu dữ liệu sẽ chỉ gây ra lỗi khi chạy chương trình.
    Dim s As String
    Dim i As Integer

    lstResults.Items.Clear()
    For i = 0 To dsNorthwind.Tables("ProductsDS").Rows.Count - 1
        ' Kiểm tra để biết dòng hiện hành có đánh dấu là deleted (đã xóa)
        If Not dsNorthwind.Tables("ProductsDS").Rows(i).RowState = _
                DataRowState.Deleted Then
            ' Lấy ProductName của mỗi record
            s = dsNorthwind.Tables("ProductsDS").Rows(i)("ProductName").ToString()
            ' Thêm ProductName vào ListBox
            lstResults.Items.Add(s)
        End If
    Next i
End Sub

Private Sub PopulateListFromTDS()

    ' Một bảng trong một typed dataset được sử dụng để thiết lập một ListBox
    ' với ProductName từ bảng Products.
    ' Một typed dataset cung cấp khả năng kiểm tra tính chính xác của tên bảng
    ' và tên các cột lúc thiết kế trước khi chương trình chạy.

    Dim s As String
    Dim i As Integer

    lstResults.Items.Clear()
```

```
        For i = 0 To tdsNorthwind.ProductsTDS.Count - 1

            ' Kiểm tra dòng hiện hành có được đánh dấu là deleted
          If Not tdsNorthwind.ProductsTDS(i).RowState = DataRowState.Deleted Then
            s = tdsNorthwind.ProductsTDS(i).ProductName.ToString()
            lstResults.Items.Add(s)
          End If
        Next i
    End Sub
End Class
```

55. Sử dụng ADO 2.6 trong .NET

Chương trình này chỉ cách sử dụng ADO 2.6 trong Visual Basic .NET, cũng như cách sử dụng ADO và ADO.NET trong cùng một ứng dụng.



Các chức năng chính:

- Cách sử dụng đối tượng Connection.
- Cách sử dụng đối tượng Command.
- Cách sử dụng đối tượng Recordset.
- Cách duyệt qua các record.
- Cách thực hiện Update (cập nhật) và Insert (thêm mới).
- Cách sử dụng kết hợp ADO và ADO .NET.

Project: **Su dung ADO 2.6**

frmMain.vb

```
Option Strict On
Imports ADODB

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const MSDE_CONNECTION_STRING As String = _
     "driver={SQL Server};server=(local)\VSDotNet;uid=sa;pwd=;database=Northwind"

    Protected Const SQL_CONNECTION_STRING As String = _
       "driver={SQL Server};server=localhost;uid=sa;pwd=;database=Northwind"

    Private ConnectionString As String = SQL_CONNECTION_STRING
    Private HasConnected As Boolean = False

    Dim cnn As New Connection()
    Dim cm As New Command()
    Dim rs As New Recordset()
    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

        PopulateSimpleNavigationForm()

    End Sub

' Code phát sinh form được bỏ qua

    Private Sub btnDataset_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnDataset.Click
        ' Đối tượng connection và recordset (ADO 2.6) được tạo để lấy dữ liệu bằng
        ' một câu lệnh SQL. Một DataAdapter sử dụng phương thức Fill để thiết lập
```

```
    ' DataSet với dữ liệu trong recordset (ADO 2.6) . Dữ liệu sau đó được kết vào
    ' control DataGrid.

    Dim strSQL As String = "SELECT CustomerID, " & _
                          "    CompanyName, " & _
                          "    ContactName, " & _
                          "    COuntry, " & _
                          "    Region, " & _
                          "    Phone, " & _
                          "    Fax " & _
                          "FROM Customers"

    rs = cnn.Execute(strSQL)

    ' Tạo các đối tượng DataSet và DataAdapter
    Dim ds As New DataSet("Recordset")
    Dim da As New OleDb.OleDbDataAdapter()

    ' Gọi phương thức Fill của DataAdapter để đưa dữ liệu từ ADO Recordset
    ' vào ADO.NET DataSet.
    da.Fill(ds, rs, "Customers")

    ' Kết DataSet vào một DataGrid
    dgMain.DataSource = ds
    dgMain.DataMember = "Customers"
End Sub

Private Sub btnFirst_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFirst.Click
    rs.MoveFirst()
    PopulateSimpleNavigationForm()
End Sub

Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLast.Click
    rs.MoveLast()
    PopulateSimpleNavigationForm()
End Sub
```

```
Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNext.Click
    ' Lệnh If này kiểm tra để tránh đi đến EOF
    If Not rs.EOF Then
      rs.MoveNext()
      If rs.EOF Then
        rs.MovePrevious()
      End If
      PopulateSimpleNavigationForm()
    End If
End Sub

Private Sub btnPrev_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrev.Click
    If Not rs.BOF Then
      rs.MovePrevious()

      If rs.BOF Then
        rs.MoveNext()
      End If
      PopulateSimpleNavigationForm()
    End If
End Sub

Private Sub btnInsert_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnInsert.Click
    ' Insert một record mới vào bảng Categories

    ' Kiểm tra dữ liệu trên form
    If txtCategoryName.Text = "" Or txtDescription.Text = "" Then
      MsgBox("Please fill in all the text boxes.", _
          MsgBoxStyle.Critical, Me.Text)
      Exit Sub
    End If

    Dim recordsEffected As Object
    Dim strSQL As String = "INSERT INTO Categories(CategoryName, " & _
                "Description) " & _
                "VALUES ('" & txtCategoryName.Text & _
                "','" & txtDescription.Text & "')"
```

```
        ' Thực hiện câu lệnh SQL
        cnn.Execute(strSQL, recordsEffected)

        If CInt(recordsEffected) = 1 Then
            MsgBox("Insert Successful!", MsgBoxStyle.Exclamation, Me.Text)
        End If

    End Sub

    Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnUpdate.Click
        ' Cập nhật field Description của bảng Categories với giá trị
        ' trong TextBox Description.

        If txtUpdateDescription.Text = "" Then
            MsgBox("Please fill in description text box.", _
                MsgBoxStyle.Critical, Me.Text)
            Exit Sub
        End If

        Dim strSQL As String = "UPDATE Categories SET Description = " & _
                               "'" & txtUpdateDescription.Text & "' " & _
                               "WHERE CategoryName = " & "'" & _
                               cbCategoryName.Text & "'"

        cm.ActiveConnection = cnn
        cm.CommandText = strSQL

        Dim recordsEffected As Object
        cm.Execute(recordsEffected)

        ' Kiểm tra xem có ít nhất một record có hiệu lực chưa
        If CInt(recordsEffected) > 0 Then
            MsgBox("Update Successful!", MsgBoxStyle.Exclamation, Me.Text)
        Else
            MsgBox("Update Failed!", MsgBoxStyle.Critical, Me.Text)
        End If
    End Sub
```

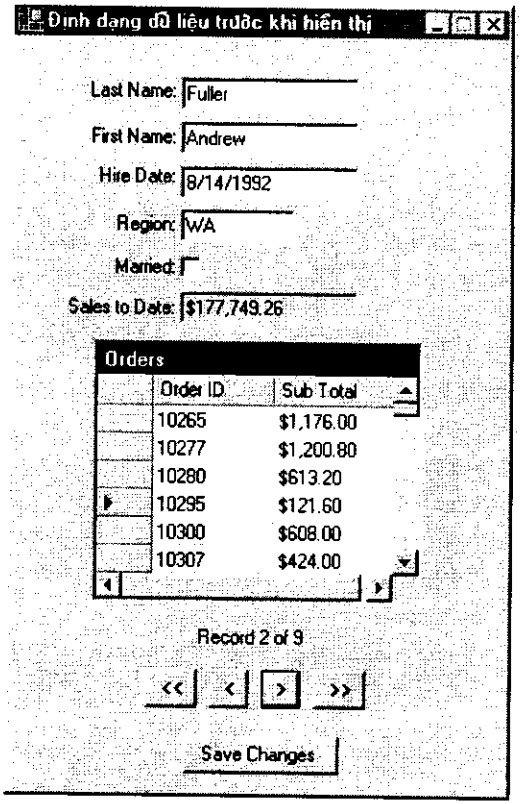
```
Private Sub cbCategoryName_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cbCategoryName.SelectedIndexChanged
    ' Khi một Category được chọn trên tab Update, Description được hiển thị
    ' trên TextBox để người sử dụng có thể thay đổi nó.
    Dim strSQL As String = "SELECT Description " & _
                "FROM Categories " & _
                "WHERE CategoryName = '" & cbCategoryName.Text & "'"

    rs.ActiveConnection = cnn
    rs.CursorType = CursorTypeEnum.adOpenStatic
    rs.Open(strSQL)

    ' Đưa nội dung vào TextBox
    txtUpdateDescription.Text = CStr(rs.Fields("Description").Value)

    rs.Close()
End Sub

Private Sub tcMain_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles tcMain.SelectedIndexChanged

    ' Đóng recordset nếu thay đổi tab
    If rs.State = 1 Then
        rs.Close()
    End If

    ' Gọi các thủ tục tương ứng khi chọn các tab.
    Select Case tcMain.SelectedIndex
        Case 0
            InitRecordNavigation()
        Case 2
            InitSimpleUpdate()
    End Select
End Sub

Private Sub InitRecordNavigation()
    ' Code cho thấy cách kết nối vào một database bằng ADO 2.6 và duyệt các record
    Dim frmStatusMessage As New frmStatus()
```

```
If Not HasConnected Then
  frmStatusMessage.Show("Connecting to SQL Server")
   ' Thực hiện kết nối vào SQL Server hoặc MSDE
  Dim IsConnecting As Boolean = True
  While IsConnecting
    Try
        ' Mở kết nối
      cnn.ConnectionTimeout = 5
      cnn.Open(ConnectionString)
      If cnn.State <> 1 Then
        Throw New System.Exception("Connection failed.")
      End If

      If Not HasConnected Then
        frmStatusMessage.Close()
      End If
      IsConnecting = False
      HasConnected = True

    Catch Exp As Exception
      If ConnectionString = SQL_CONNECTION_STRING Then
          ' Không thể kết nối vào SQL Server, chuyển qua MSDE
        ConnectionString = MSDE_CONNECTION_STRING
        frmStatusMessage.Show("Connecting to MSDE")
      Else
          ' Không tìm thấy SQL Server hoặc MSDE
        frmStatusMessage.Close()
        MsgBox("To run this sample you must have SQL Server ot MSDE.", _
            MsgBoxStyle.Critical, _
            "SQL Server/MSDE not found")
          ' Thoát chương trình nếu không có kết nối nào thành công
        End
      End If
    End Try
  End While

End If
```

```
        ' Tạo câu lệnh SQL
        Dim strSQL As String = "SELECT CompanyName, " & _
                               "      ContactName, " & _
                               "      Phone " & _
                               "FROM Customers"
        rs.ActiveConnection = cnn
        rs.CursorType = CursorTypeEnum.adOpenStatic
        rs.Open(strSQL)
        rs.MoveFirst()
    End Sub

    Private Sub InitSimpleUpdate()
        ' Thiết lập ComboBox chứa các Category
        Dim strSQL As String = "SELECT CategoryName " & _
                               "FROM Categories"

        rs.ActiveConnection = cnn
        rs.CursorType = CursorTypeEnum.adOpenStatic
        rs.Open(strSQL)

        ' Duyệt qua các record và cho chúng vào ComboBox
        While Not rs.EOF
            cbCategoryName.Items.Add(rs.Fields("CategoryName").Value)
            rs.MoveNext()
        End While
        rs.Close()
        cbCategoryName.SelectedIndex = 0
    End Sub

    Private Sub PopulateSimpleNavigationForm()
        ' Thiết lập form với dữ liệu từ recordset
        txtCompanyName.Text = CStr(rs.Fields("CompanyName").Value)
        txtContactName.Text = CStr(rs.Fields("ContactName").Value)
        txtPhone.Text = CStr(rs.Fields("Phone").Value)
    End Sub
End Class
```

56. Định dạng dữ liệu trước khi hiển thị và lưu vào database

Chương trình này chỉ cách sử dụng các thủ tục sự kiện Format và Parse khi kết dữ liệu vào các control Windows Form. Sự kiện Format phát sinh khi dữ liệu được truyền từ dữ liệu nguồn vào control, sự kiện Parse phát sinh khi dữ liệu được đưa trở lại vào dữ liệu nguồn. Bằng cách xử lý các sự kiện này ta có thể hiển thị dữ liệu theo một định dạng nhưng lưu nó với một định dạng khác.



Các chức năng chính:

- Dữ liệu được đưa vào DataSet bằng cách tạo mới một bảng Employees (sử dụng SELECT INTO) và sau đó thêm một cột để thêm một số dữ liệu.
- Có thể duyệt qua các record bằng cách nhấn chuột trên các nút duyệt, hoặc các phím.
- Dữ liệu được định dạng và được phân tách trong các thủ tục sự kiện tương ứng.

- Một DataGrid hiển thị dữ liệu của bảng thứ hai trong DataSet bằng cách lọc DataView dựa trên EmployeeID của record hiện hành.
- Thay đổi dữ liệu trên các field có thể được lưu vào database bằng cách nhấn vào nút Save.

Project: **Custom DataBinding Format Handlers**

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=northwind;" & _
        "Integrated Security=SSPI"

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL " & _
        "or MSDE with the Northwind database installed."

    Protected blnDidPreviouslyConnect As Boolean = False
    Protected connectionString As String = SQL_CONNECTION_STRING
    Private da As SqlDataAdapter
    Private dsEmployees As DataSet
    Private dtEmployeeInfo As DataTable
    Private dtEmployeeOrders As DataTable
    Private dtEmployeeSales As DataTable
    Private dvEmployeeOrders As DataView
    Private dvEmployeeSales As DataView

    ' Code phát sinh form được bỏ qua
```

```
' Tạo DataSet
#Region " Creating the Demo DataSet "
    ' Thủ tục này tạo một bảng mới trong database Northwind dựa trên bảng Employees.
    ' Sau đó đưa vào một số dữ liệu giả để thấy được cách định dạng các field DateTime
    ' và SmallInt (lưu giá trị Boolean trong ứng dụng này).
  Sub CreateDataSet()

    Dim frmStatusMessage As New frmStatus()
    If Not blnDidPreviouslyConnect Then
      frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim blnIsConnecting As Boolean = True

    While blnIsConnecting
      Try
        Dim cnnNW As New SqlConnection(connectionString)

          ' Tạo một câu lệnh kiểm tra sự tồn tại của bảng mới. Nếu đã có thì xóa
          ' nó và tạo lại. Sau đó thêm hai cột để chứa dữ liệu giả.
        Dim strSQL As String = _
          "IF EXISTS (" & _
          "SELECT * " & _
          "FROM northwind.dbo.sysobjects " & _
          "WHERE NAME = 'NewEmployees' " & _
          "AND TYPE = 'u')" & vbCrLf & _
          "BEGIN" & vbCrLf & _
          "DROP TABLE northwind.dbo.NewEmployees" & vbCrLf & _
          "END" & vbCrLf & _
          "SELECT EmployeeID, LastName, FirstName, HireDate, Region " & _
          "INTO NewEmployees " & _
          "FROM Employees " & vbCrLf & _
          "ALTER Table NewEmployees " & _
          "ADD " & _
          "IsMarried SmallInt NULL, " & _
          "CONSTRAINT [PK_Employee] PRIMARY KEY CLUSTERED" & _
          "(EmployeeID)"
```

```
  ' Đối tượng SqlCommand được sử dụng để thực hiện lệnh SQL.
Dim cmd As New SqlCommand(strSQL, cnnNW)
  ' Mở kết nối và thực hiện các lệnh SQL
cnnNW.Open()
cmd.ExecuteNonQuery()

  ' Chuẩn bị dữ liệu giả
Dim arIIsMarried As New ArrayList()
With arIIsMarried
   .Add(1)
   .Add(0)
   .Add(0)
   .Add(1)
   .Add(1)
   .Add(1)
   .Add(0)
   .Add(1)
   .Add(0)
End With

  ' Thêm các SQL Parameter để sử dụng khi đưa dữ liệu vào các cột mới
Dim paramIsMarried As New SqlParameter("@IsMarried", _
   SqlDbType.SmallInt)
cmd.Parameters.Add(paramIsMarried)

Dim paramEmployeeID As New SqlParameter("@EmployeeID", _
   SqlDbType.Int)
cmd.Parameters.Add(paramEmployeeID)

  ' Câu lệnh SQL để đưa dữ liệu vào các cột mới
cmd.CommandText = _
   "UPDATE NewEmployees " & _
   "SET IsMarried = @IsMarried " & _
   "WHERE EmployeeID = @EmployeeID"

  ' Đưa dữ liệu vào các cột với dữ liệu là ArrayList
Dim i As Int32
Dim obj As Object
```

```
For Each obj In arIsMarried
   paramIsMarried.Value = arIsMarried(i)
   paramEmployeeID.Value = i + 1
   cmd.ExecuteNonQuery()
   i += 1
Next
  ' Thay câu lệnh SQL thành lệnh lấy dữ liệu Employee
cmd.CommandText = _
   "SELECT EmployeeID, LastName, FirstName, HireDate, Region, " & _
   "      IsMarried " & _
   "FROM NewEmployees"
' SqlDataAdapter sử dụng SqlCommand để đưa dữ liệu vào DataSet
da = New SqlDataAdapter(cmd)
  ' Đối tượng SqlCommandBuilder tự động phát sinh các lệnh SQL cần
  ' thiết để cập nhật lại database (trong thủ tục btnSave_Click).
Dim scb As New SqlCommandBuilder(da)
scb.GetUpdateCommand()

  ' Tạo và đưa dữ liệu vào DataSet mới
dsEmployees = New DataSet()
da.Fill(dsEmployees, "EmployeeInfo")

  ' Lệnh SQL  để lấy các Order của Empoyee
cmd.CommandText = _
   "SELECT od.OrderID, SUM(CONVERT(money, (od.UnitPrice * " & _
   "  od.Quantity) * (1 - od.Discount) / 100) * 100) " & _
   "  AS Subtotal, o.EmployeeID" & vbCrLf & _
   "FROM [Order Details] od INNER JOIN Orders o " & _
   "   ON od.OrderID = o.OrderID" & vbCrLf & _
   "GROUP BY od.OrderID, o.EmployeeID"

  ' Đưa dữ liệu vào bảng thứ hai của DataSet
da.Fill(dsEmployees, "EmployeeOrders")

  ' Lệnh SQL để lấy Sales To Date của Emplyee
cmd.CommandText = _
   "SELECT e.employeeid, sum(UnitPrice * Quantity) as " & _
   "  'SalesToDate' " & _
   "FROM [order details] od " & _
```

```
            "INNER JOIN orders o " & _
            "ON o.orderid = od.orderid " & _
            "INNER JOIN employees e " & _
            "ON e.employeeid = o.employeeid" & vbCrLf & _
            "GROUP BY e.employeeid"

            ' Đưa dữ liệu vào bảng thứ ba của DataSet
          da.Fill(dsEmployees, "EmployeeSales")
            ' Gán các biến cho các DataTable để sử dụng sau này
          dtEmployeeInfo = dsEmployees.Tables(0)
          dtEmployeeOrders = dsEmployees.Tables(1)
          dtEmployeeSales = dsEmployees.Tables(2)
            ' Gán các biến DataView
          dvEmployeeOrders = dtEmployeeOrders.DefaultView
          dvEmployeeSales = dtEmployeeSales.DefaultView
            ' Thoát vòng lặp
          blnIsConnecting = False
          blnDidPreviouslyConnect = True
          frmStatusMessage.Close()
        Catch sqlExc As SqlException
          MessageBox.Show(sqlExc.ToString, "SQL Exception Error!", _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
          Exit While
        Catch exp As Exception
          MessageBox.Show(exp.Message)
          If connectionString = SQL_CONNECTION_STRING Then
            connectionString = MSDE_CONNECTION_STRING
            frmStatusMessage.Show("Connecting to MSDE")
          Else
            frmStatusMessage.Close()
            MessageBox.Show(CONNECTION_ERROR_MSG, _
                "Connection Failed!", MessageBoxButtons.OK, _
                MessageBoxIcon.Error)
            End
          End If
        End Try
      End While
    End Sub
#End Region
```

```
Private Sub btnFirst_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFirst.Click
    ' Di chuyển đến record đầu tiên
    FirstRecord()
End Sub

Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnLast.Click
    ' Di chuyển đến record cuối cùng
    LastRecord()
End Sub

Private Sub btnNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNext.Click
    ' Di chuyển đến record kế tiếp
    NextRecord()
End Sub
Private Sub btnPrevious_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPrevious.Click
    ' Quay lui về một record
    PreviousRecord()
End Sub

' Nếu muốn cập nhật vào database ta cần gọi phương thức Update của đối
' tượng SqlDataAdapter.
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    Try
        ' Kết thúc thao tác thay đổi hiện hành. Nếu không, khi người sử dụng thay
        ' đổi nội dung và nhấn vào nút Save mà không chuyển qua reocord khác thì
        ' các thay đổi sẽ không có tác dụng.
        Me.BindingContext(dtEmployeeInfo).EndCurrentEdit()

        ' Cập nhật database với các thay đổi đã tiến hành trên DataSet
        da.Update(dtEmployeeInfo)

        MessageBox.Show("Database successfully updated.", _
          "Custom Formatting Handler Demo", MessageBoxButtons.OK, _
          MessageBoxIcon.Information)
```

```
    Catch exp As Exception
      MessageBox.Show("There was an error when attempting to update " & _
        "the database: " & exp.Message, _
        Me.Text, MessageBoxButtons.OK, _
        MessageBoxIcon.Error)
    End Try
  End Sub

  ' Thủ tục này được gọi khi sự kiện PositionChanged của BindingContext phát sinh.
  ' Hai sự kiện này liên quan với nhau là do lệnh AddHandler trong frmMain_Load.
  Protected Sub dtEmployeeInfo_PositionChanged(ByVal sender As Object, ByVal e
  As System.EventArgs)
    BindAndFormatGrid()
    ShowTotalSales()
    ShowCurrentRecordNumber()
  End Sub
  ' Xử lý nhấn phím trên form
  Private Sub frmMain_KeyDown(ByVal sender As Object, ByVal e As
  System.Windows.Forms.KeyEventArgs) Handles MyBase.KeyDown
    If e.KeyCode = Keys.Right Then NextRecord()
    If e.KeyCode = Keys.Left Then PreviousRecord()
    If e.KeyCode = Keys.Home Then FirstRecord()
    If e.KeyCode = Keys.End Then LastRecord()
  End Sub

  Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
  Handles MyBase.Load

    ' Tạo DataSet với một số dữ liệu
    CreateDataSet()

    ' Quy định kết dữ liệu cho các control
    txtLastName.DataBindings.Add("Text", dtEmployeeInfo, "LastName")

    ' Sử dụng thuộc tính Tag ta có thể kết một giá trị ẩn, giá trị này có thể cần thiết
    ' khi duyệt qua các record. Trong chương trình này, số EmployeeID tương ứng với
    ' số dòng của DataTable, do đó có thể sử dụng thuộc tính Position như trong
    ' thủ tục ShowCurrentRecordNumber.
    txtLastName.DataBindings.Add("Tag", dtEmployeeInfo, "EmployeeID")
    txtFirstName.DataBindings.Add("Text", dtEmployeeInfo, "FirstName")
```

```
    Dim dbnHireDate As New Binding("Text", dtEmployeeInfo, "HireDate")
    AddHandler dbnHireDate.Format, AddressOf DateToString
    AddHandler dbnHireDate.Parse, AddressOf StringToDate
    txtHireDate.DataBindings.Add(dbnHireDate)

    Dim dbnRegion As New Binding("Text", dtEmployeeInfo, "Region")
    AddHandler dbnRegion.Format, AddressOf NullToString
    AddHandler dbnRegion.Parse, AddressOf StringToNull
    txtRegion.DataBindings.Add(dbnRegion)

    Dim dbnIsMarried As New Binding("Checked", dtEmployeeInfo, "IsMarried")

    AddHandler dbnIsMarried.Format, AddressOf SmallIntToBoolean
    AddHandler dbnIsMarried.Parse, AddressOf BooleanToSmallInt
    chkIsMarried.DataBindings.Add(dbnIsMarried)

    Dim dbnSalesToDate As New Binding("Text", dtEmployeeSales, "SalesToDate")
    AddHandler dbnSalesToDate.Format, AddressOf MoneyToString
    AddHandler dbnSalesToDate.Parse, AddressOf StringToMoney
    txtSalesToDate.DataBindings.Add(dbnSalesToDate)

    AddHandler Me.BindingContext(dtEmployeeInfo).PositionChanged, _
      AddressOf dtEmployeeInfo_PositionChanged

    BindAndFormatGrid()
    ShowCurrentRecordNumber()
End Sub

' Kết DataGrid vào bảng EmployeeOrders và áp dụng định dạng riêng.
Sub BindAndFormatGrid()
    ' Lọc OrderID dựa trên giá trị của thuộc tính Tag đã gán
    dvEmployeeOrders.RowFilter = "EmployeeID = " & txtLastName.Tag.ToString

    With grdOrders
      .CaptionText = "Orders"
      .DataSource = dvEmployeeOrders
    End With
```

```
    ' Trước tiên cần xóa tập hợp TableStyles
    grdOrders.TableStyles.Clear()
    Dim grdTableStyle1 As New DataGridTableStyle()
    With grdTableStyle1
        ' Luôn luôn gán MappingName để định dạng có  hiệu lực
        .MappingName = "EmployeeOrders"
    End With

    Dim grdColStyle1 As New DataGridTextBoxColumn()
    With grdColStyle1
        .MappingName = "OrderID"
        .HeaderText = "Order ID"
        .Width = 75
    End With

    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
        .MappingName = "SubTotal"
        .HeaderText = "Sub Total"
        .Format = "c"
        .Width = 75
    End With
    grdTableStyle1.GridColumnStyles.AddRange _
        (New DataGridColumnStyle() {grdColStyle1, grdColStyle2})
    grdOrders.TableStyles.Add(grdTableStyle1)
End Sub

' Xử lý sự kiện Parse cho control CheckBox
Protected Sub BooleanToSmallInt(ByVal sender As Object, ByVal e As
ConvertEventArgs)
    Select Case e.Value
        Case True
            e.Value = 1
        Case Else
            e.Value = 0
    End Select
End Sub
```

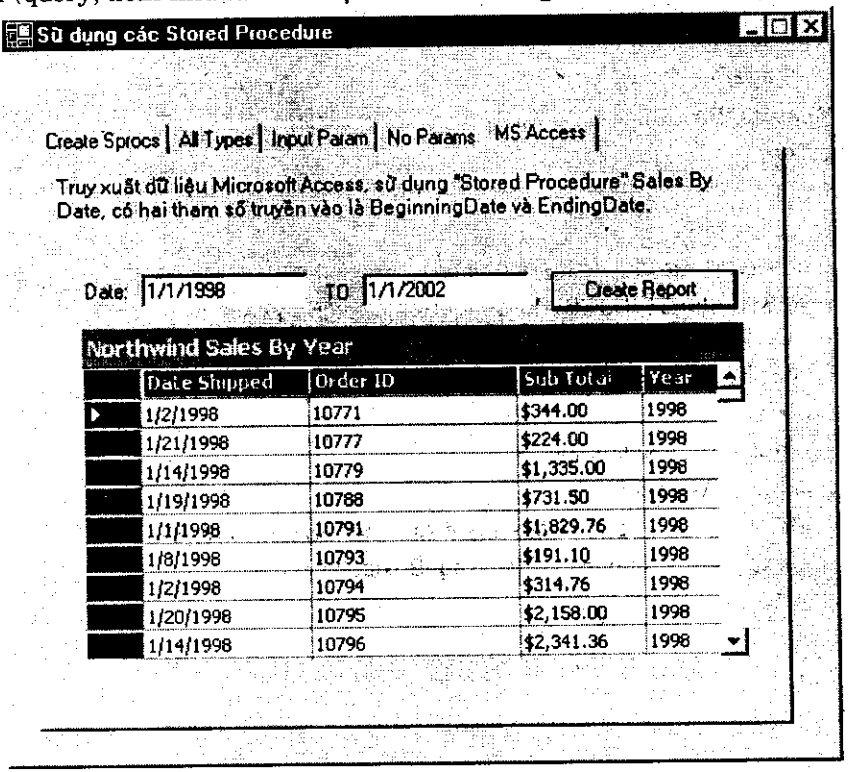
```
' Xử lý sự kiện Format cho TextBox
Protected Sub DateToString(ByVal sender As Object, ByVal e As
ConvertEventArgs)
    ' Có thể dùng một trong hai cách sau để chuyển thành kiểu ngày thích hợp
    ' e.Value = CType(e.Value, DateTime).ToString("d")
    e.Value = CType(e.Value, DateTime).ToShortDateString
End Sub

' Di chuyển BindingContext Position đến record đầu tiên
Public Sub FirstRecord()
    Me.BindingContext(dtEmployeeInfo).Position = 0
End Sub

' Di chuyển BindingContext Position đến record cuối cùng
Public Sub LastRecord()
    Me.BindingContext(dtEmployeeInfo).Position = dtEmployeeInfo.Rows.Count - 1
End Sub

' Xử lý sự kiện Format cho TextBox SalesToDate
Protected Sub MoneyToString(ByVal sender As Object, ByVal e As
ConvertEventArgs)
    e.Value = CType(e.Value, Decimal).ToString("c")
End Sub

' Di chuyển BindingContext Position đến record kế tiếp
Public Sub NextRecord()
    Me.BindingContext(dtEmployeeInfo).Position += 1
End Sub

' Xử lý sự kiện Format cho TextBox Region
Protected Sub NullToString(ByVal sender As Object, ByVal e As
ConvertEventArgs)

    If IsDBNull(e.Value) Or e.Value.ToString.Trim.Length = 0 Then
        e.Value = "[N/A]"
    End If
End Sub
```

```
    ' Di chuyển BindingContext Position lùi về một record
    Public Sub PreviousRecord()
        Me.BindingContext(dtEmployeeInfo).Position -= 1
    End Sub

    ' Hiển thị số record hiện hành
    Protected Sub ShowCurrentRecordNumber()
        lblRecordNumber.Text = "Record " & _
            Me.BindingContext(dtEmployeeInfo).Position + 1 & " of " & _
                dtEmployeeInfo.Rows.Count
    End Sub

    ' Cập nhật giá trị Sales To Date của mỗi employee khi người sử dụng
    ' chuyển qua các record khác.
    Protected Sub ShowTotalSales()
        ' Lọc Sales Total dựa trên giá trị của thuộc tính Tag
        dvEmployeeSales.RowFilter = "EmployeeID = " & txtLastName.Tag.ToString
    End Sub

    ' Xử lý sự kiện Format cho CheckBox
    Protected Sub SmallIntToBoolean(ByVal sender As Object, ByVal e As
    ConvertEventArgs)
        Select Case e.Value
            Case 1
                e.Value = True
            Case Else
                e.Value = False
        End Select
    End Sub

    ' Xử lý sự kiện Parse cho TextBox Hire Date
    Protected Sub StringToDate(ByVal sender As Object, ByVal e As ConvertEventArgs)
        Try
            e.Value = CDate(e.Value)
        Catch exp As Exception
            MessageBox.Show("Data entry error: " & exp.Message, Me.Text, _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub
```

```
    ' Xử lý sự kiện Parse cho TextBox Sales To Date
    Protected Sub StringToMoney(ByVal sender As Object, ByVal e As
    ConvertEventArgs)
        Try
            ' Double bằng với kiểu dữ liệu Money của SQL
            e.Value = CType(e.Value, Double)
        Catch exp As Exception
            MessageBox.Show("Data entry error: " & exp.Message, Me.Text, _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

    ' Xử lý sự kiện Parse cho TextBox Region
    Protected Sub StringToNull(ByVal sender As Object, ByVal e As
    ConvertEventArgs)
        If e.Value.ToString = "[N/A]" Or e.Value.ToString.Trim.Length = 0 Then
            Try
                e.Value = DBNull.Value
            Catch exp As Exception
                MessageBox.Show("Data entry error: " & exp.Message, Me.Text, _
                MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End If
    End Sub
End Class
```

57. Sử dụng Stored Procedure

Chương trình này chỉ cách lập trình tạo và sử dụng các thủ tục lưu trữ nội (Stored Procudure) với ADO .NET và SQL Server. Cách sử dụng các câu truy vấn (query, xem như là stored procedure) trong database Microsoft Access.



Các chức năng chính:

- Lập trình tạo các stored procedure trong database SQL Server "Northwind" bằng cách thực hiện các câu lệnh SQL chuẩn sử dụng namespace System.Data.SqlClient.
- Thực thi một stored procedure không yêu cầu tham số, sử dụng đối tượng SqlDataReader.
- Thực thi một stored procedure có yêu cầu tham số và đưa kết quả vào một DataGrid.
- Thực thi một stored procedure yêu cầu cả tham số nhập vào và tham số xuất ra, và sử dụng giá trị trả về.
- Thực thi một Query trong database Northwind phiên bản MS Access sử dụng các class trong namespace System.Data.OleDb.

frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.Data.OleDb
Imports System.Text

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    ' Đổi đường dẫn đến file Northwind nếu cần thiết.
    Protected Const ACCESS_CONNECTION_STRING As String = _
    "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=..\Northwind.mdb"

    Protected Const CONNECTION_ERROR_MSG As String = _
        "To run this sample, you must have SQL " & _
        "or MSDE with the Northwind database installed."

    Protected DidPreviouslyConnect As Boolean = False
    Protected HasCreatedSprocs As Boolean = False
    Protected strConn As String

' Code phát sinh form được bỏ qua
```

```
' Xử lý nút Create Report trên tab MS Access. Thủ tục này sử dụng các class trong
' namespace System.Data.OleDb để kết nối vào database Northwind (Access)
' và thực hiện một câu query nhận vào hai tham số
Private Sub btnCreateReport_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCreateReport.Click
    ' Kiểm tra ngày bắt đầu và ngày kết thúc do người sử dụng nhập vào
    If Not DatesAreValid() Then
        Exit Sub
    End If

    strConn = ACCESS_CONNECTION_STRING
    ' Khởi tạo các biến đối tượng trong namespace OleDb
    Dim ocnnNorthwind As New OleDbConnection(strConn)
    Dim ocmd As New OleDbCommand("[Sales By Year]", ocnnNorthwind)
    Dim oda As New OleDbDataAdapter(ocmd)
    ' Tạo một DataSet để chứa dữ liệu được đưa vào từ OleDbDataAdapter
    Dim dsSales As New DataSet()

    ' Mặc nhiên CommandType = Text (ví dụ: các lệnh SQL chuẩn)
    ocmd.CommandType = CommandType.StoredProcedure

    ' Thêm các tham số cần thiết cho query và gán giá trị cho chúng.
    ' Các tham số này được thêm theo cách tắt. Cách dài hơn có thể xem trong
    ' thủ tục btnGetProducts_Click
    With ocmd.Parameters
        .Add(New OleDbParameter("@BeginningDate", OleDbType.Date)).Value = _
            CDate(txtBeginningDate.Text.Trim)
        .Add(New OleDbParameter("@EndingDate", OleDbType.Date)).Value = _
            CDate(txtEndingDate.Text.Trim)
    End With

    Try
        ' Đưa dữ liệu vào DataSet
        oda.Fill(dsSales)
    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
            MessageBoxIcon.Error)
        Exit Sub
    End Try
```

```
' Chỉ kết dữ liệu và hiển thị DataGrid nếu DataSet có chứa kết quả.
    If dsSales.Tables(0).Rows.Count > 0 Then

        ' Kết DataGrid vào bảng biết trước trong DataSet.
        ' Thao tác này sẽ hiển thị kết quả.
      grdSales.DataSource = dsSales.Tables(0)

    ' Tiếp tục quy định trực tiếp các thuộc tính của DataGrid,
    ' nhưng chỉ các thuộc tính không có trong các thuộc tính của DataGridTableStyle.
      With grdSales
        .BackColor = Color.GhostWhite
        .BackgroundColor = Color.Lavender
        .BorderStyle = BorderStyle.None
        .CaptionBackColor = Color.RoyalBlue
        .CaptionFont = New Font("Tahoma", 10.0!, FontStyle.Bold)
        .CaptionForeColor = Color.Bisque
        .CaptionText = "Northwind Sales By Year"
        .Font = New Font("Tahoma", 8.0!)
        .ParentRowsBackColor = Color.Lavender
        .ParentRowsForeColor = Color.MidnightBlue
          ' Xóa tất cả các đối tượng TableStyle đã thêm vào lúc trước
          ' nếu không sẽ gây ra exception.
        .TableStyles.Clear()
      End With

        ' Các định dạng khác
      Dim grdTableStyle1 As New DataGridTableStyle()
      With grdTableStyle1
        .AlternatingBackColor = Color.GhostWhite
        .BackColor = Color.GhostWhite
        .ForeColor = Color.MidnightBlue
        .GridLineColor = Color.RoyalBlue
        .HeaderBackColor = Color.MidnightBlue
        .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
        .HeaderForeColor = Color.Lavender
        .SelectionBackColor = Color.Teal
        .SelectionForeColor = Color.PaleGreen
```

```
        ' Đừng quên gán thuộc tính MappingName
        .MappingName = dsSales.Tables(0).TableName
        .PreferredColumnWidth = 125
        .PreferredRowHeight = 15
    End With

    ' Định dạng các cột muốn xuất hiện trên DataGrid.
    Dim grdColStyle1 As New DataGridTextBoxColumn()
    With grdColStyle1
        .HeaderText = "Date Shipped"
        .MappingName = "ShippedDate"
        .Width = 100
    End With

    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
        .HeaderText = "Order ID"
        .MappingName = "OrderID"
    End With

    Dim grdColStyle3 As New DataGridTextBoxColumn()
    With grdColStyle3
        .HeaderText = "Sub Total"
        .MappingName = "Subtotal"
        .Format = "c"
        .Width = 75
        .ReadOnly = True
    End With

    Dim grdColStyle4 As New DataGridTextBoxColumn()
    With grdColStyle4
        .HeaderText = "Year"
        .MappingName = "Year"
        .Width = 50
    End With
    ' Cho các đối tượng style vào tập hợp ColumnStyles của TableStyle
    grdTableStyle1.GridColumnStyles.AddRange _
        (New DataGridColumnStyle() _
        {grdColStyle1, grdColStyle2, grdColStyle3, grdColStyle4})
    grdSales.TableStyles.Add(grdTableStyle1)
```

```
    Else
        lblNoSales.Visible = True
        grdSales.Visible = False
    End If
End Sub

' Xử lý nút Create Sprocs trên tab Create Sprocs. Thủ tục này sử dụng các class
' của namespace System.Data.SqlClient để thực hiện các lệnh SQL xóa một
' Stored Procedure (nếu đã tồn tại) và tạo lại nó.
Private Sub btnCreateSprocs_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCreateSprocs.Click

    strConn = SQL_CONNECTION_STRING

    Dim frmStatusMessage As New frmStatus()
    If Not DidPreviouslyConnect Then
        frmStatusMessage.Show("Connecting to SQL Server")
    End If

    Dim IsConnecting As Boolean = True
    While IsConnecting
        Try

            Dim northwindConnection As New SqlConnection(strConn)
            ' DROP "GetCategories":
            ' Lệnh SQL này xóa SP "GetCategories" nếu nó đã tồn tại
            Dim strSQL As String = _
                "IF EXISTS (" & _
                "SELECT * " & _
                "FROM northwind.dbo.sysobjects " & _
                "WHERE Name = 'GetCategories' " & _
                "AND TYPE = 'p')" & vbCrLf & _
                "DROP PROCEDURE GetCategories"

            ' Đối tượng SqlCommand được sử dụng để thực hiện các lệnh SQL
            Dim scmd As New SqlCommand(strSQL, northwindConnection)
```

```
Try
    ' Mở một kết nối và để nó tiếp tục còn mở cho đến khi lệnh
    ' SQL cuối cùng được thực hiện
    northwindConnection.Open()

    ' Kết nối thành công do đó thoát vòng lặp, đóng form thông báo
    IsConnecting = False
    DidPreviouslyConnect = True
    frmStatusMessage.Close()

    ' Thực hiện câu lệnh SQL sử dụng ExecuteNonQuery, sẽ hiệu quả
    ' hơn trong trường hợp không có dữ liệu trả về
    scmd.ExecuteNonQuery()

  Catch expSql As SqlException
    MessageBox.Show(expSql.ToString, Me.Text, _
      MessageBoxButtons.OK, MessageBoxIcon.Error)
    Exit Sub
  End Try

  ' CREATE "GetCategories"
  ' Lệnh SQL này tạo SP "GetCategories".
  scmd.CommandText = _
    "CREATE PROCEDURE GetCategories " & vbCrLf & _
    "AS " & vbCrLf & _
    "SELECT CategoryID, CategoryName " & _
    "FROM Northwind.dbo.Categories"

  Try
    scmd.ExecuteNonQuery()
  Catch expSql As SqlException
    MessageBox.Show(expSql.ToString, Me.Text, _
      MessageBoxButtons.OK, MessageBoxIcon.Error)
    Exit Sub
  End Try
```

```
' DROP "GetProducts"
  ' Lệnh SQL này xóa SP "GetProducts" nếu nó đã tồn tại
scmd.CommandText = _
  "IF EXISTS (" & _
  "SELECT * " & _
  "FROM northwind.dbo.sysobjects " & _
  "WHERE Name = 'GetProducts' " & _
  "AND TYPE = 'p')" & vbCrLf & _
  "DROP PROCEDURE GetProducts"

Try
  scmd.ExecuteNonQuery()

Catch expSql As SqlException
  MessageBox.Show(expSql.ToString, Me.Text, _
    MessageBoxButtons.OK, MessageBoxIcon.Error)
  Exit Sub
End Try

  ' CREATE "GetProducts"
  ' Lệnh SQL tạo SP " GetProducts "
scmd.CommandText = _
  "CREATE PROCEDURE GetProducts " & vbCrLf & _
  "@CategoryID Int " & vbCrLf & _
  "AS " & vbCrLf & _
  "SELECT ProductID, ProductName, UnitPrice, UnitsInStock " & _
  "FROM Northwind.dbo.Products " & _
  "WHERE CategoryID = @CategoryID"

Try
  scmd.ExecuteNonQuery()

Catch expSql As SqlException
  MessageBox.Show(expSql.ToString, Me.Text, _
    MessageBoxButtons.OK, MessageBoxIcon.Error)
  Exit Sub

End Try
```

```
    ' DROP "GetProductCountAndAveragePrice"
    ' Xóa SP "GetProductCountAndAveragePrice" nếu nó đã tồn tại
  scmd.CommandText = _
    "IF EXISTS (" & _
    "SELECT * " & _
    "FROM northwind.dbo.sysobjects " & _
    "WHERE Name = 'GetProductCountAndAveragePrice' " & _
    "AND TYPE = 'p')" & vbCrLf & _
    "DROP PROCEDURE GetProductCountAndAveragePrice"

  Try
    scmd.ExecuteNonQuery()

  Catch expSql As SqlException
    MessageBox.Show(expSql.ToString, Me.Text, _
      MessageBoxButtons.OK, MessageBoxIcon.Error)
    Exit Sub
  End Try

    ' CREATE "GetProductCountAndAveragePrice":
    ' Tạo SP "GetProductCountAndAveragePrice"
  scmd.CommandText = _
"CREATE PROCEDURE GetProductCountAndAveragePrice " & vbCrLf & _
    "@CategoryID Int, " & vbCrLf & _
    "@AveragePrice Int OUT" & vbCrLf & _
    "AS " & vbCrLf & _
    "DECLARE @SumProdPrices Money " & vbCrLf & _
    "SELECT @AveragePrice = SUM(UnitPrice)/COUNT(ProductID) " & _
    "FROM Northwind.dbo.Products " & _
    "WHERE CategoryID = @CategoryID" & vbCrLf & _
    "RETURN " & _
    "(SELECT COUNT(ProductID) " & _
    "FROM Northwind.dbo.Products " & _
    "WHERE CategoryID = @CategoryID)"
  Try
    scmd.ExecuteNonQuery()
  Catch expSql As SqlException
    MessageBox.Show(expSql.ToString, Me.Text, _
      MessageBoxButtons.OK, MessageBoxIcon.Error)
    Exit Sub
```

```
            Finally
                ' Đối với SP cuối cùng, đóng kết nối cho dù có lỗi nào xảy ra
              northwindConnection.Close()
            End Try

          Catch exp As Exception
            If strConn = SQL_CONNECTION_STRING Then
              strConn = MSDE_CONNECTION_STRING
              frmStatusMessage.Show("Connecting to MSDE")
            Else
              frmStatusMessage.Close()
              MessageBox.Show("To run this sample, you must have SQL " & _
                "or MSDE with the Northwind database installed. ", _
                Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Error)
              End
            End If
          End Try
        End While

        frmStatusMessage.Close()

        MessageBox.Show("The stored procedures were successfully added to the " & _
          "Northwind database.", Me.Text, MessageBoxButtons.OK, _
          MessageBoxIcon.Information)

        HasCreatedSprocs = True
      End Sub

      ' Thủ tục này xử lý nút Get Product Count và Avg Price trên tab All Types.
      ' SP trong thủ tục này yêu cầu một tham số truyền vào và gửi lại một tham số xuất
      ' và một giá trị trả về.
      Private Sub btnGetProductCountAndAvgPrice_Click(ByVal sender As
      System.Object, ByVal e As System.EventArgs) Handles
      btnGetProductCountAndAvgPrice.Click
        strConn = SQL_CONNECTION_STRING

        Dim scnnNorthwind As New SqlConnection(strConn)
        Dim scmd As New SqlCommand("GetProductCountAndAveragePrice", _
                                   scnnNorthwind)
        Dim sda As New SqlDataAdapter(scmd)
```

```
    Dim dsProducts As New DataSet()

    scmd.CommandType = CommandType.StoredProcedure

    ' Tham số truyền vào lấy giá trị từ ComboBox. Gán giá trị cho Direction là output.
    ' Cuối cùng, thêm một tham số để lấy giá trị trả về, giá trị trả về từ lệnh RETURN
    ' sử dụng trong SP, nếu không có lệnh này, giá trị trả về mặc nhiên là giá trị
  ' Thành công/Lỗi do SQL Server trả về.
  With scmd.Parameters
    .Add(New SqlParameter("@CategoryID", SqlDbType.Int)).Value = _
      cboCategoriesAllTypes.SelectedValue
      ' Mặc nhiên, giá trị của Direction là "Input", do đó chỉ cần gán giá trị cho
      ' nó đối với hai tham số sau.
    .Add(New SqlParameter("@AveragePrice", _
      SqlDbType.Money)).Direction = ParameterDirection.Output
      ' Thay vì thêm một SqlParameter để lấy giá trị trả về, ta có thể chỉ cần
      ' khai báo một biến Integer và khởi tạo bằng giá trị trả về bởi phương
      ' thức Fill của SqlDataAdapter.
    .Add(New SqlParameter("ReturnValue", _
      SqlDbType.Int)).Direction = ParameterDirection.ReturnValue
  End With

  Try
      ' Để biết thêm thông tin về dòng code này, xem khối Try...Catch trong
      ' thủ tục frmMain_Load.
    sda.Fill(dsProducts, "Products")
  Catch expSQL As SqlException
    MessageBox.Show(expSQL.ToString, Me.Text, _
      MessageBoxButtons.OK, MessageBoxIcon.Error)
    Exit Sub
  End Try

    ' Hiển thị kết quả
  lblProductCountAndAvgPrice.Text = "There are " & _
    scmd.Parameters("ReturnValue").Value.ToString & " products " & _
    "in the " & cboCategoriesAllTypes.Text & " category, at an average " & _
    "price of " & CType(scmd.Parameters("@AveragePrice").Value, _
      Double).ToString("c")
End Sub
```

```
' Xử lý cho nút Get Products trên tab Input Param. SP "GetProducts" yêu cầu một
' tham số truyền vào. SP được thực hiện và một DataGrid hiển thị kết quả.
Private Sub btnGetProducts_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetProducts.Click

    strConn = SQL_CONNECTION_STRING
    ' Xem thêm thông tin các dòng lệnh sau trong frmMain_Load
    Dim scnnNorthwind As New SqlConnection(strConn)
    Dim scmd As New SqlCommand("GetProducts", scnnNorthwind)
    Dim sda As New SqlDataAdapter(scmd)
    Dim dsProducts As New DataSet()

    ' Thêm tham số yêu cầu bởi SP. Giá trị của tham số này lấy từ ComboBox.
    ' Bên dưới là cách dài để thêm một Sql Parameter và gán các thuộc
    ' tính của nó. Xem trong btnGetProductCountAndAvgPrice_Click để thấy
    ' cách ngắn hơn.
    Dim sparCatID As New SqlParameter()
    With sparCatID
        .ParameterName = "@CategoryID"
        .SqlDbType = SqlDbType.Int
        .Value = cboCategoriesInputParam.SelectedValue
    End With

    With scmd
        .Parameters.Add(sparCatID)
        .CommandType = CommandType.StoredProcedure
    End With

    Try
        ' Đưa dữ liệu vào DataSet và có thể đặt tên cho DataTable nếu muốn.
        sda.Fill(dsProducts, "Products")

    Catch expSQL As SqlException
        MessageBox.Show(expSQL.ToString, Me.Text, _
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        Exit Sub
    End Try
```

```
    ' Kết DataGrid vào bảng trong DataSet và hiển thị.
    grdProducts.DataSource = dsProducts.Tables(0)

    With grdProducts
        .BackColor = Color.GhostWhite
        .BackgroundColor = Color.Lavender
        .BorderStyle = BorderStyle.None
        .CaptionBackColor = Color.RoyalBlue
        .CaptionFont = New Font("Tahoma", 10.0!, FontStyle.Bold)
        .CaptionForeColor = Color.Bisque
        .CaptionText = "Northwind Products"
        .Font = New Font("Tahoma", 8.0!)
        .ParentRowsBackColor = Color.Lavender
        .ParentRowsForeColor = Color.MidnightBlue
        .TableStyles.Clear()
    End With

    Dim grdTableStyle1 As New DataGridTableStyle()
    With grdTableStyle1
        .AlternatingBackColor = Color.GhostWhite
        .BackColor = Color.GhostWhite
        .ForeColor = Color.MidnightBlue
        .GridLineColor = Color.RoyalBlue
        .HeaderBackColor = Color.MidnightBlue
        .HeaderFont = New Font("Tahoma", 8.0!, FontStyle.Bold)
        .HeaderForeColor = Color.Lavender
        .SelectionBackColor = Color.Teal
        .SelectionForeColor = Color.PaleGreen
        .MappingName = dsProducts.Tables(0).TableName
        .PreferredColumnWidth = 125
        .PreferredRowHeight = 15
    End With

    Dim grdColStyle1 As New DataGridTextBoxColumn()
    With grdColStyle1
        .HeaderText = "ID"
        .MappingName = "ProductID"
        .Width = 50
    End With
```

```
    Dim grdColStyle2 As New DataGridTextBoxColumn()
    With grdColStyle2
      .HeaderText = "Name"
      .MappingName = "ProductName"
    End With

    Dim grdColStyle3 As New DataGridTextBoxColumn()
    With grdColStyle3
      .HeaderText = "Price"
      .MappingName = "UnitPrice"
      .Format = "c"
      .Width = 75
      .ReadOnly = True
    End With

    Dim grdColStyle4 As New DataGridTextBoxColumn()
    With grdColStyle4
      .HeaderText = "# In Stock"
      .MappingName = "UnitsInStock"
      .Width = 75
      .Alignment = HorizontalAlignment.Center
    End With

    grdTableStyle1.GridColumnStyles.AddRange _
      (New DataGridColumnStyle() _
      {grdColStyle1, grdColStyle2, grdColStyle3, grdColStyle4})
    grdProducts.TableStyles.Add(grdTableStyle1)

  End Sub

  ' Thủ tục xử lý nút Top Ten Products trên tab No Params.
  Private Sub btnGet10MostExpProds_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnGet10MostExpProds.Click

    Dim scnnNorthwind As New SqlConnection(SQL_CONNECTION_STRING)
    Dim scmd As New SqlCommand("[Ten Most Expensive Products]",
scnnNorthwind)

    scmd.CommandType = CommandType.StoredProcedure
```

```
        ' Mở kết nối và truyền vào giá trị để đóng kết nối thay cho việc
        ' gọi scnnNorthwind.Close()
        scnnNorthwind.Open()
        Dim sdr As SqlDataReader =
    scmd.ExecuteReader(CommandBehavior.CloseConnection)

        ' Tạo một instance của StringBuilder để nối ghép các chuỗi để hiển thị
        ' trong TextBox. Class StringBuilder được thiết kế tối ưu cho việc xử lý chuỗi.
        Dim sb As New StringBuilder()
        sb.Append("Product Name")
        sb.Append(vbTab)
        sb.Append(vbTab)
        sb.Append(vbTab)
        sb.Append("Price")
        sb.Append(vbCrLf)
        sb.Append("==========================")
        sb.Append(vbTab)
        sb.Append("==========")
        sb.Append(vbCrLf)

        ' Duyệt qua nội dung của đối tượng SqlDataReader
        While sdr.Read
            sb.Append(sdr.GetString(0))
            sb.Append(vbTab)
            sb.Append(vbTab)
            If sdr.GetString(0).Length < 20 Then
                sb.Append(vbTab)
            End If

            ' Để định dạng chuỗi như là giá trị currency, trước tiên phải chuyển nó
            ' thành kiểu có interface IFormattable
            sb.Append(sdr.GetSqlMoney(1).ToDouble.ToString("c"))
            sb.Append(vbCrLf)
        End While

        ' Hiển thị kết quả
        txtTenMostExpProds.Text = sb.ToString
    End Sub
```

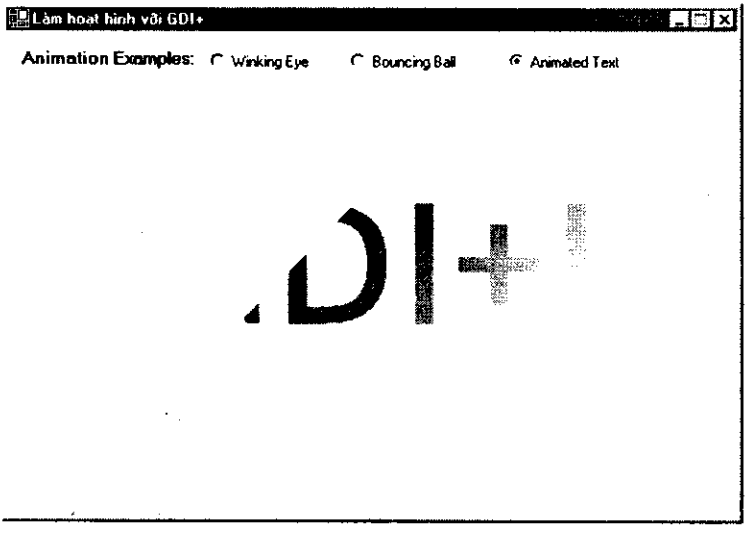
```
' Xử lý sự kiện SelectedIndexChanged của TabControl.
Private Sub tabApp_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles tabApp.SelectedIndexChanged
  If Not HasCreatedSprocs AndAlso tabApp.SelectedTab.TabIndex > 0 Then
    MessageBox.Show("You must first create the required stored procedures.", _
      Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Stop)
    tabApp.SelectedIndex = 0
  ElseIf HasCreatedSprocs AndAlso _
          cboCategoriesInputParam.Items.Count = 0 Then
    ' Đưa dữ liệu vào các ComboBox chỉ khi chúng chưa có dữ liệu và
    ' các SP đã được tạo.
    strConn = SQL_CONNECTION_STRING
    Dim scnnNorthwind As New SqlConnection(strConn)
    Dim dsCategories As DataSet
    Dim scmd As New SqlCommand("GetCategories", scnnNorthwind)
    Dim sda As New SqlDataAdapter(scmd)

    dsCategories = New DataSet()
    scmd.CommandType = CommandType.StoredProcedure
    Try
      sda.Fill(dsCategories)
    Catch expSql As SqlException
      MessageBox.Show(expSql.ToString, Me.Text, _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try

    ' Kết dữ liệu vào các ComboBox
    With cboCategoriesInputParam
      .DataSource = dsCategories.Tables(0)
      .DisplayMember = "CategoryName"
      .ValueMember = "CategoryID"
    End With

    With cboCategoriesAllTypes
      .DataSource = dsCategories.Tables(0)
      .DisplayMember = "CategoryName"
      .ValueMember = "CategoryID"
    End With
  End If
End Sub
```

```
    ' Kiểm tra ngày tháng do người sử dụng nhập vào trên tab "MS Access"
    Private Function DatesAreValid() As Boolean
        Dim objControl As Object
        For Each objControl In pgeMSAccess.Controls
            If objControl.GetType.ToString = "System.Windows.Forms.TextBox" Then
                Dim txt As TextBox = CType(objControl, TextBox)

                If Not IsDate(txt.Text.Trim) Then
                    MessageBox.Show("You must enter a correct date in the form " & _
                        "mm/dd/yyyy.", Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
                    txt.Focus()
                    txt.SelectAll()
                    Return False
                End If
            End If
        Next

        Return True
    End Function
End Class
```

Phần X - Microsoft .NET Framework

58. Làm hoạt hình với GDI+

Chương trình này biểu diễn kỹ thuật để làm hoạt hình (animation) với GDI+, bao gồm hoạt hình theo các khung hình (frame), vẽ và di chuyển các hình cơ bản trên màn hình, và hoạt hình tô màu văn bản.



Các chức năng chính:

- Làm hoạt hình kiểu frame, sử dụng các hình ảnh liên tục để hiển thị một con mắt đang chớp.
- Tạo một quả banh dội tường với kích thước và tốc độ của quả banh có phụ thuộc vào kích thước của cửa sổ.
- Làm hoạt hình cho văn bản bằng cách di chuyển các điểm bắt đầu và kết thúc vùng tô màu (màu biến thiên).

Project: **Lam hoat hinh voi GDI+**

Control Timer: tmrAnimation

frmMain.vb

```
Option Strict On
Imports System.Drawing.Drawing2D
Imports System.Drawing.Text
```

```
Public Class frmMain
    Inherits System.Windows.Forms.Form
    Friend WithEvents tmrAnimation As System.Windows.Forms.Timer
' Code phát sinh form được bỏ qua

    Const WINK_TIMER_INTERVAL As Integer = 150 ' Tính bằng milli giây
    Protected arrImages(4) As Image
    Protected intCurrentImage As Integer = 0
    Protected j As Integer = 1
    Const BALL_TIMER_INTERVAL As Integer = 25
    Private intBallSize As Integer = 16
    Private intMoveSize As Integer = 4
    Private bitmap As bitmap
    Private intBallPositionX, intBallPositionY As Integer
    Private intBallRadiusX, intBallRadiusY, intBallMoveX, intBallMoveY, _
      intBallBitmapWidth, intBallBitmapHeight As Integer
    Private intBitmapWidthMargin, intBitmapHeightMargin As Integer
    Const TEXT_TIMER_INTERVAL As Integer = 15
    Protected intCurrentGradientShift As Integer = 10
    Protected intGradiantStep As Integer = 5

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles MyBase.Load
        ' Lấy mảng các hình ảnh "chớp mắt"
      Dim i As Integer
      For i = 0 To 3
        arrImages(i) = New bitmap("..\Eye" & (i + 1).ToString & ".png")
      Next i

    End Sub

    ' Xử lý sự kiện thay đổi các nút Radio
    Private Sub RadioButtons_CheckedChanged(ByVal sender As Object, ByVal e As
  System.EventArgs) Handles optWink.CheckedChanged, optBall.CheckedChanged

      If optWink.Checked Then
        tmrAnimation.Interval = WINK_TIMER_INTERVAL
      ElseIf optBall.Checked Then
        tmrAnimation.Interval = BALL_TIMER_INTERVAL
      ElseIf optText.Checked Then
```

```
        tmrAnimation.Interval = TEXT_TIMER_INTERVAL
    End If
    OnResize(EventArgs.Empty)
End Sub

' Xử lý sự kiện Tick của Timer. Đây là thủ tục thực hiện hoạt hình
Protected Overridable Sub TimerOnTick(ByVal obj As Object, ByVal ea As
EventArgs) Handles tmrAnimation.Tick

    If optWink.Checked Then

        ' Lấy đối tượng Graphics của Form
        Dim grfx As Graphics = CreateGraphics()
        ' Gọi DrawImage, sử dụng Overload thứ 8, yêu cầu truyền vào hình ảnh
        ' sẽ được hiển thị, tọa độ X và Y (trung của tâm ảnh trên màn hình), và
        ' chiều rộng và chiều cao của ảnh.
        grfx.DrawImage(arrImages(intCurrentImage), _
            CInt((ClientSize.Width - arrImages(intCurrentImage).Width) / 2), _
            CInt((ClientSize.Height - arrImages(intCurrentImage).Height) / 2), _
            arrImages(intCurrentImage).Width, _
            arrImages(intCurrentImage).Height)
        ' Gọi phương thức Dispose (các đối tượng có cung cấp nó) sẽ
        ' tăng hiệu xuất chương trình
        grfx.Dispose()

        ' Duyệt qua các hình ảnh
        intCurrentImage += j
        If intCurrentImage = 3 Then
            ' Đây là ảnh cuối cùng, do đó làm ngược lại để mắt nhắm lại
            j = -1
        ElseIf intCurrentImage = 0 Then
            ' Đây là ảnh đầu tiên, do đó làm ngược lại để mở mắt
            j = 1
        End If
    ElseIf optBall.Checked Then

        ' Lấy đối tượng Graphics của Form
        Dim grfx As Graphics = CreateGraphics()
```

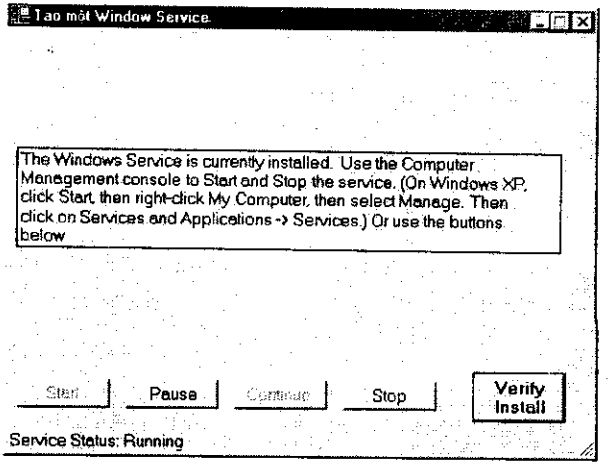
```
        ' Vẽ hình (bitmap) chứa quả banh lên Form
        grfx.DrawImage(bitmap, _
          CInt(intBallPositionX - intBallBitmapWidth / 2), _
          CInt(intBallPositionY - intBallBitmapHeight / 2), _
          intBallBitmapWidth, intBallBitmapHeight)
        grfx.Dispose()

        ' Tăng vị trí của quả banh sau khi đã vẽ lại
        intBallPositionX += intBallMoveX
        intBallPositionY += intBallMoveY

        ' Đổi hướng quả banh khi nó đụng vào đường biên
        If intBallPositionX + intBallRadiusX >= ClientSize.Width _
          Or intBallPositionX - intBallRadiusX <= 0 Then
          intBallMoveX = -intBallMoveX
          Beep()
        End If
        ' Gán giới hạn cho Y là 40 để banh không đụng vào các control phía trên
        If intBallPositionY + intBallRadiusY >= ClientSize.Height _
          Or intBallPositionY - intBallRadiusY <= 40 Then
          intBallMoveY = -intBallMoveY
          Beep()
        End If

      ElseIf optText.Checked Then

        ' Lấy đối tượng Graphics của Form
        Dim grfx As Graphics = CreateGraphics()

        ' Chọn kiểu font, văn bản, và xác định kích thước.
        Dim font As New font("Microsoft Sans Serif", 96, _
          FontStyle.Bold, GraphicsUnit.Point)
        Dim strText As String = "GDI+!"
        Dim sizfText As New SizeF(grfx.MeasureString(strText, font))

        ' Gán vị trí sẽ in văn bản ra: Chính giữa cửa sổ
        Dim ptfTextStart As New PointF( _
          CSng(ClientSize.Width - sizfText.Width) / 2, _
          CSng(ClientSize.Height - sizfText.Height) / 2)
```

```
        ' Gán vị trí bắt đầu và kết thúc của vùng tô, sau đó thay đổi giá trị này
        ' để thấy kết quả
        Dim ptfGradientStart As New PointF(0, 0)
        Dim ptfGradientEnd As New PointF(intCurrentGradientShift, 200)
        ' Tạo instance Brush sử dụng để vẽ văn bản
        Dim grBrush As New LinearGradientBrush(ptfGradientStart, _
          ptfGradientEnd, Color.Blue, BackColor)

        ' Vẽ văn bản chính giữa cửa sổ
        grfx.DrawString(strText, font, grBrush, ptfTextStart)
        grfx.Dispose()

        ' Dịch chuyển vùng tô, đảo nó lại khi đến một giá trị cụ thể
        intCurrentGradientShift += intGradiantStep
        If intCurrentGradientShift = 500 Then
          intGradiantStep = -5
        ElseIf intCurrentGradientShift = -50 Then
          intGradiantStep = 5
        End If
      End If
    End Sub

    ' Phương thức này Override phương thức OnResize trong class Control cơ sở,
    ' sự kiện này xảy ra khi control được định lại kích thước (trường hợp này là Form)
    Protected Overrides Sub OnResize(ByVal ea As EventArgs)

      If optWink.Checked Then
        ' Lấy đối tượng Graphics của Form và xóa các hết các hình ảnh đã có.
        Dim grfx As Graphics = CreateGraphics()
        ' Có thể sử dụng grfx.Clear(BackColor) hoặc Me.Invalidate() để xóa màn hình
        Me.Refresh()
        grfx.Dispose()
      ElseIf optBall.Checked Then
        Dim grfx As Graphics = CreateGraphics()
        grfx.Clear(BackColor)
        ' Gán bán kính của quả banh là phân số của chiều rộng hoặc
        ' chiều cao của client.
        Dim dblRadius As Double = Math.Min(ClientSize.Width / grfx.DpiX, _
          ClientSize.Height / grfx.DpiY) / intBallSize
```

```
        ' Gán chiều rộng và chiều cao của quả banh theo ĐPI của các trục X và Y
        intBallRadiusX = CInt(dblRadius * grfx.DpiX)
        intBallRadiusY = CInt(dblRadius * grfx.DpiY)

        grfx.Dispose()

        ' Gán khoảng cách di chuyển cho banh bằng 1 hoặc là phân số của kích
        ' thước banh. Giá trị càng lớn, banh di chuyển càng nhanh.
        intBallMoveX = CInt(Math.Max(1, intBallRadiusX / intMoveSize))
        intBallMoveY = CInt(Math.Max(1, intBallRadiusY / intMoveSize))
        intBitmapWidthMargin = intBallMoveX
        intBitmapHeightMargin = intBallMoveY

        ' Xác định kích thước chính xác của Bitmap
        intBallBitmapWidth = 2 * (intBallRadiusX + intBitmapWidthMargin)
        intBallBitmapHeight = 2 * (intBallRadiusY + intBitmapHeightMargin)

        ' Tạo một bitmap mới, truyền vào chiều rộng và chiều cao
        bitmap = New bitmap(intBallBitmapWidth, intBallBitmapHeight)

        ' Lấy đối tượng Graphics của Bitmap, xóa banh cũ và vẽ lại banh mới
        grfx = Graphics.FromImage(bitmap)
        With grfx
          .Clear(BackColor)
          .FillEllipse(Brushes.Red, New Rectangle(intBallMoveX, _
            intBallMoveY, 2 * intBallRadiusX, 2 * intBallRadiusY))
          .Dispose()
        End With
        ' reset vị trí của banh vào giữa cửa sổ
        intBallPositionX = CInt(ClientSize.Width / 2)
        intBallPositionY = CInt(ClientSize.Height / 2)
      ElseIf optText.Checked Then
        ' Lấy đối tượng Graphics của Form và xóa hình ảnh đã có
        Dim grfx As Graphics = CreateGraphics()
        grfx.Clear(BackColor)
      End If
    End Sub
End Class
```

59. Tạo một Windows Service

Chương trình này chỉ cách tạo một Windows Service (dịch vụ trên Windows) và cách triển khai nó.



Các chức năng chính:

Chương trình cho biết cách xây dựng một Windows Service bao gồm các phương thức tương ứng khi service Start, Stop, Pause, Continue, và khi service Stop do hệ thống shut down. Ngoài ra, solution này còn cung cấp một project để triển khai (đóng gói) Windows Service.

Chú ý:

- Sử dụng hệ điều hành có hỗ trợ Service. (Windows NT, 2000, XP, Windows .NET Server family).

Các bước xây dựng một Windows Service:

- **Tạo Windows Service**. Chỉ đơn giản là tạo một project Windows Service mới trong Visual Basic .NET. Ta được cung cấp các phương thức override OnStart và OnStop. Viết code cho các phương thức này, và các phương thức khác nếu cần (OnContinue và OnPause). Ta cũng cần gán các tham số tương ứng. Ví dụ, giá trị của CanPauseAndContinue phải gán là True nếu muốn cho phép các sự kiện tương ứng (Pause và Continue).
- **Thêm một Installer**. Có thể thêm một Installer bằng cách nhấn chuột phải vào cửa sổ thiết kế và chọn "Add Installer". Cần có Installer để thực hiện các thay đổi thích hợp trong registry, và để service hook vào các sự kiện đúng. Trong ServiceProcessInstaller ta cần xác định account sẽ start service. Chọn lựa phổ biến là LocalSystem và User. Trong

WindowsServiceInstaller ta cần chỉ định cách service sẽ start. Chọn Manual nếu muốn service sẽ start bởi người sử dụng. Chọn Automatic nếu muốn service start khi hệ điều hành startup.

- **Tạo bản phân phát**. Phương pháp thuận tiện là sử dụng Deployment Project để tạo một file MSI. Trước tiên, thêm một Deployment Project (wizard) vào solution có chứa project Windows Service. Sau đó tạo một Custom Action cho Primary Output của project Windows Service (nhấn chuột phải vào project Deployment và chọn View->Custom Actions. Sau đó nhấn chuột phải vào Custom Actions và chọn Add Custom Action. Khi hộp thoại mở ra, nhấp đúp chuột vào Application Folder và chọn Primari Ouput của project Windows Service. Thao tác này thêm Custom Action này vào bốn Installer Action: Install, Commit, Rollback và Uninstall). Đến đây có thể Build solution, và file MSI kết quả là file cài đặt Windows Service.

Thứ tự chạy ví dụ này:

- Cài đặt Windows Service Demo bằng cách nhấp đúp chuột vào file **Windows Service - Time Track.msi**
- Nhấp đúp vào file **Tao mot Windows Service.sln**
- Xem project **Windows Service Demo** để thấy code tạo một Windows Service.
- Xem **Windows Service - Time Track Install** để thấy một ví dụ tạo project triển khai cho một Windows Service.

Solution: **Tao mot Windows Service**

Project: **Tao mot Windows Service**

Project này cung cấp một giao diện để kiểm tra trạng thái và điều khiển Windows Service.

frmMain.vb

```
Option Strict On
Imports System.ServiceProcess

Public Class frmMain
    Inherits System.Windows.Forms.Form
```

```
Private isServiceInstalled As Boolean = False
Private isServiceRunning As Boolean
Private myService As ServiceController ' Sử dụng để tham chiếu đến Service

Friend WithEvents lblInstructions As System.Windows.Forms.Label
Friend WithEvents lblInstallationStatus As System.Windows.Forms.Label
Friend WithEvents btnStart As System.Windows.Forms.Button
Friend WithEvents btnPause As System.Windows.Forms.Button
Friend WithEvents btnContinue As System.Windows.Forms.Button
Friend WithEvents btnStop As System.Windows.Forms.Button
Friend WithEvents sbrServiceStatus As System.Windows.Forms.StatusBar
Friend WithEvents tmrCheckStatus As System.Windows.Forms.Timer
Friend WithEvents btnVerifyInstall As System.Windows.Forms.Button

' Code phát sinh form được bỏ qua

    ' Ra lệnh cho Service tiếp tục (Continue)
    Private Sub btnContinue_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnContinue.Click
        myService.Continue()
        UpdateServiceStatus()
    End Sub

    ' Ra lệnh service tạm dừng (Pause)
    Private Sub btnPause_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnPause.Click
        myService.Pause()
        UpdateServiceStatus()
    End Sub

    ' Ra lệnh service dừng hoàn toàn (Stop)
    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnStop.Click

        myService.Stop()
        UpdateServiceStatus()

    End Sub
```

```
' Ra lệnh cho service bắt đầu (Start)
Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click
    myService.Start()
    UpdateServiceStatus()
End Sub

' Kiểm tra xem service đã được cài đặt chưa
Private Sub btnVerifyInstall_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnVerifyInstall.Click
    CheckServiceInstallation()
End Sub

' Khi Form load thì kiểm tra Service đã được cài đặt chưa và trạng thái của nó là gì.
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    CheckServiceInstallation()
    UpdateServiceStatus()
    Me.tmrCheckStatus.Enabled = True
End Sub

' Cập nhật lại trạng thái của service theo chu kỳ thời gian
Private Sub tmrCheckStatus_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrCheckStatus.Tick
    UpdateServiceStatus()
End Sub

Private Sub CheckServiceInstallation()
    ' Kiểm tra service đã được cài đặt chưa
    Dim installedServices() As ServiceController
    Dim tmpService As ServiceController
    Dim i As Integer = 0

    ' Tắt Timer trong đoạn code này
    Me.tmrCheckStatus.Enabled = False

    installedServices = ServiceController.GetServices()
```

```
    For Each tmpService In installedServices
      If tmpService.DisplayName = " TimeTrackerService" Then
        isServiceInstalled = True
        Me.lblInstallationStatus.Text = _
          "The Windows Service is currently installed.  Use the " & _
          "Computer Management console to Start and Stop the " & _
          "service. (On Windows XP, click Start, then right-click " & _
          "My Computer, then select Manage. Then click on " & _
          "Services and Applications -> Services.) Or use the " & _
          "buttons below"
        ' Gán Service vào myService để sử dụng sau này
        myService = tmpService
      End If
    Next tmpService

    Me.tmrCheckStatus.Enabled = True
End Sub

Private Sub UpdateServiceStatus()
    Me.tmrCheckStatus.Enabled = False

    If Not (myService Is Nothing) Then
        ' Tạo lại myService, nếu không, trạng thái của myService không bao
        ' giờ thay đổi.
      CheckServiceInstallation()

      Select Case myService.Status
        Case ServiceControllerStatus.Stopped
          Me.btnContinue.Enabled = False
          Me.btnPause.Enabled = False
          Me.btnStart.Enabled = True
          Me.btnStop.Enabled = False
        Case ServiceControllerStatus.Running
          Me.btnContinue.Enabled = False
          Me.btnPause.Enabled = True
          Me.btnStart.Enabled = False
          Me.btnStop.Enabled = True
        Case ServiceControllerStatus.Paused
          Me.btnContinue.Enabled = True
          Me.btnPause.Enabled = False
```

```
            Me.btnStart.Enabled = False
            Me.btnStop.Enabled = True
        Case Else
            ' Trường hợp này có thể xảy ra khi một hành động đang chạy. Trong
            ' trường hợp này không cho phép người sử dụng nhấn vào bất kỳ nút nào.
            Me.btnContinue.Enabled = False
            Me.btnPause.Enabled = False
            Me.btnStart.Enabled = False
            Me.btnStop.Enabled = False
    End Select

    ' Hiển thị trạng thái trên Status Bar
    Me.sbrServiceStatus.Text = "Service Status: " + myService.Status.ToString()
  Else
    Me.btnContinue.Enabled = False
    Me.btnPause.Enabled = False
    Me.btnStart.Enabled = False
    Me.btnStop.Enabled = False
  End If
  Me.tmrCheckStatus.Enabled = True
 End Sub
End Class
```

Project: **Windows Service Demo**

TimeTracker Windows Service.vb

```
Option Strict On
Imports System.ServiceProcess

Public Class TimeTrackerService
  Inherits System.ServiceProcess.ServiceBase

  ' Khai báo các biến đối tượng DataTime và TimeSpan
  Private timeStart As DateTime ' Sử dụng để biết thời gian bắt đầu của service
  Private timeEnd As DateTime ' Sử dụng để biết thời gian kết thúc của service
  Private timeElapsed As New TimeSpan(0)
  ' Sử dụng để tính toán khoảng thời gian giữ timeEnd và timeStart
  Private timeDifference As TimeSpan
  Private isPaused As Boolean = False ' Để biết service có tạm dừng hay không
```

```
#Region " Component Designer generated code "

    Public Sub New()
        MyBase.New()

        ' This call is required by the Component Designer.
        InitializeComponent()

        ' Add any initialization after the InitializeComponent() call

    End Sub

    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    ' Điểm xử lý chính của tiến trình (process)
    <MTAThread()> _
    Shared Sub Main()
        Dim ServicesToRun() As System.ServiceProcess.ServiceBase

        ' Có thể nhiều NT Service chạy trong cùng một process. Để thêm một service
        ' khác, thay đổi dòng bên dưới để tạo đối tượng service thứ hai. Ví dụ:
        '
        '   ServicesToRun = New System.ServiceProcess.ServiceBase () {New _
        '                       TimeTrackerService, New MySecondUserService}
        '
        ServicesToRun = New System.ServiceProcess.ServiceBase() {New _
                                TimeTrackerService()}
        System.ServiceProcess.ServiceBase.Run(ServicesToRun)
    End Sub
```

```
    'Required by the Component Designer
    Private components As System.ComponentModel.IContainer

    ' NOTE: The following procedure is required by the Component Designer
    ' It can be modified using the Component Designer.
    ' Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        '
        'TimeTrackerService
        '
        Me.CanPauseAndContinue = True
        Me.CanShutdown = True
        Me.ServiceName = "TimeTrackerService"

    End Sub
#End Region

    ' OnContinue xảy ra khi người sử dụng yêu cầu một service đã tạm dừng tiếp tục
    ' hoạt động. Để có thể gọi phương thức này thì thuộc tính CanPauseAndContinue
    ' phải có giá trị là True (mặc nhiên là False).
    Protected Overrides Sub OnContinue()
        ' Bắt đầu tính lại thời gian hoạt động. Gán timeStart bằng thời gian hiện hành,
        ' và gán isPaused = False.
        If isPaused Then
            timeStart = DateTime.Now
        End If
        isPaused = False

        EventLog.WriteEntry("Service Continued at " + _
            DateTime.Now().ToString())
    End Sub

    ' OnPause xảy ra khi người sử dụng yêu cầu Service tạm dừng. Để có thể gọi
    ' phương thức này thì thuộc tính CanPauseAndContinue phải có giá trị
    ' là True (mặc nhiên là False).
    Protected Overrides Sub OnPause()
        ' Lưu lại thời gian đã trôi qua vào timeElapsed, và đợi người sử dụng Stop
        ' hoặc Continue service
        timeEnd = DateTime.Now()
```

```
    If Not isPaused Then
        timeDifference = timeEnd.Subtract(timeStart)
        timeElapsed = timeElapsed.Add(timeDifference)
    End If
    isPaused = True

    EventLog.WriteEntry("Service was Paused at " + _
        DateTime.Now().ToString())
End Sub

' OnShutdown xảy ra khi shut down máy tính và service chưa được Stop.
' Thuộc tính CanShutdown phải có giá trị là True.
Protected Overrides Sub OnShutdown()
    ' Ép service phải được stop
    Me.OnStop()
End Sub

' OnStart được gọi khi Service được start
Protected Overrides Sub OnStart(ByVal args() As String)

    ' Reset timeElapsed bằng 0. Điều này cần thiết vì người sử dụng có
    ' thể ReStart mà không tạm dừng hoặc dừng service.
    timeElapsed = New TimeSpan(0)

    ' Khởi tạo thời gian bắt đầu (Start)
    timeStart = DateTime.Now()
    isPaused = False

    EventLog.WriteEntry("Service was Started at " + _
        timeStart.ToString())

End Sub

' OnStop được gọi bất cứ khi nào service được dừng lại
Protected Overrides Sub OnStop()

    ' Tính toán các khoảng thời gian cần thiết
    timeEnd = DateTime.Now()
```

```
    If Not isPaused Then
      timeDifference = timeEnd.Subtract(timeStart)
      timeElapsed = timeElapsed.Add(timeDifference)
    End If

    EventLog.WriteEntry("Service was Stopped at " + _
      timeEnd.ToString())
    EventLog.WriteEntry("Service ran for a total time of " + _
      timeElapsed.ToString())
  End Sub

End Class
```

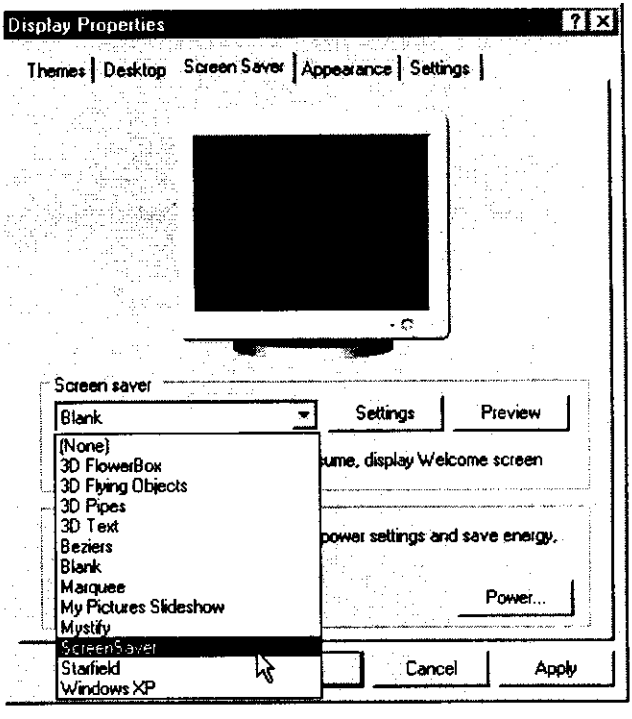
ProjectInstaller.vb

```
Option Strict On
Imports System.ComponentModel
Imports System.Configuration.Install

' Cần phải tạo Installer để service được cài đặt vào hệ thống là một service.
<RunInstaller(True)> Public Class ProjectInstaller
  Inherits System.Configuration.Install.Installer

#Region " Component Designer generated code "

  Public Sub New()
    MyBase.New()

    'This call is required by the Component Designer.
    InitializeComponent()

    'Add any initialization after the InitializeComponent() call

  End Sub

  'Installer overrides dispose to clean up the component list.
  Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
      If Not (components Is Nothing) Then
        components.Dispose()
```

```
        End If
      End If
      MyBase.Dispose(disposing)
    End Sub

    'Required by the Component Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Component Designer
    'It can be modified using the Component Designer.
    'Do not modify it using the code editor.

    Friend WithEvents myServiceProcessInstaller As
    System.ServiceProcess.ServiceProcessInstaller
    Friend WithEvents myWindowsServiceInstaller As
    System.ServiceProcess.ServiceInstaller
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
      Me.myServiceProcessInstaller = New
    System.ServiceProcess.ServiceProcessInstaller()
      Me.myWindowsServiceInstaller = New System.ServiceProcess.ServiceInstaller()
      '
      'myServiceProcessInstaller
      '
      Me.myServiceProcessInstaller.Account =
    System.ServiceProcess.ServiceAccount.LocalSystem
      Me.myServiceProcessInstaller.Password = Nothing
      Me.myServiceProcessInstaller.Username = Nothing
      '
      'myWindowsServiceInstaller
      '
      Me.myWindowsServiceInstaller.ServiceName = " TimeTrackerService"
      '
      'ProjectInstaller
      '
      Me.Installers.AddRange(New System.Configuration.Install.Installer() _
                {Me.myServiceProcessInstaller, Me.myWindowsServiceInstaller})
    End Sub

  #End Region
  End Class
```

60. Tạo một Screen Saver với GDI+

Chương trình này chỉ một số kỹ thuật tạo một chương trình bảo vệ màn hình bằng Visual Basic .NET.



Các chức năng chính:

- Tạo chương trình Screen Saver làm đầy màn hình bằng một form, và vẽ các hình ảnh lên form đó.
- Thiết lập một file Options để người sử dụng định nghĩa cách Screen Saver sẽ hoạt động.
- Tương tác với các sự kiện chuột và bàn phím để đóng Screen Saver.

Chú ý:

Khi tự viết các Screen Saver có một số điều cần nhớ: Bảo đảm là các thuộc tính của Form có giá trị như sau:

- FormBorderStyle = None
- WindowState = Maximized
- MaximizeBox = False
- MinimizeBox = False
- ShowInTaskbar = False

Sau đó, trước khi có thể sử dụng được Screen Saver mới này, ta phải đổi phần mở rộng từ .EXE thành .SCR, và chép nó vào thư mục hệ thống của Windows (mặc nhiên trong WinXP là C:\Windows\System32).

Đến đây ta có thể sử dụng trình điều khiển Screen Saver của Windows để chọn Screen Saver này làm trình bảo vệ màn hình mặc nhiên của Windows.

Trong Solution này có hai Project: một Project để chép file Screen Saver vào hệ thống: **Tao Screensaver su dung GDI+**, và một Project tạo một Screen Saver: **Screensaver**.

Project: **Screensaver**

frmScreenSaver.vb

```
Option Strict On

' Đây là form chính của SS. Nó làm tất cả công việc vẽ màn hình, xử lý khi nào
' kết thúc và trả điều khiển về cho Windows.
Public Class frmSceenSaver
    Inherits System.Windows.Forms.Form

    ' Khai báo các biến sẽ dùng cho SS
    Private m_Graphics As Graphics
    Private m_Random As New Random()
    ' Đối tượng Options chứa thông tin về các chọn lựa do người sử dụng chọn
    Private m_Options As New Options()
    ' Sử dụng để xác định có di chuyển chuột khỏi vị trí ban đầu
    Private m_IsActive As Boolean = False
    ' Sử dụng để xác định chuột có thực sự đã di chuyển
    Private m_MouseLocation As Point

    Friend WithEvents tmrUpdateScreen As System.Windows.Forms.Timer

' Code phát sinh form được bỏ qua

    ' Đóng SS khi người sử dụng di chuyển chuột. Sự kiện này rất nhạy cảm, do đó
    ' ta cần kiểm tra xem chuột đã di chuyển ít nhất một số pixel hay chưa.
    Private Sub frmSceenSaver_MouseMove(ByVal sender As Object, ByVal e As
    System.Windows.Forms.MouseEventArgs) Handles MyBase.MouseMove
```

```
    ' Nếu MouseLocation có giá trị (0,0) thì lưu giá trị vị trí hiện hành vào nó.
    ' Nếu có giá trị khác, kiểm tra xem người sử dụng đã di chuyển chuột được
    ' ít nhất là 10 pixel thì thoát SS.
    If Not m_IsActive Then
      Me.m_MouseLocation = New Point(e.X, e.Y)
      m_IsActive = True
    Else
      If Math.Abs(e.X - Me.m_MouseLocation.X) > 10 Or _
        Math.Abs(e.Y - Me.m_MouseLocation.Y) > 10 Then
          ' Chuột đã được di chuyển do đó thoát chương trình
        Application.Exit()
      End If
    End If
  End Sub

  Private Sub frmSceenSaver_Load(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles MyBase.Load
    ' Tạo đối tượng Graphics dùng để vẽ
    Me.m_Graphics = Me.CreateGraphics()
    ' Lấy các thông số Options đã lưu. Nếu chưa có file Options thì sẽ tạo nó
    m_Options.LoadOptions()
    ' Chọn tốc độ dựa trên giá trị đã được lưu
    Select Case m_Options.Speed
      Case "Slow"
        Me.tmrUpdateScreen.Interval = 500
      Case "Fast"
        Me.tmrUpdateScreen.Interval = 100
      Case Else
        Me.tmrUpdateScreen.Interval = 200
    End Select

    ' Cho Timer hoạt động
    Me.tmrUpdateScreen.Enabled = True
  End Sub

  ' Đóng SS nếu người sử dụng nhấn chuột
  Private Sub frmSceenSaver_MouseDown(ByVal sender As Object, ByVal e As
  System.Windows.Forms.MouseEventArgs) Handles MyBase.MouseDown
    Application.Exit()
  End Sub
```

```
        ' Vẽ một hình Elip hay hình Chữ nhật dựa trên chọn lựa của người sử dụng
        If m_Options.Shape = "Ellipses" Then
            m_Graphics.FillEllipse(New SolidBrush(myColor), myRect)
        Else
            m_Graphics.FillRectangle(New SolidBrush(myColor), myRect)
        End If
    End Sub

    ' Thủ tục này được thực hiện đầu tiên khi SS chạy.
    <STAThread()> Shared Sub Main(ByVal args As String())

        ' Kiểm tra xem có tham số nào truyền vào hay không. Nếu không, có nghĩa
        ' là người sử dụng nhấp đúp chuột vào file .scr
        If args.Length > 0 Then
            ' Có truyền tham số vào. Windows sẽ tự động truyền một trong các
            ' tham số "/s", "/p" hoặc "/c" tùy vào thao tác.

            ' Khi nhấn nút Preview
            If args(0).ToLower = "/p" Then
                ' Chỉ đơn giản thoát chương trình
                Application.Exit()
            End If

            ' Hiển thị  hộp thoại các chọn lựa (Options)
            If args(0).ToLower.Trim().Substring(0, 2) = "/c" Then
                ' Tạo một form frmOptions và hiển thị nó
                Dim userOptionsForm As New frmOptions()
                userOptionsForm.ShowDialog()

                ' Thoát ứng dụng
                Application.Exit()
            End If

            ' Chỉ đơn giản là thực  hiện SS
            If args(0).ToLower = "/s" Then
                ' Tạo một form frmSceenSaver và hiển thị nó
                Dim screenSaverForm As New ScreenSaver.frmSceenSaver()
                screenSaverForm.ShowDialog()
```

```
            ' Thoát ứng dụng khi form được đóng lại
            Application.Exit()
        End If

    Else

        ' Tạo một form frmSceenSaver và hiển thị nó
        Dim screenSaverForm As New ScreenSaver.frmSceenSaver()
        screenSaverForm.ShowDialog()

        Application.Exit()
    End If
  End Sub
End Class
```

Optoins.vb

```
Option Strict On
Imports System.Xml.Serialization
Imports System.Xml
Imports System.IO

' Đây là class dùng để chứa các thông tin Options.
<Serializable()> Public Class Options

    ' Khai báo các biến và gán các giá trị mặc nhiên cho chúng
    Private m_Speed As String = "Fast"
    Private m_Shape As String = "Ellipse"
    Private m_IsTransparent As Boolean = True

    ' Gán vào OptionsPath đường dẫn cố định đến file lưu thông tin
    Private OptionsPath As String = Environment.CurrentDirectory & _
                                    "\Screen Saver Options.opt"


    Public Property IsTransparent() As Boolean
        Get
            Return m_IsTransparent
        End Get
```

```
    Set(ByVal Value As Boolean)
      m_IsTransparent = Value
    End Set
  End Property

  Public Property Shape() As String
    Get
      Return m_Shape
    End Get
    Set(ByVal Value As String)
      m_Shape = Value
    End Set
  End Property

  Public Property Speed() As String
    Get
      Return m_Speed
    End Get
    Set(ByVal Value As String)
      m_Speed = Value
    End Set
  End Property

  Public Function IsOptionFileExisting() As Boolean
    Dim myIO As New System.IO.FileInfo(OptionsPath)
    Return myIO.Exists()
  End Function

  Public Sub LoadOptions()

    Dim myOptions As New Options() ' Tạo đối tượng Options

    If myOptions.IsOptionFileExisting() Then

      ' Lấy các giá trị Options
      ' Tạo đối tượng XmlSerializer để lấy các giá trị Options
      Dim mySerializer As New XmlSerializer(GetType(Options))
      ' Tạo một StreamReader để trỏ tới file Options
      Dim myTextReader As New StreamReader(OptionsPath)
```

```
        ' Tạo một đối tượng XmlTextReader để thực sự đọc các Options
        Dim myXmlReader As New Xml.XmlTextReader(myTextReader)

        If mySerializer.CanDeserialize(myXmlReader) Then
            myOptions = CType(mySerializer.Deserialize(myXmlReader), Options)
        Else
            ' Tạo file option mới
            myOptions.SaveOptions()
        End If

        myXmlReader.Close()
        myTextReader.Close()

        Me.Speed = myOptions.Speed
        Me.IsTransparent = myOptions.IsTransparent
        Me.Shape = myOptions.Shape

    End If
End Sub

' Lưu các các option do người dùng chọn xuống đĩa
Public Sub SaveOptions()

    ' Tạo một StreamWriter ghi chồng lên file đã tồn tại
    Dim myWriter As New System.IO.StreamWriter(OptionsPath)

    ' Tạo một đối tượng chuyển đối tượng sang dạng file
    Dim myXmlSerializer As New XmlSerializer(Me.GetType())

    ' Chuyển đối tượng Options hiện hành sang file
    myXmlSerializer.Serialize(myWriter, Me)

    myWriter.Close()

End Sub

End Class
```

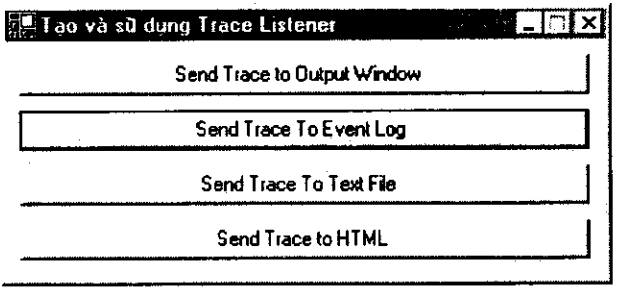
Project: **Tao Screensaver su dung GDI+**

frmMain.vb

```
Option Strict On
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    ' Nút này đóng ứng dụng
    Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnClose.Click
        Application.Exit()
    End Sub

    ' Nút này cài đặt SS vào hệ thống bằng cách chép file SCR vào thư mục
    ' hệ thống của Windows
    Private Sub btnInstall_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnInstall.Click

        Dim fileName As String = "ScreenSaver.scr"
        Dim sourceFile As String = Environment.CurrentDirectory & "\..\" & fileName
        Dim destFile As String = Environment.SystemDirectory & "\" & fileName

        ' Tiến hành chép, nếu có lỗi thì thông báo
        Try
            File.Copy(sourceFile, destFile, True)
        Catch ex As Exception
            MsgBox(ex.ToString(), MsgBoxStyle.Exclamation, Me.Text)
        End Try

    End Sub

End Class
```

61. Tạo và sử dụng các Trace Listener

Chương trình này chỉ cách sử dụng các trình theo dõi có sẵn của hệ thống để theo dõi và bắt lỗi code, và cách xây dựng một trình theo dõi riêng.



Các chức năng chính:

- Cách sử dụng trace listener mặc nhiên để xuất các thông báo ra cửa sổ Output trong IDE.
- Cách ghi thông báo vào Application event log.
- Cách ghi thông báo vào một file văn bản.
- Cách tạo một trace listener riêng, nó sẽ ghi các thông báo vào một tài liệu HTML.

Project: **Tao va su dung Trace Listener**

frmMain.vb

```
Option Strict On
Imports System
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub cmdTraceToEventLog_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles cmdTraceToEventLog.Click
        ' Thủ tục này chỉ cách tạo và thêm một trace listener có nhiệm vụ gửi các
        ' thông báo vào Application Event Log.

        ' Tạo một instance EventLog và gán cho nó một nguồn cần theo dõi
        Dim myLog As New EventLog()
```

```
    myLog.Source = Me.Text

    ' Tạo một trace listener cho Application Event Log.
    Dim tlEventLog As New EventLogTraceListener(myLog)

    Trace.Listeners.Clear()

    ' Cho tlEventLog vào tập hợp Listeners
    Trace.Listeners.Add(tlEventLog)

    ' Ghi kết quả vào event log.
    Trace.WriteLine("This is a test of event log tracing")

End Sub

Private Sub cmdTraceToFile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdTraceToFile.Click
    ' Thêm một trace listener có nhiệm vụ gửi thông báo vào một file văn bản có
    ' tên TraceOutput.txt. File này được tạo trong cùng thư mục với file .EXE.

    ' Tạo file xuất có tên TraceOutput.txt
    Dim traceFile As System.IO.Stream = File.Create("TraceOutput.txt")

    ' Khai báo một số biến để hiển thị trong trace output
    Dim ProductType As String = "Computer"
    Dim Price As Integer = 2000
    Dim UnitsInStock As Integer = 23

    ' Tạo một instance của TextWriterTraceListener để xuất ra file văn bản
    Dim tlTextFile As New TextWriterTraceListener(traceFile)

    ' Xóa các phần tử trong collection
    Trace.Listeners.Clear()
    Trace.Listeners.Add(tlTextFile)

    Trace.WriteLine("******* Trace Output to File *******")
    Trace.WriteLine("Trace output for " & Me.Text)
    Trace.WriteLine("")
```

```
    Trace.Indent()
    Trace.WriteLine("This line is indented")
    Trace.WriteLine("Product Type = " & ProductType)
    Trace.WriteLine("Price = $" & Price)

    Trace.WriteLineIf(Price > 1800, "Price > $1800")
    Trace.Unindent()

    tlTextFile.Flush()
    tlTextFile.Close()

    ' Hiển thị nội dung của file văn bản
    System.Diagnostics.Process.Start("Notepad.exe", "TraceOutput.txt")
End Sub

Private Sub cmdTraceToHTML_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdTraceToHTML.Click
    ' Tạo một TraceListener gửi thông điệp ra một file HTML
    ' có tên TraceOutputHTML.htm.

    ' Tạo file để lưu kết quả
    Dim traceFile As System.IO.Stream = File.Create("TraceOutputHTML.htm")

    Dim ProductType As String = "Computer"
    Dim Price As Integer = 2000
    Dim UnitsInStock As Integer = 23

    Dim tlHTMLFile As New HTMLTraceListener(traceFile)
    tlHTMLFile.WriteHeader("HTML Trace Output Example for " & Me.Text)

    Trace.Listeners.Clear()
    Trace.Listeners.Add(tlHTMLFile)

    Trace.Indent()
    Trace.WriteLine("This line is indented")
    Trace.Unindent()
    Trace.WriteLine("Product Type = " & ProductType)
    Trace.WriteLine("Price = $" & Price)

    Trace.WriteLineIf(Price > 1800, "Price > $1800")
```

```
        tlHTMLFile.Flush()
        tlHTMLFile.Close()

        ' Hiển thị file HTML
        System.Diagnostics.Process.Start("TraceOutputHTML.htm")
    End Sub

    Private Sub cmdTraceToOutput_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles cmdTraceToOutput.Click
        ' Sử dụng trace listener mặc nhiên gửi trace output ra cửa sổ Output của IDE

        Dim ProductType As String = "Computer"
        Dim Price As Integer = 2000
        Dim UnitsInStock As Integer = 23

        ' Tạo một instance của trace listener mặc nhiên
        Dim tlDefault As New DefaultTraceListener()

        ' Xóa các trace listener ra khỏi collection
        Trace.Listeners.Clear()

        ' Thêm trace listener mặc nhiên vào collection
        Trace.Listeners.Add(tlDefault)

        Trace.WriteLine("******* Trace OutPut Start *******")
        Trace.WriteLine("Output window trace information")

        Trace.Indent()
        Trace.WriteLine("This line is indented")
        Trace.WriteLine("Product Type = " & ProductType)
        Trace.WriteLine("Price = $" & Price)

        Trace.WriteLineIf(Price > 1800, "Price > $1800")
        Trace.Unindent()
        Trace.WriteLine("*******  Trace Output End  *******")

    End Sub

End Class
```

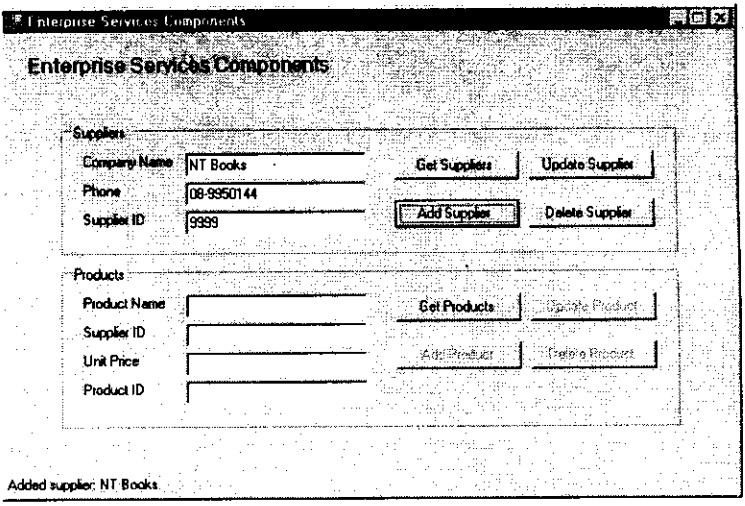
HTMLTraceListener.vb

```
Option Strict On
Imports System
Imports System.IO

' Class này được tạo để xuất các thông điệp vào một file HTML

Public Class HTMLTraceListener
    Inherits System.Diagnostics.TextWriterTraceListener

    ' Sử dụng constructor của TextWriterTraceListener để ghi vào một stream
    Public Sub New(ByVal stream As Stream)
        MyBase.New(stream)
    End Sub

    Public Sub WriteHeader(ByVal Title As String)
        ' Phương thức mới này thêm tựa đề cho tài liệu HTML
        Writer.WriteLine("<head>")
        Writer.WriteLine("<title>" & Title & "</title>")
        Writer.WriteLine("</head>")
        Writer.WriteLine("<H2>" & Title & "</H2>")
        Writer.WriteLine("<P><HR>")

    End Sub

    Protected Overloads Overrides Sub WriteIndent()
        ' Override WriteIndent để xử lý thụt đầu dòng theo định dạng HTML
        Dim i As Integer
        Dim j As Integer
        If IndentLevel > 0 Then
            For i = 1 To IndentLevel
                For j = 1 To IndentSize
                    Writer.Write(" ")
                Next j
            Next i
        End If

    End Sub
```

```
    Public Overloads Overrides Sub WriteLine(ByVal message As String)
        ' Override WriteLine Method để thêm các tag vào các thông điệp
        Writer.Write("<B>" & Now() & " - ")

        If NeedIndent Then
            WriteIndent()
        End If

        Writer.WriteLine(message & "</B><BR>")
    End Sub
End Class
```

62. Tạo Enterprise Service Component

Chương trình này chỉ cách tạo và làm việc với các Enterprise Service Component.



Các chức năng chính:

Ứng dụng này đáp ứng một hệ thống để duy trì một database gồm các nhà cung cấp (Suppliers) và các sản phẩm (Products). Nó cung cấp các chức năng để thêm, cập nhật, truy xuất và xóa các nhà cung cấp và các sản phẩm. Chú ý, tất cả các truy xuất dữ liệu đều được giả lập.

- Gọi COM+ Service để thực hiện các phiên giao dịch (transaction), just-in-time activation, object pooling và object construction.
- Sử dụng Dynamic Registration để thêm các Enterprise Components vào COM+ Catolog.

Solution này gồm hai project: **EnterpriseComponents** cung cấp hai class Supplier và Product làm cơ sở cho các component trong chương trình này. **Create Enterprise Services Components** sử dụng các component trên.

Project: **EnterpriseComponents**

AssemblyInfo.vb

```
Imports System.Reflection
Imports System.Runtime.InteropServices

' Lệnh Imports này cho phép sử dụng các thuộc tính như "ApplicationNameAttribute"
' và "ApplicationActivation", chú ý là cần thêm Reference đến Namspace này
' để import nó.
Imports System.EnterpriseServices

' ApplicationNameAttribute quy định tên (Name) của COM+
<Assembly: ApplicationNameAttribute("Enterprise Services Components")>

' ApplicationIDAttribute cung cấp COM+ một GUID như là định danh của ứng dụng
<Assembly: ApplicationIDAttribute("F3F4E0DA-6712-4AA9-9F48-871A81FD2844")>

' ApplicationActivation chỉ định các components được load ở đâu khi chúng
' được kích hoạt
' Library : Các component chạy trong tiến trình của chương trình tạo chúng
' Server : Các component chạy trong một tiến trình của hệ thống, dllhost.exe
<Assembly: ApplicationActivation(ActivationOption.Library)>

' DescriptionAttribute được hiển thị trong phần thuộc tính của ứng dụng
' trong COM+ Explorer
<Assembly: DescriptionAttribute("Data Access Components")>

' AssemblyKeyFile chỉ định file chứa cặp khóa Public/Private. File này phải
' nằm trong thư mục chứa ứng dụng.
'
' file .snk được phát sinh ở cửa sổ dòng lệnh bằng lệnh sn.exe, như sau:
' sn -k KeyFile.snk
<Assembly: AssemblyKeyFile("..\..\KeyFile.snk")>
```

Product.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.EnterpriseServices

' Các thuộc tính này thiết lập component để nó như là COM+
'
' Transaction(TransactionOption.Required)
'     Component này đòi hỏi có Transaction
'
' JustInTimeActivation(True)
'     COM+ sẽ kích hoạt đối tượng và ngưng hoạt động khi cần, giải phóng tài
'     nguyên khi đối tượng không đang trong sử dụng.
'
' ObjectPooling(Enabled:=True, MinPoolSize:=2, MaxPoolSize:=10)
'     Component sử dụng Object Pooling, cho phép tạo các đối tượng mà không bị hủy
'     khi chúng được giải phóng. Object Pooling tăng hiệu xuất chủa ứng dụng đối với
'     các đối tượng "tốn nhiều chi phí" để tạo nó. Component này sẽ giữ lại ít nhất
'     hai đối tượng và nhiều nhất là 10 đối tượng vào một thời điểm.
'
' ConstructionEnabledAttribute(Default:="Server=localhost;
' DataBase=Northwind;Integrated Security=SSPI")

<Transaction(TransactionOption.Required), _
JustInTimeActivation(True), _
ObjectPooling(Enabled:=True, MinPoolSize:=2, MaxPoolSize:=10), _
ConstructionEnabledAttribute(Default:="Server=localhost;DataBase=Northwind;Integr
  ated Security=SSPI")> _
Public Class Product
    Inherits ServicedComponent

    Private m_strConnectionString As String


    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"
```

```
    Protected Const MSDE_CONNECTION_STRING As String = _
      "Server=(local)\NetSDK;" & _
      "DataBase=Northwind;" & _
      "Integrated Security=SSPI"
    ' Bộ khởi tạo mặc nhiên
    Public Sub New()
      MyBase.New()
      ' Gán chuỗi kết nối mặc nhiên.
      m_strConnectionString = SQL_CONNECTION_STRING
      DoTracing("Product instantiated at " & DateTime.Now.ToLongTimeString)
    End Sub

    ' Áp dụng phương thức Construct chỉ khi component là một phần của
    ' một ứng dụng COM+
    Protected Overrides Sub Construct(ByVal constructString As String)
      ' Phương thức này được gọi ngay sau khi constructor được thực hiện
      If constructString.Length > 0 Then
        m_strConnectionString = constructString
      End If
    End Sub

    ' Giả lập gọi đến một stored procedure để thêm một product vào bảng Products.
    ' AutoCompleteAttribute: nếu procedure tồn tại mà không có exception
    ' thì SetComplete tự động được gọi, nếu khác thì SetAbort được gọi
    <AutoCompleteAttribute(True)> _
    Public Sub AddProduct(ByVal ProductName As String, ByVal SupplierID As
    Integer, ByVal UnitPrice As Decimal)
      '...
      Try
        'cmd = New SqlCommand("AddProduct", cnn)
        '...
        'cmd.ExecuteNonQuery()
        DoTracing("Product.AddProduct called at " & _
                                DateTime.Now.ToLongTimeString)
      Catch exp As Exception
        Throw New Exception(exp.Message, exp.InnerException)
      End Try
    End Sub
```

```
    ' Giả lập gọi đến SP xóa một Product
    <AutoCompleteAttribute(True)> _
    Public Sub DeleteProduct(ByVal ProductID As Integer)
        '...
        Try
            'cmd = New SqlCommand("DeleteProduct", cnn)
            '...
            'cmd.ExecuteNonQuery()
            DoTracing("Product.DeleteProduct called at " & _
                                DateTime.Now.ToLongTimeString)
        Catch exp As Exception
            Throw New Exception(exp.Message, exp.InnerException)
        End Try
    End Sub

    ' Giả lập gọi một SP lấy danh sách của tất cả Product
    <AutoCompleteAttribute(True)> _
    Public Function GetProducts() As DataTable
        '...
        Try
            '...
            'da.Fill(dt)
            DoTracing("Product.GetProducts called at " & _
                                DateTime.Now.ToLongTimeString)
            'Return dt
        Catch exp As Exception
            Throw New Exception(exp.Message, exp.InnerException)
        End Try
    End Function

    ' Giả lập gọ SP cập nhật một product
    <AutoCompleteAttribute(True)> _
    Public Sub UpdateProduct(ByVal ProductID As Integer, ByVal ProductName As
    String, ByVal SupplierID As Integer, ByVal UnitPrice As Decimal)
        '...
        Try
            'cmd = New SqlCommand("UpdateProduct", cnn)
            '...
            'cmd.ExecuteNonQuery()
```

```
        DoTracing("Product.UpdateProduct called at " & _
                                DateTime.Now.ToLongTimeString)
    Catch exp As Exception
        Throw New Exception(exp.Message, exp.InnerException)
    End Try
End Sub

' Khởi tạo đối tượng cho người sử dụng hiện hành
Protected Overrides Sub Activate()
    ' Cho code vào đây, ví dụ mở kết nối, tạo các file tạm...
    DoTracing("Product activated at " & DateTime.Now.ToLongTimeString)
End Sub

' Chỉ cho COM+ biết đối tượng có thể được giữ lại hay không.
Protected Overrides Function CanBePooled() As Boolean
    DoTracing("Product.CanBePooled called at " & _
                            DateTime.Now.ToLongTimeString)
    Return True
End Function

' Reset đối tượng cho người sử dụng kế tiếp
Protected Overrides Sub Deactivate()
    DoTracing("Product deactivated at " & DateTime.Now.ToLongTimeString)
End Sub

' Ghi vào Application Log
Private Sub WriteToLog(ByVal strMsg As String)
    Dim oEventLog As New EventLog("Application")
    oEventLog.Source = "CreateEnterpriseServicesComponents"
    oEventLog.WriteEntry(strMsg)
End Sub

' Báo cáo các trạng thái của đối tượng hiện hành
Private Sub DoTracing(ByVal strMsg As String)
    ' Tạo một trace listener cho event log
    ' Dim logTraceListener As New EventLogTraceListener( _
    '                                "Product-- EnterpriseComponents")

    'Trace.Listeners.Add(logTraceListener)
```

```
        ' Tạo một trace listener gửi các thông báo ra cửa sổ giao tiếp
        ' Nó có thể ghi vào một file thay vì ghi ra cửa sổ
        Dim consoleTraceListener As New _
                                TextWriterTraceListener(System.Console.Out)

        Trace.Listeners.Add(consoleTraceListener)

        Trace.WriteLine(strMsg)
    End Sub
End Class
```

Supplier.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.EnterpriseServices

' Xem mô tả các thuộc tính ở trên

<Transaction(TransactionOption.Required), _
JustInTimeActivation(True), _
ObjectPooling(Enabled:=True, MinPoolSize:=2, MaxPoolSize:=10), _
ConstructionEnabledAttribute(Default:="Server=localhost;DataBase=Northwind;Integr
  ated Security=SSPI")> _
Public Class Supplier
    Inherits ServicedComponent

    Private m_strConnectionString As String

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"
```

```
Public Sub New()
    MyBase.New()
    m_strConnectionString = SQL_CONNECTION_STRING
    DoTracing("Supplier instantiated at " & DateTime.Now.ToLongTimeString)
End Sub

Protected Overrides Sub Construct(ByVal constructString As String)
    If constructString.Length > 0 Then
        m_strConnectionString = constructString
    End If
End Sub

' Giả lập gọi SP thêm một supplier
'
<AutoCompleteAttribute(True)> _
Public Sub AddSupplier(ByVal CompanyName As String, ByVal Phone As String)
    '...
    Try
        ' cmd = New SqlCommand("AddSupplier", cnn)
        '...
        ' cmd.ExecuteNonQuery()
        DoTracing("Supplier.AddSupplier called at " & _
                                DateTime.Now.ToLongTimeString)
    Catch exp As Exception
        Throw New Exception(exp.Message, exp.InnerException)
    End Try
End Sub

' Giả lập gọi một SP để xóa một supplier
<AutoCompleteAttribute(True)> _
Public Sub DeleteSupplier(ByVal SupplierID As Integer)
    '...
    Try
        'cmd = New SqlCommand("DeleteSupplier", cnn)
        '...
        'cmd.ExecuteNonQuery()
        DoTracing("Supplier.DeleteSupplier called at " & _
                                DateTime.Now.ToLongTimeString)
```

```
    Catch exp As Exception
      Throw New Exception(exp.Message, exp.InnerException)
    End Try
  End Sub

  ' Giả lập gọi SP để lấy danh sách tất cả supplier
  <AutoCompleteAttribute(True)> _
  Public Function GetSuppliers() As DataTable
    '...
    Try
      '...
      'da.Fill(dt)
      DoTracing("Supplier.GetSuppliers called at " & _
                DateTime.Now.ToLongTimeString)
      'Return dt
    Catch exp As Exception
      Throw New Exception(exp.Message, exp.InnerException)
    End Try
  End Function

  ' Giả lập cập nhật một supplier
  <AutoCompleteAttribute(True)> _
  Public Sub UpdateSupplier(ByVal SupplierID As Integer, ByVal CompanyName As
String, ByVal Phone As String)
    '...
    Try
      'cmd = New SqlCommand("UpdateSupplier", cnn)
      '...
      'cmd.ExecuteNonQuery()
      DoTracing("Supplier.UpdateSupplier called at " & _
                DateTime.Now.ToLongTimeString)
    Catch exp As Exception
      Throw New Exception(exp.Message, exp.InnerException)
    End Try
  End Sub

  Protected Overrides Sub Activate()
    DoTracing("Supplier activated at " & DateTime.Now.ToLongTimeString)
  End Sub
```

```
    Protected Overrides Function CanBePooled() As Boolean

        DoTracing("Supplier.CanBePooled called at " & _
                        DateTime.Now.ToLongTimeString)
        Return True
    End Function

    Protected Overrides Sub Deactivate()
        DoTracing("Supplier deactivated at " & DateTime.Now.ToLongTimeString)
    End Sub

    Private Sub WriteToLog(ByVal strMsg As String)
        Dim oEventLog As New EventLog("Application")
        oEventLog.Source = "EnterpriseServicesComponents"
        oEventLog.WriteEntry(strMsg)
    End Sub

    Private Sub DoTracing(ByVal strMsg As String)
        'Dim logTraceListener As New EventLogTraceListener( _
        '                        "Supplier-- EnterpriseComponents")

        'Trace.Listeners.Add(logTraceListener)

        Dim consoleTraceListener As New _
                TextWriterTraceListener(System.Console.Out)

        Trace.Listeners.Add(consoleTraceListener)
        Trace.WriteLine(strMsg)
    End Sub
End Class
```

Project: **Create Enterprise Services Components**

frmMain.vb

```
Option Strict On
Imports CreateEnterpriseServicesComponents

Public Class frmMain
    Inherits System.Windows.Forms.Form
```

```
' Code phát sinh form được bỏ qua

 Private Sub btnAddProduct_Click(ByVal sender As System.Object, ByVal e As
 System.EventArgs) Handles btnAddProduct.Click
    Try
      Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
      Dim oProduct As New Product()
      Dim strProductName As String
      Dim intSupplierID As Integer
      Dim decUnitPrice As Decimal

      strProductName = txtProductName.Text.Trim
      intSupplierID = CInt(txtSupplierID2.Text.Trim)
      decUnitPrice = CDec(txtUnitPrice.Text.Trim)

      oProduct.AddProduct(strProductName, _
        intSupplierID, decUnitPrice)
      sbrStatus.Text = "Added product: " & strProductName
    Catch exp As Exception
      MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
    Finally
      Cursor.Current = System.Windows.Forms.Cursors.Default
    End Try
 End Sub

 Private Sub btnAddSupplier_Click(ByVal sender As System.Object, ByVal e As
 System.EventArgs) Handles btnAddSupplier.Click
    Try
      Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
      Dim oSupplier As New Supplier()
      Dim strCompanyName As String
      Dim strPhone As String

      strCompanyName = txtCompanyName.Text.Trim
      strPhone = txtPhone.Text.Trim

      oSupplier.AddSupplier(strCompanyName, strPhone)
      sbrStatus.Text = "Added supplier: " & strCompanyName
```

```
    Catch exp As Exception
      MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
    Finally
      Cursor.Current = System.Windows.Forms.Cursors.Default
    End Try
  End Sub

  Private Sub btnDeleteProduct_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeleteProduct.Click
    Try
      Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
      Dim oProduct As New Product()
      Dim intID As Integer

      intID = CInt(txtProductID.Text.Trim)
      oProduct.DeleteProduct(intID)
      sbrStatus.Text = "Deleted product: " & intID
    Catch exp As Exception
      MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
    Finally
      Cursor.Current = System.Windows.Forms.Cursors.Default
    End Try
  End Sub

  Private Sub btnDeleteSupplier_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnDeleteSupplier.Click
    Try
      Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
      Dim oSupplier As New Supplier()
      Dim intID As Integer
      intID = CInt(txtSupplierID1.Text.Trim)
      oSupplier.DeleteSupplier(intID)
      sbrStatus.Text = "Deleted supplier: " & intID
    Catch exp As Exception
      MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
    Finally
      Cursor.Current = System.Windows.Forms.Cursors.Default
    End Try
  End Sub
```

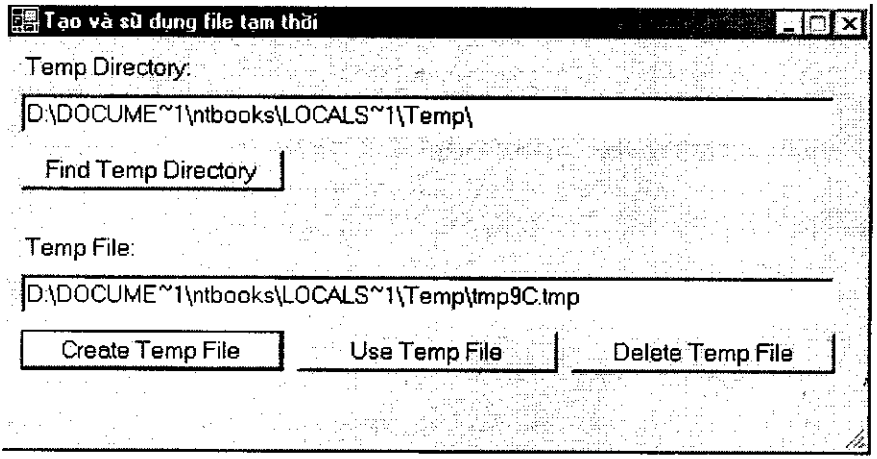
```
Private Sub btnGetProducts_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetProducts.Click
    Try
        Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
        Dim oProduct As New Product()
        oProduct.GetProducts()
        sbrStatus.Text = "Got list of products"
    Catch exp As Exception
        MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
    Finally
        Cursor.Current = System.Windows.Forms.Cursors.Default
    End Try
End Sub

Private Sub btnGetSuppliers_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetSuppliers.Click
    Try
        Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
        Dim oSupplier As New Supplier()
        oSupplier.GetSuppliers()
        sbrStatus.Text = "Got list of suppliers"
    Catch exp As Exception
        MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
    Finally
        Cursor.Current = System.Windows.Forms.Cursors.Default
    End Try
End Sub

Private Sub btnUpdateProduct_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUpdateProduct.Click
    Try
        Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
        Dim oProduct As New Product()
        Dim strProductName As String
        Dim intSupplierID As Integer
        Dim decUnitPrice As Decimal
        Dim intProductID As Integer

        strProductName = txtProductName.Text.Trim
        intSupplierID = CInt(txtSupplierID2.Text.Trim)
```

```
        decUnitPrice = CDec(txtUnitPrice.Text.Trim)
        intProductID = CInt(txtProductID.Text.Trim)

        oProduct.UpdateProduct(intProductID, strProductName, _
          intSupplierID, decUnitPrice)
        sbrStatus.Text = "Updated product: " & intProductID
      Catch exp As Exception
        MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
      Finally
        Cursor.Current = System.Windows.Forms.Cursors.Default
      End Try
    End Sub

    Private Sub btnUpdateSupplier_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles btnUpdateSupplier.Click
      Try
        Cursor.Current = System.Windows.Forms.Cursors.WaitCursor
        Dim oSupplier As New Supplier()
        Dim strCompanyName As String
        Dim strPhone As String
        Dim intID As Integer
        strCompanyName = txtCompanyName.Text.Trim
        strPhone = txtPhone.Text.Trim
        intID = CInt(txtSupplierID1.Text.Trim)
        oSupplier.UpdateSupplier(intID, strCompanyName, strPhone)
        sbrStatus.Text = "Updated supplier: " & strCompanyName
      Catch exp As Exception
        MsgBox(exp.ToString(), MsgBoxStyle.OKOnly, Me.Text)
      Finally
        Cursor.Current = System.Windows.Forms.Cursors.Default
      End Try
    End Sub

    Private Sub textBoxes_TextChanged(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles txtCompanyName.TextChanged,
    txtPhone.TextChanged, txtSupplierID1.TextChanged,
    txtProductName.TextChanged, txtSupplierID2.TextChanged,
    txtUnitPrice.TextChanged, txtProductID.TextChanged
      btnAddSupplier.Enabled = _
        txtCompanyName.Text.Trim.Length <> 0 AndAlso _
```

```
        txtPhone.Text.Trim.Length <> 0
    btnUpdateSupplier.Enabled = btnAddSupplier.Enabled AndAlso _
        txtSupplierID1.Text.Trim.Length <> 0
    btnDeleteSupplier.Enabled = txtSupplierID1.Text.Trim.Length <> 0

    btnAddProduct.Enabled = _
        txtProductName.Text.Trim.Length <> 0 AndAlso _
        txtSupplierID2.Text.Trim.Length <> 0 AndAlso _
        txtUnitPrice.Text.Trim.Length <> 0
    btnUpdateProduct.Enabled = btnAddProduct.Enabled AndAlso _
        txtProductID.Text.Trim.Length <> 0
    btnDeleteProduct.Enabled = txtProductID.Text.Trim.Length <> 0
  End Sub
End Class
```

63. Tạo và sử dụng File tạm thời

Chương trình này chỉ cách tạo và sử dụng các file tạm thời, cũng như xác định thư mục TEMP trên máy tính của người sử dụng. Việc tạo và sử dụng các resource này trong .NET Framework rất đơn giản.



Các chức năng chính:

- Tìm thư mục TEMP trên máy tính của người sử dụng.
- Tạo một file tạm thời sử dụng phương thức Path.GetTempFileName().
- Quy định thuộc tính của file tạm là "Temporary" để cải thiện hiệu xuất của chương trình.
- Xóa file tạm.

Project: **Tao va su dung file tam thoi**

frmMain.vb

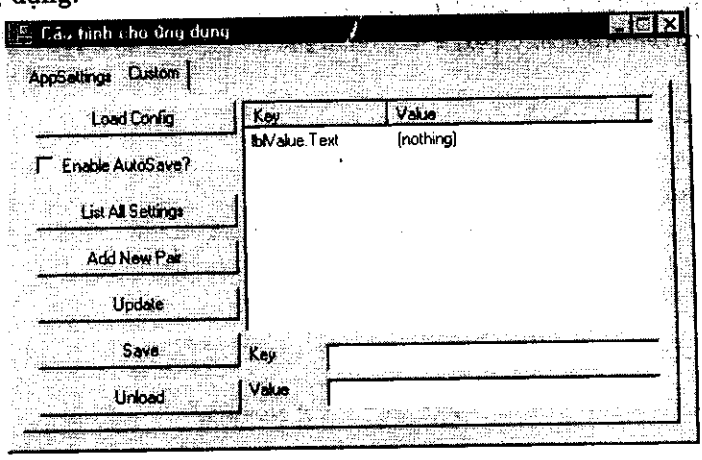
```
Option Strict On
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Tạo biến lưu tên file
    Dim m_FileName As String = ""

' Code phát sinh form được bỏ qua

    ' Thủ tục này tạo một file tạm thời và gán thuộc tính Attribute của nó là Temporary
    Private Sub btnCreateTempFile_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCreateTempFile.Click

        ' Xóa nội dung trên thanh trạng thái
        Me.sbrStatus.Text = ""

        Try
            ' Trả về đường dẫn và tên của file mới tạo. Chú ý là phương thức
            ' GetTempFileName() thực sự tạo một file 0-byte và trả về tên của file được tạo.
            m_FileName = Path.GetTempFileName()

            ' Tạo một đối tượng FileInfo để quản lý các thuộc tính của file tạm
            Dim myFileInfo As New FileInfo(m_FileName)

            myFileInfo.Attributes = FileAttributes.Temporary

        Catch exc As Exception
            ' Thông báo nếu có lỗi xảy ra
            Me.sbrStatus.Text = "Unable to create a TEMP file or set its attributes."
        End Try
        ' Hiển thị tên của file tạm vừa tạo
        Me.txtTempFile.Text = m_FileName
    End Sub
```

```
' Xóa file tạm vừa tạo
Private Sub btnDeleteTempFile_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeleteTempFile.Click

    If m_FileName = "" Then
        Me.sbrStatus.Text = "Temp file not yet created."
        Return
    Else
        Me.sbrStatus.Text = ""
    End If

    ' Thực hiện xóa file
    Try
        File.Delete(m_FileName)
        m_FileName = ""
        Me.txtTempFile.Text = ""
    Catch exc As Exception
        ' Hiển thị lỗi nếu có
        Me.sbrStatus.Text = "Error deleteing Temp file."
    End Try
End Sub


' Tìm đường dẫn của thư mục TEMP
Private Sub btnFindTempDirectory_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnFindTempDirectory.Click

    Dim tempPathString As String = ""

    Me.sbrStatus.Text = ""

    ' Thực hiện lấy đường dẫn, đặt trong Try Cath để bắt lỗi khi người sử
    ' dụng không có quyền hạn trên thư mục này.
    Try
        tempPathString = Path.GetTempPath()

    Catch sex As Security.SecurityException
        ' Hiển thị lỗi khi không có quyền hạn
        Me.sbrStatus.Text = "You do not have the required permissions."
```

```
        Catch exc As Exception
            Me.sbrStatus.Text = "Unable to retrieve TEMP directory path."
        End Try
        ' hiển thị đường dẫn đến thư mục TEMP
        Me.txtTempDirectory.Text = tempPathString

    End Sub

    ' Thủ tục này sử dụng file tạm vừa tạo.
    Private Sub btnUseTempFilebtnUseTempFile_Click(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    btnUseTempFilebtnUseTempFile.Click

        If m_FileName = "" Then
            Me.sbrStatus.Text = "Temp file not yet created."
            Return
        Else
            Me.sbrStatus.Text = ""
        End If

        ' Ghi vào file một số văn bản và đọc lại nó
        Try
            ' Ghi vào file tạm
            Dim myWriter As StreamWriter = File.AppendText(m_FileName)
            myWriter.WriteLine("Data written to temporary file.")
            myWriter.Flush()
            myWriter.Close()

            ' Đọc nội dung file tạm
            Dim myReader As StreamReader = File.OpenText(m_FileName)
            ' Hiển thị nội dung vừa đọc
            Me.sbrStatus.Text = "Temp file: " + myReader.ReadToEnd()
            myReader.Close()

        Catch exc As Exception
            Me.sbrStatus.Text = "Error writing to or reading from Temp file."
        End Try
    End Sub
End Class
```

64. Cấu hình cho ứng dụng

Chương trình này chỉ cách đọc và ghi vào phần **appSettings** trong file cấu hình của ứng dụng.



Các chức năng chính:

- Để cấu hình thông số cho ứng dụng ta chỉ cần thêm file **app.config** vào Solution bằng cách chọn **Project | Add New Item** và chọn phần **Application Configuration File**. File này được thêm vào thư mục bin khi biên dịch. Sau đó ta có thể thêm các thông số riêng để người sử dụng hay các nhà quản trị có thể thay đổi các thông số của ứng dụng bằng trình soạn thảo file văn bản.
- Ta có thể cấu hình cho từng control cụ thể thông qua file cấu hình. Chỉ cần thêm thuộc tính **Dynamic Property** cho control nào ta muốn. Các giá trị Dynamic Property được định nghĩa trong phần **appSetting** của file cấu hình sẽ được load khi chương trình chạy. Ta có thể truy xuất các giá trị này thông qua **ConfigurationSettings.AppSettings**.
- Để thay đổi các giá trị này trong lúc chương trình đang chạy, ta cần sử dụng các class trong namespace **System.Xml**.

Project: **Cau hinh cho ung dung**

app.config

```
<?xml version="1.0" encoding="Windows-1252" ?>
<configuration>
  <appSettings>
    <!-- Các thuộc tính cấu hình cho ứng dụng đặt ở đây -->
```

```
        <!--  Ví dụ: <add key="settingName" value="settingValue"/> -->
        <add key="lblValue.Text" value="(nothing)" />
    </appSettings>
</configuration>
```

frmMain.vb

```
Option Strict On

Imports System.Configuration
Imports System.Collections
Imports System.Xml

Public Class frmMain
    Inherits System.Windows.Forms.Form
    ' Biến lưu tên file cấu hình
    Private mstrCFGFile As String
    ' Chú ý là ConfigurationSettings.AppSettings chỉ cho phép đọc các giá trị và
    ' Specialized.NameValueCollection cho phép đọc và thay đổi giá trị.
    Dim mAppSet As Specialized.NameValueCollection

    ' Tạo một class để làm việc với các thông số của ứng dụng
    Private mcustAppSettings As AppSettings
    ' Cụ thể cho một cặp thông số
    Private mcustAppSet As AppSetting

' Code phát sinh form được bỏ qua

    Private Sub cmdLoadAS_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles cmdLoadAS.Click
        ' Truy xuất các thông số đã được lấy khi chương trình bắt đầu chạy
        mAppSet = ConfigurationSettings.AppSettings

        MessageBox.Show("Applicaton Settings have been loaded.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
        Me.cmdListByIndex.Enabled = True
        Me.cmdListByKey.Enabled = True
        Me.cmdReadValue.Enabled = True
        Me.cmdLoadAS.Enabled = False
    End Sub
```

```
Private Sub cmdListByKey_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdListByKey.Click
    ' Thủ tục này liệt kê các phần tử bằng cách lấy danh sách các khóa (key) và
    ' truy xuất đến các phần tử thông qua khóa của nó
  If Not mAppSet Is Nothing Then
    Me.lstSettings.Items.Clear()
    Dim keys() As String
    keys = mAppSet.AllKeys

    Dim key As String
    For Each key In keys
      Me.lstSettings.Items.Add(key & ": " & mAppSet.Item(key))
    Next
  End If
End Sub

Private Sub cmdListByIndex_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdListByIndex.Click
    ' Thủ tục này liệt kê các phần tử bằng cách sử dụng chỉ số. Bắt đầu từ 0
  If Not mAppSet Is Nothing Then
    Me.lstSettings.Items.Clear()
    Dim i As Integer
    For i = 0 To mAppSet.Count - 1
      Me.lstSettings.Items.Add(mAppSet.GetKey(i) & ": " & mAppSet.Item(i))
    Next
  End If
End Sub

Private Sub cmdReadValue_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdReadValue.Click
  Me.lblValue.Text = mAppSet.Item(Me.txtKey.Text)
End Sub

Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    ' Lấy tên của file cấu hình
  Dim ad As AppDomain = AppDomain.CurrentDomain
  Me.mstrCFGFile = ad.SetupInformation.ConfigurationFile
End Sub
```

```
Private Sub cmdLoadCfg_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdLoadCfg.Click
    ' thủ tục này sử dụng một instance của class AppSettings để lấy các thông
    ' số trong file cấu hình.
  Try
    Dim blnAutoSave As Boolean
    If Me.chkAutoSave.CheckState = CheckState.Checked Then
      blnAutoSave = True
    Else
      blnAutoSave = False
    End If

    If Not mcustAppSettings Is Nothing Then
      mcustAppSettings.Dispose()
      mcustAppSettings = Nothing
    End If

    mcustAppSettings = New AppSettings(Me.mstrCFGFile, blnAutoSave)

    If Me.lvSettings.Items.Count > 0 Then
      Me.lvSettings.Items.Clear()
    End If

    Me.cmdAddNew.Enabled = True
    Me.cmdListAllSettings.Enabled = True
    Me.cmdUpdate.Enabled = True
    Me.cmdSave.Enabled = True
    Me.cmdUnload.Enabled = True

  Catch exp As Exception
    MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                    MessageBoxIcon.Error)
  End Try
End Sub

Private Sub cmdListAllSettings_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdListAllSettings.Click
  Me.ListCustomSettings()
End Sub
```

```
Private Sub lvSettings_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles lvSettings.SelectedIndexChanged
    ' Đưa cặp giá trị key/value được chọn vào các TextBox tương ứng
    Dim strKey As String
    Dim strValue As String
    If Me.lvSettings.SelectedItems.Count = 1 Then
      Me.txtSettingKey.Text = Me.lvSettings.SelectedItems(0).SubItems(0).Text
      Me.txtSettingValue.Text = Me.lvSettings.SelectedItems(0).SubItems(1).Text
    End If
End Sub

Private Sub cmdAddNew_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdAddNew.Click
    ' Thêm một phần tử mới vào Collection
    Try
      If Me.txtSettingKey.Text = String.Empty Then
        MessageBox.Show("You must enter a value for the key.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Me.txtSettingKey.Select()
        Exit Sub
      End If
      If Me.txtSettingValue.Text = String.Empty Then
        MessageBox.Show("You must enter a value for the value field.", _
              Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Me.txtSettingValue.Select()
        Exit Sub
      End If
      mcustAppSet = New AppSetting(Me.txtSettingKey.Text, _
                                   Me.txtSettingValue.Text)
      mcustAppSettings.Add(mcustAppSet)
      Me.ListCustomSettings()
      MessageBox.Show("New setting added.", Me.Text, _
                      MessageBoxButtons.OK, MessageBoxIcon.Information)
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, _
                      MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
```

```
Private Sub ListCustomSettings()
    ' Liệt kê tất cả các thông số trong ListView
  If Not mcustAppSettings Is Nothing Then
    Try
      Me.lvSettings.Items.Clear()
      For Each mcustAppSet In mcustAppSettings
        With Me.lvSettings.Items.Add(mcustAppSet.Key)
          .SubItems.Add(mcustAppSet.Value)
        End With
      Next
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, _
                      MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
  End If
End Sub

Private Sub frmMain_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
  Me.UnloadConfigSettings(True)
End Sub

Private Sub cmdUpdate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUpdate.Click
    ' Cập nhật phần tử đã tồn tại
  Try
    If Me.txtSettingKey.Text = String.Empty Then
      MessageBox.Show("You must enter a value for the key.", Me.Text, _
                      MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
      Me.txtSettingKey.Select()
      Exit Sub
    End If
    If Me.txtSettingValue.Text = String.Empty Then
      MessageBox.Show("You must enter a value for the value field.", _
            Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
      Me.txtSettingValue.Select()
      Exit Sub
    End If
```

```
        mcustAppSet = New AppSetting(Me.txtSettingKey.Text, _
                                     Me.txtSettingValue.Text)
        mcustAppSettings.Update(mcustAppSet)

        Me.ListCustomSettings()

        MessageBox.Show("Existing setting updated.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)

    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
    End Try
End Sub

Private Sub cmdSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdSave.Click
    ' Lưu lại các thay đổi
    If Not mcustAppSettings Is Nothing Then
        Try
            mcustAppSettings.Save()
            MessageBox.Show("Configuration file saved..", Me.Text, _
                            MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch exp As Exception
            MessageBox.Show(exp.Message, exp.Source, _
                            MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End If
End Sub

Private Sub cmdUnload_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdUnload.Click

    Me.UnloadConfigSettings(False)

End Sub
```

```
    Private Sub UnloadConfigSettings(ByVal FormClosing As Boolean)
        ' Unload file cấu hình bằng cách gọi phương thức Dispose và
        ' giải phóng tham chiếu đến đối tượng
        Try
            If Not mcustAppSettings Is Nothing Then
                mcustAppSettings.Dispose()
                mcustAppSettings = Nothing

                If Not FormClosing Then
                    Me.lvSettings.Items.Clear()

                    Me.txtSettingKey.Text = String.Empty
                    Me.txtSettingValue.Text = String.Empty

                    Me.cmdListAllSettings.Enabled = False
                    Me.cmdAddNew.Enabled = False
                    Me.cmdUpdate.Enabled = False
                    Me.cmdSave.Enabled = False
                    Me.cmdLoadCfg.Select()
                    Me.cmdUnload.Enabled = False
                End If
            End If

        Catch exp As Exception
            MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                            MessageBoxIcon.Error)
        End Try
    End Sub

End Class

Public Class AppSetting
    ' Class này ánh xạ các cặp name/value

    Private mParent As AppSettings
    Private mstrKey As String
    Private mstrValue As String
```

```
Public Property Key() As String
  Get
    Return mstrKey
  End Get

  Set(ByVal Value As String)
    Me.UpdateParent()
    Value = mstrKey
  End Set
End Property

Public Property Value() As String
  Get
    Return mstrValue
  End Get
  Set(ByVal Value As String)
    Me.UpdateParent()
    mstrValue = Value
  End Set
End Property

Private Sub UpdateParent()
  If Not Me.mParent Is Nothing Then
    Me.mParent.Update(Me)
  End If
End Sub

Public Sub New()
  Me.New(String.Empty, String.Empty)
End Sub

Public Sub New(ByVal Key As String, ByVal Value As String)
  Me.New(Key, Value, Nothing)
End Sub

Friend Sub New(ByVal Key As String, ByVal Value As String, ByVal Parent As
AppSettings)
  MyBase.New()
  Me.mstrKey = Key
  Me.mstrValue = Value
```

```
        Me.mParent = Parent
    End Sub
End Class

Public Class AppSettings
    Implements IEnumerable, IDisposable

    ' Class này xử lý phần configuration//appSettings của file cấu hình.
    ' XPath được sử dụng để tìm các giá trị khi được yêu cầu

    Private cfg As New XmlDocument()
    Private xAS As XmlNode
    Private mstrFileName As String
    Private mblnAutoSave As Boolean
    Private mblnDirty As Boolean
    Private maItems() As AppSetting
    Private mblnDisposing As Boolean
    Private mblnDisposed As Boolean

    Private Const APPSETTINGS_ELEMENT As String = "configuration//appSettings"
    Private Const NEWELEMENT As String = "add"
    Private Const XPATH_KEY_ADD As String = "//add"
    Private Const XPATH_KEY_ADD_KEY As String = "//add[@key='{0}']"

    Public Sub New(ByVal ConfigFile As String)
        Me.New(ConfigFile, False)
    End Sub

    Public Sub New(ByVal ConfigFile As String, ByVal AutoSave As Boolean)

        MyBase.New()

        If ConfigFile.Length = 0 Then
            Throw New ArgumentNullException("You must pass in a valid file name.")
        Else
            If System.IO.File.Exists(ConfigFile) Then
                Try

                    cfg.Load(ConfigFile)
```

```
        Catch exp As Exception
          Throw New System.IO.FileLoadException("The file name specified" & _
                  " could not be loaded. Please see InnerException" & _
                  " for more information", exp)
        End Try

          ' Lấy phần appSettings chính để ta có thể thêm các thông số mới
        xAS = cfg.SelectSingleNode(APPSETTINGS_ELEMENT)

        If xAS Is Nothing Then
          Throw New ConfigurationException("The file specified, while a valid" & _
                          " XML document, is not a valid configuration file.")
        End If

          ' Đến đây ta cần lưu lại tên file
        mstrFileName = ConfigFile

        Me.AutoSave = AutoSave
      Else
        Throw New System.IO.FileNotFoundException(String.Format( _
                  "The file name specified {0} does not exist.", ConfigFile))
      End If
    End If
  End Sub

  Public Sub Dispose() Implements System.IDisposable.Dispose
    Me.mbInDisposing = True
    If Me.Dirty Then
      Me.Save()
    End If
    Me.mbInDisposed = True
    Me.mbInDisposing = False
    GC.SuppressFinalize(Me)
  End Sub

  Protected Overrides Sub Finalize()
    MyBase.Finalize()
    If Me.Dirty Then
      If Not Me.mbInDisposed Then
        If Not Me.mbInDisposing Then
```

```
            Me.Dispose()
          End If
        End If
      End If
    End Sub

    Public Property AutoSave() As Boolean
      Get
        Return Me.mblnAutoSave
      End Get
      Set(ByVal Value As Boolean)
        Me.mblnAutoSave = Value
      End Set
    End Property

    Public ReadOnly Property Dirty() As Boolean
      Get
        Return Me.mblnDirty
      End Get
    End Property

    Public Sub Add(ByVal NewSetting As AppSetting)
      Dim newElem As XmlElement
      Dim newAttr As XmlAttribute

      newElem = cfg.CreateElement(NEWELEMENT)
      newAttr = cfg.CreateAttribute("key")
      newAttr.Value = NewSetting.Key
      newElem.Attributes.Append(newAttr)
      newAttr = cfg.CreateAttribute("value")
      newAttr.Value = NewSetting.Value
      newElem.Attributes.Append(newAttr)
      xAS.AppendChild(newElem)
      Me.mblnDirty = True
      If Me.AutoSave Then
        Me.Save()
      End If
    End Sub
```

```
    Public Function Add(ByVal Key As String, ByVal Value As String) As AppSetting
        Dim NewSetting As New AppSetting(Key, Value, Me)
        Me.Add(NewSetting)
        Return NewSetting
    End Function

    Public Function Item(ByVal Key As String) As AppSetting
        Dim xNode As XmlNode
        Dim strSearch As String = XPATH_KEY_ADD_KEY

        xNode = xAS.SelectSingleNode(String.Format(strSearch, Key))

        If xNode Is Nothing Then
            Throw New ArgumentOutOfRangeException("Key", Key, _
                                "The item you wanted to update does not exists.")
        Else
            Dim las As New AppSetting()
            las.Key = Key
            las.Value = xNode.Attributes.Item(1).Value

            Return las
        End If
    End Function

    Public Sub RemoveByKey(ByVal Setting As AppSetting)
        Dim xNode As XmlNode
        Dim strSearch As String = XPATH_KEY_ADD_KEY

        xNode = xAS.SelectSingleNode(String.Format(strSearch, Setting.Key))

        If Not xNode Is Nothing Then

        End If
        Me.mblnDirty = True
        If Me.AutoSave Then
            Me.Save()
        End If

    End Sub
```

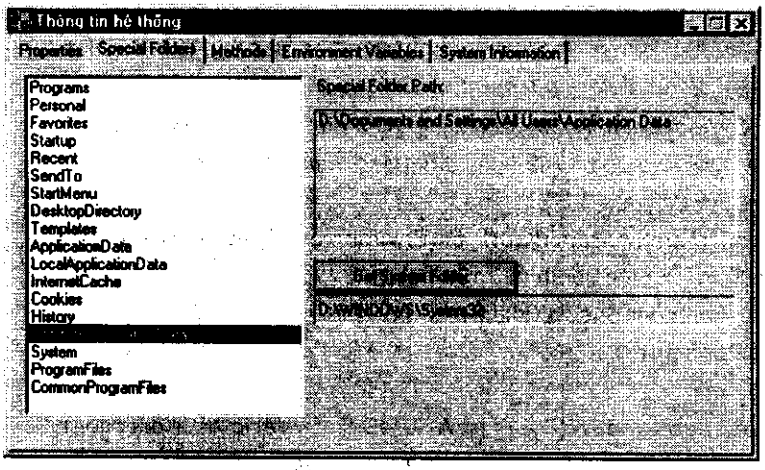
```
Public Sub RemoveByKey(ByVal Key As String)

End Sub

Public Function Update(ByVal Key As String, ByVal Value As String) As AppSetting
    Dim NewSetting As New AppSetting(Key, Value, Me)
    Me.Update(NewSetting)
    Return NewSetting
End Function

Public Sub Update(ByVal NewSetting As AppSetting)
    Dim xNode As XmlNode
    Dim strSearch As String = XPATH_KEY_ADD_KEY

    xNode = xAS.SelectSingleNode(String.Format(strSearch, NewSetting.Key))

    If xNode Is Nothing Then
        Throw New ArgumentOutOfRangeException("Key", NewSetting.Key, _
                                    "The item you wanted to update does not exists.")
    Else
        xNode.Attributes.Item(1).Value = NewSetting.Value
    End If

    Me.mbInDirty = True

    If Me.AutoSave Then
        Me.Save()
    End If
End Sub

Public Sub Save()
    ' Ta không cần Try Catch ở đây, nếu có lỗi sẽ được bắt ở hàm gọi
    cfg.Save(Me.mstrFileName)
    Me.mbInDirty = False
End Sub

Private Function GetAllItems() As AppSetting()
    Dim xNode As XmlNode
    Dim xNodeList As XmlNodeList
    Dim atts As XmlAttributeCollection
```

```
    xNodeList = xAS.SelectNodes(XPATH_KEY_ADD)

    Dim xa As XmlAttribute
    Dim asa(xNodeList.Count - 1) As AppSetting
    Dim asi As AppSetting
    Dim i As Integer = -1

    For Each xNode In xNodeList
      i += 1
      atts = xNode.Attributes
      With atts
        asi = New AppSetting(.Item(0).Value, .Item(1).Value, Me)
      End With
      asa(i) = asi
    Next
    Return asa
  End Function

  Private Function GetEnumerator() As System.Collections.IEnumerator Implements
System.Collections.IEnumerable.GetEnumerator
    Me.maItems = Me.GetAllItems()
    Return New Iterator(Me.maItems)
  End Function

  Private Class Iterator
    Implements IEnumerator

    ' Class private này cung cấp các hàm cần thiết để For...Each làm việc được
    Private mData() As AppSetting
    Private Index As Integer = -1
    Public Sub New(ByVal Keys() As AppSetting)
      mData = Keys
    End Sub

    Private ReadOnly Property Current() As Object Implements
System.Collections.IEnumerator.Current
      Get
        Return mData(Index)
      End Get
    End Property
```

```
    Private Function MoveNext() As Boolean Implements _
                        System.Collections.IEnumerator.MoveNext
      Index += 1
      If (Index <= (mData.Length - 1)) Then
        Return True
      End If
    End Function

    Private Sub Reset() Implements System.Collections.IEnumerator.Reset
      Index = -1
    End Sub
  End Class
End Class
```

65. Lấy thông tin hệ thống và môi trường.

Chương trình này biểu diễn cách sử dụng các class cung cấp thông tin môi trường (environment), có trong namespace System.Environment.



Các chức năng chính:

- Chương trình này chỉ cách sử dụng các phương thức và thuộc tính của các đối tượng trong namespace System.Environment. Tab Properties cho biết các thuộc tính của System.Environment.
- Tab Methods cho thấy kết quả chạy các phương thức của namespace này.
- Tab Special Folders cho phép lấy đường dẫn của bất kỳ thư mục đặc biệt nào của Windows.
- Tab Environment Variables hoạt động tương tự.

- Cuối cùng tab SystemInfomation lấy tất cả thông tin cung cấp bởi đối tượng SystemInformation.

Project: **Thong tin he thong**

frmMain.vb

```
Option Strict On
Imports System.Environment

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

#Region " Enum Handling "
    Private Sub LoadList(ByVal lst As ListBox, ByVal typ As Type)
        lst.DataSource = System.Enum.GetNames(typ)
    End Sub

    Private Function GetSpecialFolderFromList() As Environment.SpecialFolder
        ' lstFolders.SelectedItem trả về tên của một thư mục đặc biệt.
        ' System.Enum.Parse sẽ chuyển nó thành một đối tượng tương ứng.
        ' Ctype sau đó chuyển kiểu đối tượng thành đối tượng
        Return CType( _
        System.Enum.Parse(GetType(Environment.SpecialFolder), _
        lstFolders.SelectedItem.ToString), _
        Environment.SpecialFolder)
    End Function

    Private Sub LoadList(ByVal lst As ListBox, ByVal ic As ICollection)
        ' Dùng cho ListBox hiển thị nội dung của đối tượng có
        ' sử dụng Interface ICollection. Trước tiên phải copy nội dung vào đối tượng có
        ' sử dụng IList (ví dụ: một mảng) và sau đó kết vào đối tượng này.
        Dim astrItems(ic.Count - 1) As String
        ic.CopyTo(astrItems, 0)
        lst.DataSource = astrItems
    End Sub

#End Region
```

```
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExit.Click
    Environment.Exit(CInt(nudExitCode.Value))
End Sub

Private Sub btnExpand_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnExpand.Click
    lblExpandResults.Text = _
                Environment.ExpandEnvironmentVariables(txtExpand.Text)
End Sub

Private Sub btnGetEnvironmentVariable_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnGetEnvironmentVariable.Click
    lblTEMP.Text = Environment.GetEnvironmentVariable("TEMP")
End Sub

Private Sub btnGetSystemFolder_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnGetSystemFolder.Click
    ' Nếu chỉ muốn lấy một đường dẫn của một thư mục đặc biệt, truyền vào
    ' một trong các giá trị của enumeration SpecialFolder. GetFolderPath chính xác
    ' là System.Environment.GetFolderPath

    lblSystemFolder.Text = GetFolderPath(SpecialFolder.System)
End Sub

Private Sub btnRefreshTickCount_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnRefreshTickCount.Click

    lblTickCount.Text = Environment.TickCount.ToString
End Sub

Private Sub btnStackTrace_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStackTrace.Click

    Try
        MsgBox(Environment.StackTrace, MsgBoxStyle.OKOnly, Me.Text)

    Catch exp As System.Security.SecurityException
        MsgBox("A security exception occurred." & vbCrLf & exp.Message, _
                                        MsgBoxStyle.Critical, exp.Source)
```

```
    Catch exp As System.Exception
      MsgBox("An unexpected error occurred accessing the StackTrace." & _
                    vbCrLf & exp.Message, MsgBoxStyle.Critical, exp.Source)
    End Try
  End Sub

  Private Sub frmEnvironment_Load(ByVal sender As Object, ByVal e As
  System.EventArgs) Handles MyBase.Load
    LoadList(lstFolders, GetType(Environment.SpecialFolder))
    LoadList(lstEnvironmentVariables, GetEnvironmentVariables.Keys)

    ' Hiển thị các thuộc tính trên tab Properties
    LoadProperties()

    ' Sử dụng các phương thức đơn giản
    RunMethods()
    ' Lấy các giá trị từ class SystemInformation
    LoadSystemInformation()
  End Sub

  Private Sub lstEnvironmentVariables_SelectedIndexChanged(ByVal sender As
  System.Object, ByVal e As System.EventArgs) Handles
  lstEnvironmentVariables.SelectedIndexChanged
    ' lstEnvironmentVariables chứa danh sách các biến môi trường.
    ' GetEnvironmentVariable trả về giá trị tương ứng với biến được chọn
    lblEnvironmentVariable.Text = _
    GetEnvironmentVariable(lstEnvironmentVariables.SelectedItem.ToString)
  End Sub

  Private Sub lstFolders_SelectedIndexChanged(ByVal sender As System.Object,
  ByVal e As System.EventArgs) Handles lstFolders.SelectedIndexChanged
    Dim sf As Environment.SpecialFolder
    ' GetSpecialFolderFromList trả về một thành phần
    ' của enum Environment.SpecialFolder
    sf = GetSpecialFolderFromList()
    lblSpecialFolder.Text = GetFolderPath(sf)
  End Sub
```

```
Private Sub lvwSystemInformation_Resize(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles lvwSystemInformation.Resize
    lvwSystemInformation.Columns(1).Width = _
    lvwSystemInformation.ClientRectangle.Width -
lvwSystemInformation.Columns(0).Width
End Sub

Private Sub LoadProperties()
    ' Thủ tục này gán giá trị cho các phần tử trên tab Properties.
    lblWorkingSet.Text = Environment.WorkingSet.ToString
    lblVersion.Text = Environment.Version.ToString
    lblUserName.Text = Environment.UserName
    lblUserDomainName.Text = Environment.UserDomainName
    lblTickCount.Text = Environment.TickCount.ToString
    lblSystemDirectory.Text = Environment.SystemDirectory
    lblOSVersion.Text = Environment.OSVersion.ToString
    lblMachineName.Text = Environment.MachineName
    lblCurrentDirectory.Text = Environment.CurrentDirectory
    lblCommandLine.Text = Environment.CommandLine
End Sub

Private Sub LoadSystemInformation()
    ' Cho thông tin về các thuộc tính Shared của class SystemInfomation vào
    ' control ListView. Chuỗi của phần tử là tên của thuộc tính cung cấp bởi class
    ' SystemInformation
    With lvwSystemInformation.Items.Add("ArrangeDirection")
        .SubItems.Add(SystemInformation.ArrangeDirection.ToString())
    End With
    With lvwSystemInformation.Items.Add("ArrangeStartingPosition")
        .SubItems.Add(SystemInformation.ArrangeStartingPosition.ToString())
    End With
    With lvwSystemInformation.Items.Add("BootMode")
        .SubItems.Add(SystemInformation.BootMode.ToString())
    End With
    With lvwSystemInformation.Items.Add("Border3DSize")
        .SubItems.Add(SystemInformation.Border3DSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("BorderSize")
        .SubItems.Add(SystemInformation.BorderSize.ToString())
    End With
```

```
With lvwSystemInformation.Items.Add("CaptionButtonSize")
    .SubItems.Add(SystemInformation.CaptionButtonSize.ToString())
End With
With lvwSystemInformation.Items.Add("CaptionHeight")
    .SubItems.Add(SystemInformation.CaptionHeight.ToString())
End With
With lvwSystemInformation.Items.Add("ComputerName")
    .SubItems.Add(SystemInformation.ComputerName.ToString())
End With
With lvwSystemInformation.Items.Add("CursorSize")
    .SubItems.Add(SystemInformation.CursorSize.ToString())
End With
With lvwSystemInformation.Items.Add("DbcsEnabled")
    .SubItems.Add(SystemInformation.DbcsEnabled.ToString())
End With
With lvwSystemInformation.Items.Add("DebugOS")
    .SubItems.Add(SystemInformation.DebugOS.ToString())
End With
With lvwSystemInformation.Items.Add("DoubleClickSize")
    .SubItems.Add(SystemInformation.DoubleClickSize.ToString())
End With
With lvwSystemInformation.Items.Add("DoubleClickTime")
    .SubItems.Add(SystemInformation.DoubleClickTime.ToString())
End With
With lvwSystemInformation.Items.Add("DragFullWindows")
    .SubItems.Add(SystemInformation.DragFullWindows.ToString())
End With
With lvwSystemInformation.Items.Add("DragSize")
    .SubItems.Add(SystemInformation.DragSize.ToString())
End With
With lvwSystemInformation.Items.Add("FixedFrameBorderSize")
    .SubItems.Add(SystemInformation.FixedFrameBorderSize.ToString())
End With
With lvwSystemInformation.Items.Add("FrameBorderSize")
    .SubItems.Add(SystemInformation.FrameBorderSize.ToString())
End With
With lvwSystemInformation.Items.Add("HighContrast")
    .SubItems.Add(SystemInformation.HighContrast.ToString())
End With
```

```
With lvwSystemInformation.Items.Add("HorizontalScrollBarArrowWidth")
    .SubItems.Add(SystemInformation.HorizontalScrollBarArrowWidth.ToString())
End With
With lvwSystemInformation.Items.Add("HorizontalScrollBarHeight")
    .SubItems.Add(SystemInformation.HorizontalScrollBarHeight.ToString())
End With
With lvwSystemInformation.Items.Add("HorizontalScrollBarThumbWidth")
    .SubItems.Add(SystemInformation.HorizontalScrollBarThumbWidth.ToString())
End With
With lvwSystemInformation.Items.Add("IconSize")
    .SubItems.Add(SystemInformation.IconSize.ToString())
End With
With lvwSystemInformation.Items.Add("IconSpacingSize")
    .SubItems.Add(SystemInformation.IconSpacingSize.ToString())
End With
With lvwSystemInformation.Items.Add("KanjiWindowHeight")
    .SubItems.Add(SystemInformation.KanjiWindowHeight.ToString())
End With
With lvwSystemInformation.Items.Add("MaxWindowTrackSize")
    .SubItems.Add(SystemInformation.MaxWindowTrackSize.ToString())
End With
With lvwSystemInformation.Items.Add("MenuButtonSize")
    .SubItems.Add(SystemInformation.MenuButtonSize.ToString())
End With
With lvwSystemInformation.Items.Add("MenuCheckSize")
    .SubItems.Add(SystemInformation.MenuCheckSize.ToString())
End With
With lvwSystemInformation.Items.Add("MenuFont")
    .SubItems.Add(SystemInformation.MenuFont.ToString())
End With
With lvwSystemInformation.Items.Add("MenuHeight")
    .SubItems.Add(SystemInformation.MenuHeight.ToString())
End With
With lvwSystemInformation.Items.Add("MidEastEnabled")
    .SubItems.Add(SystemInformation.MidEastEnabled.ToString())
End With
With lvwSystemInformation.Items.Add("MinimizedWindowSize")
    .SubItems.Add(SystemInformation.MinimizedWindowSize.ToString())
End With
```

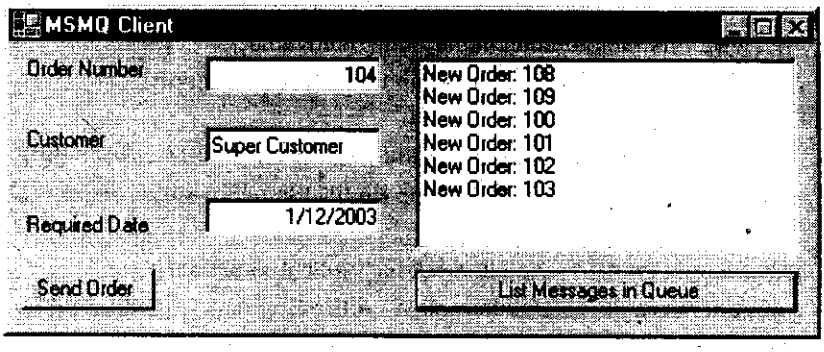
```
    With lvwSystemInformation.Items.Add("MinimizedWindowSpacingSize")
      .SubItems.Add(SystemInformation.MinimizedWindowSpacingSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("MinimumWindowSize")
      .SubItems.Add(SystemInformation.MinimumWindowSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("MinWindowTrackSize")
      .SubItems.Add(SystemInformation.MinWindowTrackSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("MonitorCount")
      .SubItems.Add(SystemInformation.MonitorCount.ToString())
    End With
    With lvwSystemInformation.Items.Add("MonitorsSameDisplayFormat")
      .SubItems.Add(SystemInformation.MonitorsSameDisplayFormat.ToString())
    End With
    With lvwSystemInformation.Items.Add("MouseButtons")
      .SubItems.Add(SystemInformation.MouseButtons.ToString())
    End With
    With lvwSystemInformation.Items.Add("MouseButtonsSwapped")
      .SubItems.Add(SystemInformation.MouseButtonsSwapped.ToString())
    End With
    With lvwSystemInformation.Items.Add("MousePresent")
      .SubItems.Add(SystemInformation.MousePresent.ToString())
    End With
    With lvwSystemInformation.Items.Add("MouseWheelPresent")
      .SubItems.Add(SystemInformation.MouseWheelPresent.ToString())
    End With
    With lvwSystemInformation.Items.Add("MouseWheelScrollLines")
      .SubItems.Add(SystemInformation.MouseWheelScrollLines.ToString())
    End With
    With lvwSystemInformation.Items.Add("NativeMouseWheelSupport")
      .SubItems.Add(SystemInformation.NativeMouseWheelSupport.ToString())
    End With
    With lvwSystemInformation.Items.Add("Network")
      .SubItems.Add(SystemInformation.Network.ToString())
    End With
    With lvwSystemInformation.Items.Add("PenWindows")
      .SubItems.Add(SystemInformation.PenWindows.ToString())
    End With
```

```
    With lvwSystemInformation.Items.Add("PrimaryMonitorMaximizedWindowSize")
SubItems.Add(SystemInformation.PrimaryMonitorMaximizedWindowSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("PrimaryMonitorSize")
      .SubItems.Add(SystemInformation.PrimaryMonitorSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("RightAlignedMenus")
      .SubItems.Add(SystemInformation.RightAlignedMenus.ToString())
    End With
    With lvwSystemInformation.Items.Add("Secure")
      .SubItems.Add(SystemInformation.Secure.ToString())
    End With
    With lvwSystemInformation.Items.Add("ShowSounds")
      .SubItems.Add(SystemInformation.ShowSounds.ToString())
    End With
    With lvwSystemInformation.Items.Add("SmallIconSize")
      .SubItems.Add(SystemInformation.SmallIconSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("ToolWindowCaptionButtonSize")
      .SubItems.Add(SystemInformation.ToolWindowCaptionButtonSize.ToString())
    End With
    With lvwSystemInformation.Items.Add("ToolWindowCaptionHeight")
      .SubItems.Add(SystemInformation.ToolWindowCaptionHeight.ToString())
    End With
    With lvwSystemInformation.Items.Add("UserDomainName")
      .SubItems.Add(SystemInformation.UserDomainName.ToString())
    End With
    With lvwSystemInformation.Items.Add("UserInteractive")
      .SubItems.Add(SystemInformation.UserInteractive.ToString())
    End With
    With lvwSystemInformation.Items.Add("UserName")
      .SubItems.Add(SystemInformation.UserName.ToString())
    End With
    With lvwSystemInformation.Items.Add("VerticalScrollBarArrowHeight")
      .SubItems.Add(SystemInformation.VerticalScrollBarArrowHeight.ToString())
    End With
    With lvwSystemInformation.Items.Add("VerticalScrollBarThumbHeight")
      .SubItems.Add(SystemInformation.VerticalScrollBarThumbHeight.ToString())
    End With
```

```
        With lvwSystemInformation.Items.Add("VerticalScrollBarWidth")
            .SubItems.Add(SystemInformation.VerticalScrollBarWidth.ToString())
        End With
        With lvwSystemInformation.Items.Add("VirtualScreen")
            .SubItems.Add(SystemInformation.VirtualScreen.ToString())
        End With
        With lvwSystemInformation.Items.Add("WorkingArea")
            .SubItems.Add(SystemInformation.WorkingArea.ToString())
        End With
    End Sub

    Private Sub RunMethods()
        ' Environment.GetLogicalDrives trả về một mảng chứa tên các ổ đĩa.

        ' Khi đã có danh sách các ổ đĩa, ta có thể lấy thông tin về các file trên các
        ' đĩa này. Xem các class File và Directory
        lstLogicalDrives.DataSource = _
        Environment.GetLogicalDrives()

        lstCommandLineArgs.DataSource = _
        Environment.GetCommandLineArgs()
    End Sub
End Class
```

66. MSMQ (Microsoft Message Queuing)

Chương trình này chỉ cách gửi một thông điệp MSMQ từ một ứng dụng Windows Forms có chứa instance của một đối tượng đã được serialize sử dụng XML. Thông điệp được nhận và xử lý bởi một Windows Service.



Các chức năng chính:

Phần này có ba project kết hợp làm việc với nhau:

- Một Windows Forms Client dùng để gửi các thông điệp
- Một Windows Service dùng để xử lý các thông điệp gửi từ client.
- Một Class Library cung cấp một class public được dùng để làm nội dung của thông điệp.

Client

Chỉ là một ứng dụng đơn giản cho phép gửi các Order vào hàng đợi (queue). Ngoài ra có thể kiểm tra được các message chưa được xử lý.

Service

Windows Service này cho thấy cách xử lý các message đến trong một hàng đợi MSMQ. Sử dụng một component MSMQ đợi các message đến và chuyển nó thành kiểu MSMQOrder. Sau đó sử dụng thread pool có sẵn của CLR để thực hiện phương thức Process của instance MSMQOrder. Sau đó tiếp tục đợi các message khác. Service này hỗ trợ các sự kiện Start, Stop, Pause, Resume, Shutdown và Power Change.

Class Library

Class Library này, server.dll, cung cấp một class public có tên MSMQOrders. Nó có ba trường Public và một phương thức có tên Process.

Để chạy được các project này, có ba điều cần làm:

- Phải tạo một hàng đợi (queue) private có tên Orders.
- Biên dịch cả ba project theo thứ tự :Class Library, Client, Service.
- Windows Service phải được cài đặt.

Cài đặt Windows Service bằng cách sử dụng công cụ dòng lệnh, InstallUtil.exe, làm theo các bước sau:

- Sau khi biên dịch ba project, mở Visual Studio .NET Command prompt.
- Chuyển đến thư mục bin của Windows Service
- Gõ **installutil WindowsService.exe** và nhấn Enter.
- Start service này bằng dòng lệnh **net start WatchMSMQ**.

Để gỡ bỏ service này ra khỏi hệ thống, gõ lệnh sau vào cửa sổ dòng lệnh:
installutil -u WindowsService.exe

Sau khi đã cài đặt thành công service, chạy client và gửi các message. Để thấy được các message trong queue, Stop service WatchMSMQ và gửi các message từ Client, sau đó nhấn nút List Messages in Queue. Tiếp theo, Start

service WatchMSMQ và nhấn nút List Messages in Queue ta sẽ thấy các message đã được xử lý.

Project: **Server** (Class Library)

MSMQOrders.vb

```
Public Class MSMQOrders
    ' Ta có thể sử dụng các Property cho ứng dụng thực tế
    Public Number As Integer
    Public Customer As String
    Public RequiredBy As Date

    Public Sub Process(ByVal State As Object)
        ' Ghi ra một thông báo để biết message đã được xử lý
        Trace.WriteLine(Number & " - " & Customer & " - " & RequiredBy)
        ' Làm cho thread tạm ngưng trong giây lát để giả lập thực hiện một công
        ' việc thực sự, ví dụ như ghi thông tin vào database
        Threading.Thread.Sleep(2000)
    End Sub
End Class
```

Project: **WindowsService**

app.config

```
<?xml version="1.0" encoding="Windows-1252"?>
<configuration>
  <appSettings>
    <!-- User application and configured property settings go here.-->
    <!-- Example: <add key="settingName" value="settingValue"/> -->

    <!-- Change the period (.) in the path name if you wish to use another machine
    besides the local one. -->
    <!-- Remove the \private$ if you have a public queue. -->
  <add key="msmqOrders.Path" value="FormatName:DIRECT=OS:.\private$\orders" />
  </appSettings>
</configuration>
```

WatchMSMQ.vb

```
Imports System.ServiceProcess
Imports System.Messaging
Imports System.Threading

Public Class WatchMSMQ
  Inherits System.ServiceProcess.ServiceBase

  ' Cần thiết khi kết lại vào queue trong trường hợp ta đã tạm dừng
  Private m_Path As String

#Region " Component Designer generated code "

  Public Sub New()
   MyBase.New()

   ' This call is required by the Component Designer.
   InitializeComponent()

   ' Add any initialization after the InitializeComponent() call
   Me.m_Path = Me.qOrders.Path
  End Sub

  'UserService overrides dispose to clean up the component list.
  Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
   If disposing Then
     If Not (components Is Nothing) Then
       components.Dispose()
     End If
   End If
   MyBase.Dispose(disposing)
  End Sub

  ' The main entry point for the process
  <MTAThread()> _
  Shared Sub Main()
   Dim ServicesToRun() As System.ServiceProcess.ServiceBase

   ' More than one NT Service may run within the same process. To add
   ' another service to this process, change the following line to
```

```
    ' create a second service object. For example,
    '
    '   ServicesToRun = New System.ServiceProcess.ServiceBase () {New Service1,
    New MySecondUserService}
    '
    ServicesToRun = New System.ServiceProcess.ServiceBase() {New
    WatchMSMQ()}

    System.ServiceProcess.ServiceBase.Run(ServicesToRun)
  End Sub

  'Required by the Component Designer
  Private components As System.ComponentModel.IContainer

  ' NOTE: The following procedure is required by the Component Designer
  ' It can be modified using the Component Designer.
  ' Do not modify it using the code editor.
  Friend WithEvents qOrders As System.Messaging.MessageQueue

  <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    Dim configurationAppSettings As System.Configuration.AppSettingsReader = New
    System.Configuration.AppSettingsReader()
    Me.qOrders = New System.Messaging.MessageQueue()
    '
    'qOrders
    '
    Me.qOrders.Formatter = New System.Messaging.XmlMessageFormatter(New
    String() {"Server.MSMQOrders,Server"})
    Me.qOrders.Path = CType(configurationAppSettings.GetValue("msmqOrders.Path",
    GetType(System.String)), String)
    '
    'WatchMSMQ
    '
    Me.CanHandlePowerEvent = True
    Me.CanPauseAndContinue = True
    Me.CanShutdown = True
    Me.ServiceName = "WatchMSMQ"

  End Sub
#End Region
```

```
Protected Overrides Sub OnStart(ByVal args() As String)
 Try
   Me.HookQueue()
 Catch exp As MessageQueueException
   Me.EventLog.WriteEntry(exp.Message)
 Catch exp As Exception
   Me.EventLog.WriteEntry(exp.Message)
 End Try
End Sub

Protected Overrides Sub OnStop()
 UnhookQueue()
End Sub

Protected Overrides Sub OnContinue()
 HookQueue()
End Sub

Protected Overrides Sub OnPause()
 UnhookQueue()
End Sub

Protected Overrides Function OnPowerEvent(ByVal powerStatus As
System.ServiceProcess.PowerBroadcastStatus) As Boolean

End Function

Protected Overrides Sub OnShutdown()
 UnhookQueue()
End Sub

Private Sub HookQueue()
 Try
   If Me.qOrders Is Nothing Then
     Me.qOrders = New System.Messaging.MessageQueue(Me.m_Path)
   End If

   ' Bắt đầu đợi các message tới
   Me.qOrders.BeginReceive()
```

```
    Catch exp As MessageQueueException
      Me.EventLog.WriteEntry(exp.Message)
    Catch exp As Exception
      Me.EventLog.WriteEntry(exp.Message)
    End Try
  End Sub


  Private Sub UnhookQueue()
    Try
      With Me.qOrders
        .Close()
        .Dispose()
      End With

      Me.qOrders = Nothing

    Catch exp As MessageQueueException
      Me.EventLog.WriteEntry(exp.Message)
    Catch exp As Exception
      Me.EventLog.WriteEntry(exp.Message)
    End Try
  End Sub

  Private Sub qOrders_ReceiveCompleted(ByVal sender As System.Object, ByVal e
  As System.Messaging.ReceiveCompletedEventArgs) Handles
  qOrders.ReceiveCompleted
    ' Sự kiện này xảy ra khi một message được nhận
    Try
      ' Lấy message
      Dim m As Message
      m = qOrders.EndReceive(e.AsyncResult)

      ' Ép phần thân message thành một instance của một đối tượng
      Dim o As Server.MSMQOrders
      o = CType(m.Body, Server.MSMQOrders)
```

```
        ' Nếu chúng ta sử dụng dòng code sau:

        ' o.Process()

        ' service của chúng ta sẽ đợi cho đến khi nó kế thúc. Thay vào đó,
        ' ta sử dụng Thread Pool.
        ThreadPool.QueueUserWorkItem(AddressOf o.Process)
        ' tiếp tục đợi các message
        Me.qOrders.BeginReceive()

    Catch exp As MessageQueueException
      Me.EventLog.WriteEntry(exp.Message)
    Catch exp As Exception
      Me.EventLog.WriteEntry(exp.Message)
    End Try
  End Sub
End Class
```

WatchMSMQInstaller.vb

```
Imports System.ComponentModel
Imports System.Configuration.Install

<RunInstaller(True)> Public Class ProjectInstaller
  Inherits System.Configuration.Install.Installer

#Region " Component Designer generated code "

  Public Sub New()
    MyBase.New()

    'This call is required by the Component Designer.
    InitializeComponent()

    'Add any initialization after the InitializeComponent() call

  End Sub
```

```
'Installer overrides dispose to clean up the component list.
Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
  If disposing Then
    If Not (components Is Nothing) Then
      components.Dispose()
    End If
  End If
  MyBase.Dispose(disposing)
End Sub

'Required by the Component Designer
Private components As System.ComponentModel.IContainer

'NOTE: The following procedure is required by the Component Designer
'It can be modified using the Component Designer.
'Do not modify it using the code editor.
Friend WithEvents WatchMSMQServiceProcessInstaller As
System.ServiceProcess.ServiceProcessInstaller
Friend WithEvents WatchMSMQInstaller As
System.ServiceProcess.ServiceInstaller
<System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
 Me.WatchMSMQServiceProcessInstaller = New _
                System.ServiceProcess.ServiceProcessInstaller()
 Me.WatchMSMQInstaller = New System.ServiceProcess.ServiceInstaller()
 '
 'WatchMSMQServiceProcessInstaller
 '
 Me.WatchMSMQServiceProcessInstaller.Account = _
                  System.ServiceProcess.ServiceAccount.LocalSystem
 Me.WatchMSMQServiceProcessInstaller.Password = Nothing
 Me.WatchMSMQServiceProcessInstaller.Username = Nothing
 '
 'WatchMSMQInstaller
 '
   Me.WatchMSMQInstaller.DisplayName = "Watch MSMQ"
 Me.WatchMSMQInstaller.ServiceName = "WatchMSMQ"
 Me.WatchMSMQInstaller.ServicesDependedOn = New String() {"MSMQ"}
 '
 'ProjectInstaller
 '
```

```
    Me.Installers.AddRange(New System.Configuration.Install.Installer() _
            {Me.WatchMSMQInstaller, Me.WatchMSMQServiceProcessInstaller})

  End Sub

#End Region
End Class
```

Project: **Client**

References: Server, System.messaging

app.config

```
<?xml version="1.0" encoding="Windows-1252"?>
<configuration>
  <appSettings>
    <!-- User application and configured property settings go here.-->
    <!-- Example: <add key="settingName" value="settingValue"/> -->

    <!-- Change the period (.) in the path name if you wish to use another machine
    besides the local one. -->
    <!-- Remove the \private$ if you have a public queue. -->
    <add key="qOrders.Path" value="FormatName:DIRECT=OS:.\private$\orders" />
  </appSettings>
</configuration>
```

frmMain.vb

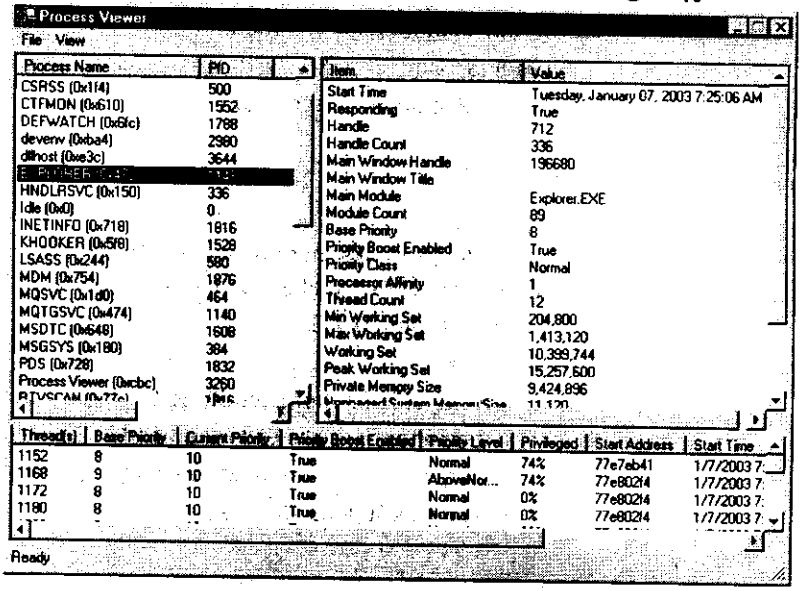
```
Option Strict On
Imports System.Messaging

Public Class frmMain
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

  Private Sub cmdSend_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdSend.Click
    ' Ta tạo một message và gửi nó vào Orders queue
    ' Đường dẫn đến hàng đợi này được lưu trong file app.config
```

```
    Try
        ' Tạo một instance của đối tượng như bình thường
      Dim o As New Server.MSMQOrders()
        ' Gán các thuộc tính
      o.Number = CInt(Me.txtOrderNumber.Text)
      o.Customer = Me.txtCustomer.Text
      o.RequiredBy = CDate(Me.txtReqDate.Text)
        ' Gửi nó vào hàng đợi
      Me.qOrders.Send(o, "New Order: " & o.Number)
        ' Tăng số thứ tự (không cần thiết, chỉ phục vụ cho mục đích hiển thị)
      Me.txtOrderNumber.Text = (o.Number + 1).ToString()
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, _
                      MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
  End Sub

  Private Sub frmMain_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    ' Gán Required Date là 5 ngày kể từ ngày hôm nay
    Dim d As Date = Date.Today.AddDays(5)
    Me.txtReqDate.Text = d.ToShortDateString
  End Sub

  Private Sub cmdScanQ_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdScanQ.Click
    ' Quét queue để lấy thông tin về các message chưa được nhận. Thao tác
    ' này không xóa các message.
    Try
      Me.lstMessages.Items.Clear()
      Dim m As Message
      For Each m In Me.qOrders
        Me.lstMessages.Items.Add(m.Label)
      Next
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK,
MessageBoxIcon.Error)
    End Try
  End Sub
```

```
    Private Sub txtOrderNumber_Validating(ByVal sender As Object, ByVal e As
    System.ComponentModel.CancelEventArgs) Handles txtOrderNumber.Validating
        Dim txt As TextBox = CType(sender, TextBox)
        If Not IsNumeric(txt.Text) Then
            txt.Text = "1"
            MessageBox.Show("Only numbers are allowed for an Order Number." & _
                            " Setting value to 1", Me.Text, MessageBoxButtons.OK, _
                            MessageBoxIcon.Exclamation)
        End If
    End Sub

    Private Sub txtReqDate_Validating(ByVal sender As Object, ByVal e As
    System.ComponentModel.CancelEventArgs) Handles txtReqDate.Validating
        Dim txt As TextBox = CType(sender, TextBox)
        If Not IsDate(txt.Text) Then
            txt.Text = Date.Today.AddDays(5).ToShortDateString()
            MessageBox.Show("Only numbers are allowed for an Order Number." & _
                            " Setting value to : " & Me.txtReqDate.Text, Me.Text, _
                            MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        End If
    End Sub
End Class
```

67. Process Viewer

Chương trình này sử dụng các class trong namespace System.Diagnostics để thu thập thông tin về các chương trình (tiến trình) đang chạy.



Các chức năng chính:

- Chương trình này sử dụng ba ListView trên form Main để hiển thị tất cả các tiến trình đang chạy trên máy tính hiện hành. Khi một tiến trình (process) được chọn, các ListView còn lại sẽ hiển thị các dữ liệu liên quan. Chú ý là dữ liệu của process được lấy từ class ThreadInfo, không phải lấy từ System.Threading.Thread. Ngoài ra còn có một form con để hiển thị chi tiết các module (DLL) được chạy bởi process được chọn.

Có ba process không cung cấp đầy đủ thông tin:

- **_Total**: Cung cấp tóm tắt thời gian thực hiện, sử dụng bộ xử lý
- **Idle**: Không phải là một process thực sự
- **System**: Không có module cung cấp thông tin

Chương trình có sử dụng một ContextMenu và một MainMenu.

Project: **Process Viewer**

frmMain.vb

```
Option Strict On
Imports System.Diagnostics

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Sử dụng Collection để lưu các Process nhằm mục đích truy xuất nhanh hơn
    Private mcolProcesses As New Collection()
    ' Tham chiếu đến Form con để hiển thị chi tiết các module
    Private mfrmMod As frmModules
    ' Các hằng chuỗi để hiển thị trong các listview
    Private Const PID_NA As String = "N/A"
    Private Const PPOCESS_NAME_TOTAL As String = "_Total (0x0)"
    Private Const PROCESS_IDLE As String = "Idle"
    Private Const PROCESS_SYSTEM As String = "System"

      Private mits As ListView.ListViewItemCollection

' Code phát sinh form được bỏ qua

  Private Sub frmMain_Closing(ByVal sender As Object, ByVal e As
   System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
      ' Giải phóng form con nếu nó đang mở
     If Not mfrmMod Is Nothing Then
       Try
         With mfrmMod
           .Owner = Nothing
           .ParentProcess = Nothing
           .Close()
           .Dispose()
         End With
         mfrmMod = Nothing
       Catch exp As Exception
         MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                         MessageBoxIcon.Error)
       End Try
     End If
   End Sub
```

```
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    ' Lấy danh sách các process
  EnumProcesses()
End Sub

Private Sub lvProcesses_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles lvProcesses.SelectedIndexChanged
    ' Xử lý khi chọn process
  Try
    Dim lv As ListView = CType(sender, ListView)

    If lv.SelectedItems.Count = 1 Then
      ' Lấy ID của process từ cột đầu tiên
      Dim strProcessId As String = lv.SelectedItems(0).SubItems(1).Text

      If strProcessId = PID_NA Then
        Me.lvProcessDetail.Items.Clear()
        Me.lvThreads.Items.Clear()
        Exit Sub
      End If

      Dim p As Process

      p = CType(mcolProcesses.Item(strProcessId), Process)
        ' Lấy thông tin mới nhất
      p.Refresh()

        ' Lấy chi tiết của process
      EnumProcess(p)
        ' Lấy chi tiết của thread
      EnumThreads(p)
    End If
  Catch exp As Exception
    MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                    MessageBoxIcon.Error)
  End Try
End Sub
```

```
Private Sub mnuModules_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuModules.Click, ctxViewMods.Click
    ' Mở form con để hiển thị thông tin module
  Try
    Dim lv As ListView = Me.lvProcesses

    If lv.SelectedItems.Count = 1 Then

      Dim strProcessId As String = lv.SelectedItems(0).SubItems(1).Text
      If strProcessId = PID_NA Then
        Exit Sub
      End If

      Dim p As Process

      p = CType(mcolProcesses.Item(strProcessId), Process)

        ' Không thực hiện công việc với process IDLE
        ' Ta sẽ nhận được thông báo lỗi access denied (từ chối truy xuất)
      If p.ProcessName = PROCESS_IDLE Then
        Exit Sub
      End If
        ' Khônh có gì để hiển thị
      If p.ProcessName = PROCESS_SYSTEM Then
        Exit Sub
      End If
      p.Refresh()
      Try
        Dim i As Integer = p.Modules.Count
      Catch exp As System.ComponentModel.Win32Exception
        MessageBox.Show("Sorry, you are not authorized to" & _
        '               "read this information.", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Exit Sub
      End Try
      If mfrmMod Is Nothing Then
        mfrmMod = New frmModules()
      End If
        ' Truyền process cho form được chọn
      mfrmMod.ParentProcess = p
```

```
            ' Lấy thông tin module
            mfrmMod.RefreshModules()
            ' Hiển thị form
            mfrmMod.ShowDialog(Me)
        End If
    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
    End Try
End Sub

Private Sub mnuRefresh_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuRefresh.Click, ctxRefresh.Click
    ' Lấy lại danh sách các process
    Me.sbInfo.Text = "Refreshing list, please wait"
    Me.sbInfo.Refresh()
    EnumProcesses()
    Me.sbInfo.Text = "Ready"
End Sub

Private Sub AddNameValuePair(ByVal Item As String, ByVal SubItem As String)
    ' Thủ tục giúp thêm cặp name/value vào listview
    With mits.Add(Item)
        .SubItems.Add(SubItem)
    End With
End Sub

Private Sub EnumThreads(ByVal p As Process)
    ' Lấy thông tin thread của process
    Try
        Me.lvThreads.Items.Clear()

        Dim strProcessId As String

        If strProcessId = PID_NA Then
            Me.lvThreads.Items.Add(PID_NA)
        Else

            Dim t As ProcessThread
```

```
    ' Các Timespan cho một thread cụ thể
  Dim tpt As TimeSpan
  Dim tppt As TimeSpan
  Dim tupt As TimeSpan

  For Each t In p.Threads
      ' Lấy thời gian của thread và lưu lại
    tppt = t.PrivilegedProcessorTime
    tupt = t.UserProcessorTime
    tpt = t.TotalProcessorTime
      ' % User Processor Time
    Dim strPUPT As String = CDbl(tupt.Ticks / tpt.Ticks).ToString("#0%")
    If tupt.Ticks = 0 Then
      strPUPT = "0%"
    End If
      ' % Privileged Processor Time
    Dim strPPPT As String = CDbl(tppt.Ticks / tpt.Ticks).ToString("#0%")
    If tppt.Ticks = 0 Then
      strPPPT = "0%"
    End If

    Dim strTPT As String
    With tpt
      strTPT = (.Days.ToString("00") & "." & .Hours.ToString("00") & _
            ":" & .Minutes.ToString("00") & ":" & .Seconds.ToString("00"))
    End With

    With Me.lvThreads.Items.Add(t.Id.ToString())
      .SubItems.Add(t.BasePriority.ToString())
      .SubItems.Add(t.CurrentPriority.ToString())
      Try
        .SubItems.Add(t.PriorityBoostEnabled.ToString())
      Catch exp As System.ComponentModel.Win32Exception
        .SubItems.Add("N/A")
      End Try
      Try
        .SubItems.Add(t.PriorityLevel.ToString())
      Catch exp As System.ComponentModel.Win32Exception
        .SubItems.Add("N/A")
      End Try
```

```
            .SubItems.Add(strPPPT)
            .SubItems.Add(Hex(t.StartAddress.ToInt32()).ToLower())
            .SubItems.Add(t.StartTime.ToShortDateString() &" " & _
                                        t.StartTime.ToShortTimeString())
            .SubItems.Add(strTPT)
            .SubItems.Add(strPUPT)
          End With
        Next
      End If
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                      MessageBoxIcon.Error)

    End Try
  End Sub

  Private Sub EnumProcess(ByVal p As Process)
    ' Lấy thông tin process
    Dim lv As ListView = Me.lvProcessDetail
    lv.Items.Clear()
    If p.ProcessName = PROCESS_IDLE Then
      Exit Sub
    End If

    mits = lv.Items

    Const NA As String = "Not Authorized"

    Try
      AddNameValuePair("Start Time", p.StartTime.ToLongDateString() & _
                       " " & p.StartTime.ToLongTimeString())
      AddNameValuePair("Responding", p.Responding.ToString())

      Try
        AddNameValuePair("Handle", p.Handle.ToString())
      Catch exp As System.ComponentModel.Win32Exception
        AddNameValuePair("Handle", NA)
      End Try
```

```
AddNameValuePair("Handle Count", p.HandleCount.ToString("N0"))

Try
  AddNameValuePair("Main Window Handle", _
                   p.MainWindowHandle.ToString())
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Main Window Handle", NA)
End Try

Try
  AddNameValuePair("Main Window Title", p.MainWindowTitle)
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Main Window Title", NA)
End Try

Try
  If p.MainModule Is Nothing Then
    AddNameValuePair("Main Module", String.Empty)
  Else
    AddNameValuePair("Main Module", p.MainModule.ModuleName)
  End If
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Main Module", NA)
End Try

Try
  AddNameValuePair("Module Count", p.Modules.Count.ToString("N0"))
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Module Count", NA)
End Try

AddNameValuePair("Base Priority", p.BasePriority.ToString())

Try
  AddNameValuePair("Priority Boost Enabled", _
                   p.PriorityBoostEnabled.ToString())
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Priority Boost Enabled", NA)
End Try
```

```
Try
  AddNameValuePair("Priority Class", p.PriorityClass.ToString())
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Priority Class", NA)
End Try

Try
  AddNameValuePair("Processor Affinity", _
                   p.ProcessorAffinity.ToInt32.ToString())
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Processor Affinity", NA)
End Try
AddNameValuePair("Thread Count", p.Threads.Count.ToString())

Try
  AddNameValuePair("Min Working Set", _
                   p.MinWorkingSet.ToInt32.ToString("N0"))
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Min Working Set", NA)
End Try
Try
  AddNameValuePair("Max Working Set", _
                   p.MaxWorkingSet.ToInt32.ToString("N0"))
Catch exp As System.ComponentModel.Win32Exception
  AddNameValuePair("Max Working Set", NA)
End Try

AddNameValuePair("Working Set", p.WorkingSet.ToString("N0"))
AddNameValuePair("Peak Working Set", p.PeakWorkingSet.ToString("N0"))

AddNameValuePair("Private Memory Size", _
                 p.PrivateMemorySize.ToString("N0"))
AddNameValuePair("Nonpaged System Memory Size", _
                 p.NonpagedSystemMemorySize.ToString("N0"))
AddNameValuePair("Paged Memory Size", _
                 p.PagedMemorySize.ToString("N0"))
AddNameValuePair("Peak Paged Memory Size", _
                 p.PeakPagedMemorySize.ToString("N0"))
AddNameValuePair("Virtual Memory Size", _
                 p.VirtualMemorySize.ToString("N0"))
```

```
        AddNameValuePair("Peak Virtual Memory Size", _
                         p.PeakVirtualMemorySize.ToString("N0"))

      Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)

      End Try
    End Sub

    Private Sub EnumProcesses()
      ' Liệt kê tất cả các tiến trình
      Try
        Dim Processes() As Process

        Dim tpt As TimeSpan
        Dim tppt As TimeSpan
        Dim tupt As TimeSpan

        Dim mtpt As TimeSpan
        Dim mtppt As TimeSpan
        Dim mtupt As TimeSpan
        Dim i As Integer
        Dim p As Process

        If Not mcolProcesses Is Nothing Then
          mcolProcesses = New Collection()
        End If

        If Me.lvProcesses.Items.Count > 0 Then
          Me.lvProcesses.Items.Clear()
          Me.lvProcessDetail.Items.Clear()
          Me.lvThreads.Items.Clear()
        End If
        Processes = Process.GetProcesses()
```

```
        For Each p In Processes
            mcolProcesses.Add(p, p.Id.ToString())

              ' Lấy thời gian của bộ xử lý
            tppt = p.PrivilegedProcessorTime
            tupt = p.UserProcessorTime
            tpt = p.TotalProcessorTime

            mtpt = mtpt.Add(tpt)
            mtppt = mtppt.Add(tppt)
            mtupt = mtupt.Add(tupt)

              ' % User Processor Time
            Dim strPUPT As String = CDbl(tupt.Ticks / tpt.Ticks).ToString("#0%")
              ' % Privileged Processor Time
            Dim strPPPT As String = CDbl(tppt.Ticks / tpt.Ticks).ToString("#0%")

            Dim strTPT As String
            With tpt
              strTPT = (.Days.ToString("00") & "." & .Hours.ToString("00") & _
                        ":" & .Minutes.ToString("00") & ":" & .Seconds.ToString("00"))
            End With


            With Me.lvProcesses.Items.Add(p.ProcessName & " (0x" & _
                                          Hex(p.Id).ToLower() & ")")
              .SubItems.Add(p.Id.ToString())
              .SubItems.Add(strTPT)
              .SubItems.Add(strPPPT)
              .SubItems.Add(strPUPT)
            End With
        Next

          ' % Total User Processor Time
        Dim mstrPUPT As String = CDbl(mtupt.Ticks / mtpt.Ticks).ToString("#0%")
          ' % Total Privileged Processor Time
        Dim mstrPPPT As String = CDbl(mtppt.Ticks / mtpt.Ticks).ToString("#0%")

        Dim mstrTPT As String
```

```
        With mtpt
          mstrTPT = (.Days.ToString("00") & "." & .Hours.ToString("00") & _
                     ":" & .Minutes.ToString("00") & ":" & .Seconds.ToString("00"))
        End With

        ' Cho vào ListView
        With Me.lvProcesses.Items.Add(PROCESS_NAME_TOTAL)
          .SubItems.Add(PID_NA)
          .SubItems.Add(mstrTPT)
          .SubItems.Add(mstrPPPT)
          .SubItems.Add(mstrPUPT)
        End With
      Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
      End Try
    End Sub
End Class
```

frmModules.vb

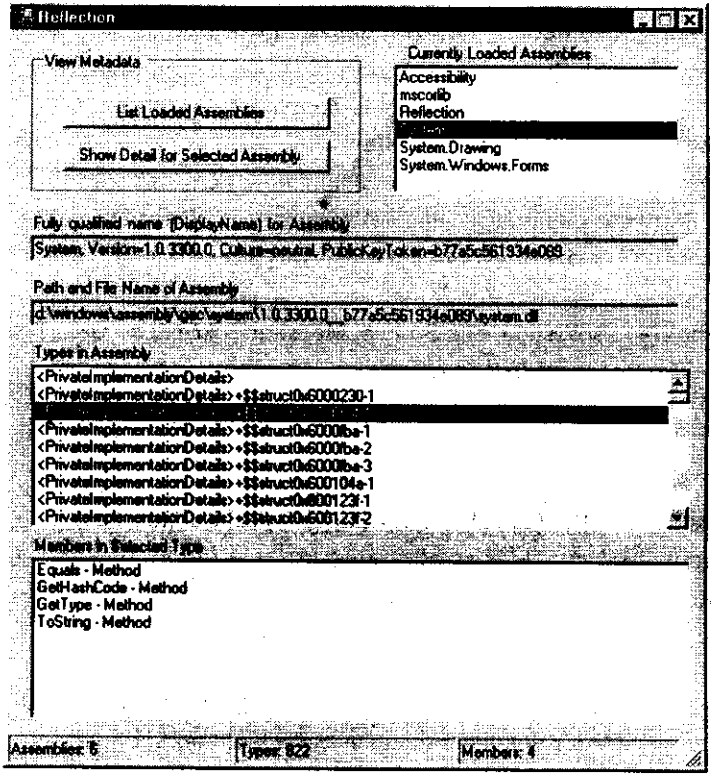
```
Public Class frmModules
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
  Private m_ParentProcess As Process
  Private mits As ListView.ListViewItemCollection
  Private mcolModules As New Collection()

  Friend Property ParentProcess() As Process
    Get
      Return ParentProcess
    End Get
    Set(ByVal Value As Process)
      m_ParentProcess = Value
      If m_ParentProcess Is Nothing Then
        mcolModules = Nothing
      End If
    End Set
  End Property
```

```
Private Sub EnumModules()
 Try
   Me.lvModules.Items.Clear()
   Dim mf As ProcessModule
   If Not mcolModules Is Nothing Then
     mcolModules = New Collection()
   End If
   Dim m As ProcessModule
   For Each m In m_ParentProcess.Modules
     Me.lvModules.Items.Add(m.ModuleName)

     Try
       mcolModules.Add(m, m.ModuleName)
     Catch exp As ArgumentException
       ' Bị trùng lắp, không xử lý và tiếp tục
     Catch exp As Exception
       MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                       MessageBoxIcon.Error)
     End Try

   Next
 Catch exp As Exception
   MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                   MessageBoxIcon.Error)
 End Try
End Sub

Friend Sub RefreshModules()
 Me.sbInfo.Text = "Process = " & m_ParentProcess.ProcessName
 Me.lvModDetail.Items.Clear()
 EnumModules()
End Sub

Private Sub mnuClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuClose.Click
 Me.Hide()

End Sub
```

```
Private Sub EnumModule(ByVal m As ProcessModule)
 Me.lvModDetail.Items.Clear()
 mits = Me.lvModDetail.Items
 Try
   AddNameValuePair("Base Address", Hex(m.BaseAddress.ToInt32).ToLower())
   AddNameValuePair("Entry Point Address", _
                    Hex(m.EntryPointAddress.ToInt32).ToLower())
   AddNameValuePair("File Name", m.FileName)
   AddNameValuePair("File Version", m.FileVersionInfo.FileVersion.ToString())
   AddNameValuePair("File Description", m.FileVersionInfo.FileDescription)
   AddNameValuePair("Memory Size", m.ModuleMemorySize.ToString("N0"))
 Catch exp As Exception
   MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                   MessageBoxIcon.Error)
 End Try
End Sub

Private Sub AddNameValuePair(ByVal Item As String, ByVal SubItem As String)
 With mits.Add(Item)
   .SubItems.Add(SubItem)
 End With
End Sub
Private Sub lvModules_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles lvModules.SelectedIndexChanged
 Try
   Dim lv As ListView = CType(sender, ListView)
   If lv.SelectedItems.Count = 1 Then
     Dim strMod As String = lv.SelectedItems(0).Text
     Dim m As ProcessModule = CType(mcolModules.Item(strMod), _
                                    ProcessModule)
     EnumModule(m)
   End If
 Catch exp As Exception
   MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                   MessageBoxIcon.Error)
 End Try
End Sub

End Class
```

68. Reflection

Chương trình này chỉ cách truy xuất đến các assembly để lấy thông tin các kiểu có trong assembly.



Các chức năng chính:

Chương trình này cho phép chọn một assembly, sau đó kiểm tra metadata của nó, bao gồm cả danh sách của các kiểu có bên trong nó. Người sử dụng cũng có thể chọn một kiểu cụ thể và xem các thành phần của nó. Để hiển thị các assembly đã được load nhấn vào nút List Loaded Assemblies.

Project: **Reflection**

frmMain.vb

```
Option Strict On
Imports System.Reflection ' Namespace dành cho kiểu Assembly

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
```

```
' Biến này được sử dụng để có thể theo dõi assembly hiện hành
' Assembly là từ khóa trong VB.NET, do đó nó phải được đặt trong []
Private CurrentAsm As [Assembly]

Private Sub cmdListLoadedAssemblies_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles cmdListLoadedAssemblies.Click
    ' Thủ tục này lấy tất cả các assembly đã được load và liệt kê chúng trong ListBox

    Dim asms() As [Assembly]

    ' Lấy các assembly đã load từ AppDomain hiện hành
    asms = AppDomain.CurrentDomain.GetAssemblies()

    ' Xóa các phần tử hiện có trong ListBox
    lstLoadedAssemblies.Items.Clear()

    ' Liệt kê các assembly
    Dim asm As [Assembly]
    For Each asm In asms
        lstLoadedAssemblies.Items.Add(asm.GetName.Name)
    Next
    Me.UpdatePanels(Me.lstLoadedAssemblies.Items.Count)

    ' Nếu có các assembly được liệt kê thì cho phép xem chi tiết của nó
    cmdAssemblyDetail.Enabled = (lstLoadedAssemblies.Items.Count > 0)

    'Xóa các chi tiết đã hiển thị trước đó
    txtDisplayName.Text = vbNullString
    txtLocation.Text = vbNullString
    lstTypes.Items.Clear()
    lstMembers.Items.Clear()
    CurrentAsm = Nothing
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    UpdatePanels(0, 0, 0)
End Sub
```

```
Private Sub lstLoadedAssemblies_SelectedIndexChanged(ByVal sender As
System.Object, _
  ByVal e As System.EventArgs) _
  Handles lstLoadedAssemblies.SelectedIndexChanged

  txtDisplayName.Text = vbNullString
  txtLocation.Text = vbNullString
  lstTypes.Items.Clear()
  lstMembers.Items.Clear()
  CurrentAsm = Nothing
End Sub

Private Sub lstTypes_SelectedIndexChanged(ByVal sender As System.Object, _
  ByVal e As System.EventArgs) _
  Handles lstTypes.SelectedIndexChanged
    ' Thủ tục này lấy các thành phần của kiểu được chọn
  Me.Cursor.Current = Cursors.WaitCursor
    ' Lấy tham chiếu đến kiểu được chọn
  Dim t As Type = CurrentAsm.GetType(lstTypes.Text)

    ' Xóa thông tin cũ
  lstMembers.Items.Clear()
    ' Lấy thông tin mới, hiển thị tên và kiểu của thành phần
  Dim mi As MemberInfo
  For Each mi In t.GetMembers()
    lstMembers.Items.Add(mi.Name & " - " & mi.MemberType.ToString)
  Next
  With Me
    .UpdatePanels(.lstLoadedAssemblies.Items.Count, _
                  .lstTypes.Items.Count, .lstMembers.Items.Count)
  End With
  Me.Cursor.Current = Cursors.Default

End Sub
```

```
Private Sub ShowAssemblyDetail(ByVal sender As System.Object, _
 ByVal e As System.EventArgs) _
 Handles cmdAssemblyDetail.Click, _
 lstLoadedAssemblies.DoubleClick
   ' Hiển thị metadata của assembly được chọn.

   ' Bảo đảm là có một assembly được chọn
  If lstLoadedAssemblies.SelectedIndex = -1 Then
    MessageBox.Show("Please select an assembly to view from the list", _
                    Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Warning)
    Exit Sub
  End If
  Me.Cursor.Current = Cursors.WaitCursor
   ' Assembly là từ khóa trong VB.NET, do đó phải đặt nó trong []
  Dim asm As [Assembly]
  For Each asm In AppDomain.CurrentDomain.GetAssemblies()
    If asm.GetName.Name = lstLoadedAssemblies.Text Then Exit For
  Next

  CurrentAsm = asm

  txtDisplayName.Text = CurrentAsm.FullName
  txtLocation.Text = CurrentAsm.Location()
   ' Hiển thị danh sách các kiểu của assembly
  Dim t As Type
  For Each t In CurrentAsm.GetTypes
    lstTypes.Items.Add(t.FullName)
  Next
  Me.UpdatePanels(Me.lstLoadedAssemblies.Items.Count, _
                  Me.lstTypes.Items.Count)
  Me.Cursor.Current = Cursors.Default
End Sub

' Đây là ba hàm overload (UpdatePanels) cập nhật nội dung của thanh trạng thái
Private Sub UpdatePanels(ByVal AsmCount As Integer)

  UpdatePanels(AsmCount, 0)
End Sub
```

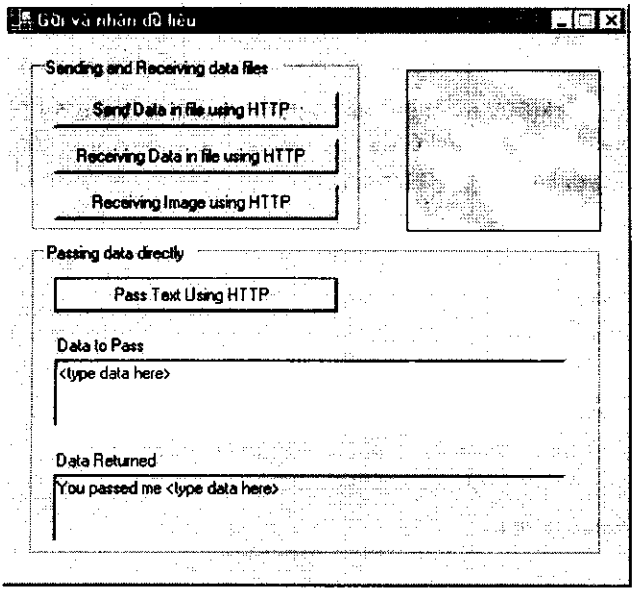
```
    Private Sub UpdatePanels(ByVal AsmCount As Integer, ByVal TypeCount As
    Integer)
        UpdatePanels(AsmCount, TypeCount, 0)
    End Sub

    Private Sub UpdatePanels(ByVal AsmCount As Integer, ByVal TypeCount As
    Integer, ByVal MethodCount As Integer)
        Dim pAsm As StatusBarPanel = Me.sbInfo.Panels(0)
        Dim pTyp As StatusBarPanel = Me.sbInfo.Panels(1)
        Dim pMbr As StatusBarPanel = Me.sbInfo.Panels(2)
        Const ASM_TEXT As String = "Assemblies: {0}"
        Const TYP_TEXT As String = "Types: {0}"
        Const MBR_TEXT As String = "Members: {0}"
        pAsm.Text = String.Format(ASM_TEXT, AsmCount.ToString)
        pTyp.Text = String.Format(TYP_TEXT, TypeCount.ToString)
        pMbr.Text = String.Format(MBR_TEXT, MethodCount.ToString)
    End Sub
End Class
```

Sample.vb

```
Option Strict On
Public Class Sample
    ' Class này phục vụ cho mục đích kiểm tra. Xem metadata của nó bằng cách
    ' chọn Reflection, Reflection.Sample.
    Public x As Integer
    Protected y As Long
    Private z As Short

    Public Sub Method1(ByVal i As Integer)
    End Sub

    Public Sub Method2()
    End Sub

    Public Function Method3() As String
    End Function

    Private Sub Method4()
    End Sub
End Class
```

69. Gửi và nhận dữ liệu thông qua HTTP

Chương trình này chỉ cách gửi và nhận dữ liệu sử dụng HTTP. Gửi và nhận các Stream sử dụng các class trong System.Net.



Các chức năng chính:

- Chương trình có bốt nút lệnh, hai TextBox, và một PictureBox. Và một thư mục ảo (virtual directory) tên là VDir1 dùng để nhận và trả dữ liệu về. Ba nút đầu tiên cho thấy cách gửi và nhận dữ liệu từ một file. Nút cuối cùng cho thấy cách gửi và nhận văn bản từ ứng dụng Web.
- Ứng dụng Windows hoạt động dựa trên bốn file ASPX nằm trên thư mục ảo VDir1.

Chú ý: Cần tạo một Virtual Directory trên IIS có tên là VDir1 (ánh xạ đến VDir1 trong thư mục của Solution này)

Có ba file dữ liệu sử dụng trong chương trình này: datafile.txt, mypic.jpg, myxmlfile.xml, cần có một bản sao của datafile.txt nằm trong cùng thư mục với chương trình .exe để gửi lên server.

Project: **Gui va nhan du lieu**

frmMain.vb

```
Option Strict On
Imports System.IO ' Sử dụng để làm việc với các Stream
Imports System.Net ' Sử dụng để trao đổi với HTTP Server
```

```
Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub cmdPassText_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles cmdPassText.Click
        ' Thủ tục này gửi và nhận văn bản từ HTTP Server.

        Dim req As WebRequest
        Dim rsp As WebResponse

        Try
            ' Tạo một instance WebRequest
            req = WebRequest.Create("http://localhost/VDir1/PassText.aspx")

            ' Sử dụng POST để gửi dữ liệu
            req.Method = "POST"

            ' Gói văn bản vào một StreamWriter
            Dim sw As New StreamWriter(req.GetRequestStream())
            sw.WriteLine(txtDataPassed.Text())
            sw.Close()

            ' Gửi văn bản đến Web Server
            rsp = req.GetResponse()

            ' Gói stream trả về từ Web server vào một StreamReader
            Dim sr As New StreamReader(rsp.GetResponseStream())

            ' Đưa văn bản nhận được vào TextBox
            txtDataReturned.Text = sr.ReadLine()

            MessageBox.Show("Passing text completed successfully!", Me.Text, _
                            MessageBoxButtons.OK, MessageBoxIcon.Information)

        Catch exp As Exception
            MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                            MessageBoxIcon.Stop)
```

```
    Catch exp As Exception
        MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                        MessageBoxIcon.Stop)
    End Try
End Sub

Private Sub cmdReceiveImageFile_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdReceiveImageFile.Click
    ' Phương thức này yêu cầu và nhận về một hình ảnh từ Web Server

    Dim ms As MemoryStream    ' Sử dụng để đưa hình ảnh vào PictureBox
    Dim req As WebRequest

    Try
        ' Tạo đối tượng WebRequest
        req = WebRequest.Create("http://localhost/VDir1/ReceiveImage.aspx")

        ' Sử dụng GET để nhận dữ liệu
        req.Method = "GET"
        Dim rsp As WebResponse = req.GetResponse()

        Try
            ' Tạo một MemortStream để chứa hình ảnh
            ms = New MemoryStream()
            ' Copy hình ảnh vào MemoryStream
            CopyData(rsp.GetResponseStream(), ms)
            ' Hiển thị hình ảnh trên PictureBox
            picDownloadImage.Image = Image.FromStream(ms)

            MessageBox.Show("Image file received successfully!", Me.Text, _
                            MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch exp As Exception
            MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                            MessageBoxIcon.Stop)
        Finally
            If Not ms Is Nothing Then ms.Close()
            If Not rsp Is Nothing Then rsp.GetResponseStream.Close()

        End Try
```

```
    Finally
        ' Bảo đảm các stream được đóng lại
        If Not req Is Nothing Then req.GetRequestStream().Close()
        If Not rsp Is Nothing Then rsp.GetResponseStream().Close()

    End Try
End Sub

Private Sub cmdReceiveDataFile_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmdReceiveDataFile.Click
    ' Phương thức này yêu cầu và nhận về một file XML từ WebServer.

    Dim fs As FileStream    ' Dùng để truy xuất đến file cục bộ
    Dim req As WebRequest
    Try
        ' Tạo một instance Request
        req = WebRequest.Create("http://localhost/VDir1/ReceiveData.aspx")

        ' Sử dụng GET để nhận dữ liệu
        req.Method = "GET"
        Dim rsp As WebResponse = req.GetResponse()

        Try
            ' Tạo file để chứa nội dung
            fs = New FileStream("ReceivedXMLFile.xml", FileMode.Create)

            ' Chép nội dung nhận được vào file vừa tạo
            CopyData(rsp.GetResponseStream(), fs)

        Catch exp As Exception
            MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                            MessageBoxIcon.Stop)
        Finally
            If Not rsp Is Nothing Then rsp.GetResponseStream.Close()
            If Not fs Is Nothing Then fs.Close()
        End Try

        MessageBox.Show("Receive of data file completed successfully!", Me.Text, _
                        MessageBoxButtons.OK, MessageBoxIcon.Information)
```

```
    Catch exp As Exception
      MessageBox.Show(exp.Message, Me.Text, MessageBoxButtons.OK, _
                      MessageBoxIcon.Stop)
    End Try
  End Sub

  Private Sub cmdSendFileData_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdSendFileData.Click
    ' Phương thức này lấy một file cục bộ tên datafile.txt và gửi nội dung của
    ' nó đến Web Server.

    Dim fs As FileStream
    Dim req As WebRequest    ' Tham chiếu đến WebRequest
    Try
      ' Mở file
      fs = New FileStream("datafile.txt", FileMode.Open)
      ' Tạo instance WebRequest
      req = WebRequest.Create("http://localhost/VDir1/SendData.aspx")
      ' Sử dụng Post để gửi nội dung file
      req.Method = "POST"
      ' Copy từ file vào RequestStream
      CopyData(fs, req.GetRequestStream())
    Finally
      If Not req Is Nothing Then req.GetRequestStream.Close()
      If Not fs Is Nothing Then fs.Close()
    End Try

    ' Thực hiện gửi dữ liệu đến Web Server
    Dim rsp As WebResponse = req.GetResponse()
  End Sub

  Private Sub CopyData(ByVal FromStream As Stream, ByVal ToStream As Stream)
    ' Thủ tục này Copy nội dung từ một Stream vào một Stream khác

    ' Dùng để theo dõi số byte đọc từ FromStream
    Dim intBytesRead As Integer
    ' Kích thước lớn nhất cho mỗi lần đọc
    Const intSize As Integer = 4096
    Dim bytes(intSize) As Byte
```

```
        intBytesRead = FromStream.Read(bytes, 0, intSize)
        While intBytesRead > 0
            ToStream.Write(bytes, 0, intBytesRead)
            intBytesRead = FromStream.Read(bytes, 0, intSize)
        End While
    End Sub
End Class
```

Thư mục ảo VDir1 chứa các file sau:

SendData.aspx

```
<%@Page language="VB" debug="true" %>

<%@Import namespace="System.IO" %>

<Script runat=Server>
    Sub Page_Load(sender As Object, args As EventArgs)
    ' Phương thức này nhận nội dung từ RequestStream và copy nội dung vào
    ' một file cục bộ

    Dim fs As FileStream

    Try

        ' Lấy đường dẫn hiện hành của ứng dụng Web
        Dim appbase As String
        appbase = AppDomain.CurrentDomain.SetupInformation.ApplicationBase()

        'Tạo hoặc ghi chồng lên file trong thư mục trên WebServer
        fs = New FileStream(appbase & "datafile.txt", FileMode.Create)

        ' Chép nội dung vào file trên Web Server
        CopyData(Request.InputStream, fs)

    Finally
        Request.InputStream.Close()
        If Not fs Is Nothing Then fs.Close()
    End Try
    End Sub
```

```
    Private Sub CopyData(ByVal FromStream As Stream, ByVal ToStream As Stream)
        Dim nBytesRead As Integer

        Const m_Size As Integer = 4096
        Dim bytes(m_Size) As Byte

        nBytesRead = FromStream.Read(bytes, 0, m_Size)
        While nBytesRead > 0
            ToStream.Write(bytes, 0, nBytesRead)
            nBytesRead = FromStream.Read(bytes, 0, m_Size)
        End While

    End Sub
</Script>
```

PassText.aspx

```
<%@Page language=VB debug="true" %>

<%@Import namespace="System.IO" %>

<Script runat=Server>
    Sub Page_Load(sender As Object, args As EventArgs)
    ' Thủ tục này nhận văn bản  từ client và trả về văn bản cho client.

     Dim sr As StreamReader
     Dim sw As StreamWriter

     Try

       sr = New StreamReader(Request.InputStream)

       dim str As String = sr.ReadToEnd()
       sr.Close

       sw = New StreamWriter(Response.OutputStream)
       sw.WriteLine("You passed me " & str)
       sw.Close
```

```
    Finally

      If Not sw Is Nothing Then sw.Close()
      If Not sr Is Nothing Then sr.Close()

    End Try
  End Sub
</Script>
```

RecieveData.aspx

```
<%@Page language="VB" debug="true" %>

<%@Import namespace="System.IO" %>

<Script runat=Server>
  Sub Page_Load(sender As Object, args As EventArgs)
  ' Phương thức này lấy nội dung trong một file trên server và trả về client

    Dim fs As FileStream

    Try
      Dim appbase As String
      appbase = AppDomain.CurrentDomain.SetupInformation.ApplicationBase()
      fs = New FileStream(appbase & "myXMLfile.xml", FileMode.Open)
      CopyData(fs, Response.OutputStream)
    Finally
      Response.OutputStream.Close()
      If Not fs Is Nothing Then fs.Close()

    End Try
  End Sub

  Private Sub CopyData(ByVal FromStream As Stream, ByVal ToStream As Stream)
      Dim nBytesRead As Integer

      Const m_Size As Integer = 4096
      Dim bytes(m_Size) As Byte
      nBytesRead = FromStream.Read(bytes, 0, m_Size)
```

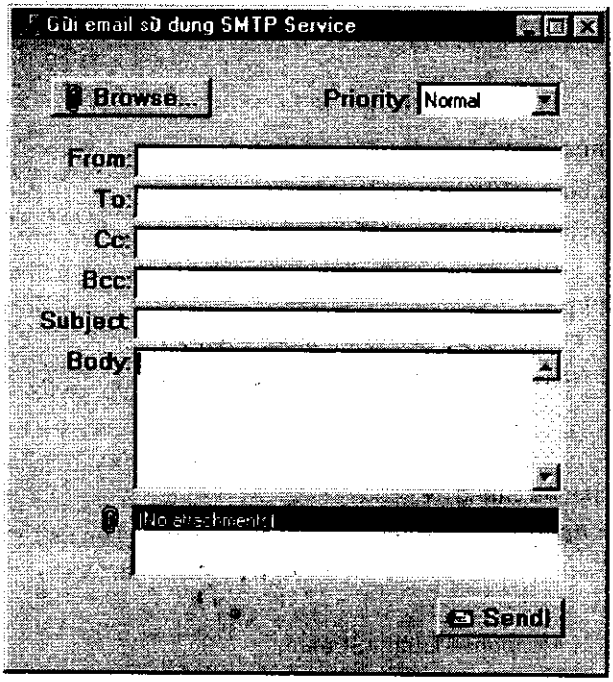
```
        While nBytesRead > 0
            ToStream.Write(bytes, 0, nBytesRead)
            nBytesRead = FromStream.Read(bytes, 0, m_Size)
        End While
    End Sub
</Script>
```

ReceiveImage.aspx

```
<%@Page language="VB" debug="true" %>
<%@Import namespace="System.IO" %>

<Script runat=Server>
    Sub Page_Load(sender As Object, args As EventArgs)
    ' Thủ tục này lấy một hình ảnh từ thư mục trên Web Server và gửi về client
     Dim fs As FileStream
     Try
       Dim appbase As String
       appbase = AppDomain.CurrentDomain.SetupInformation.ApplicationBase()
       fs = New Filestream(appbase & "mypic.jpg", FileMode.Open)

       CopyData(fs, Response.OutputStream)
     Finally
       Response.OutputStream.Close()
       If Not fs Is Nothing Then fs.Close()
     End Try
    End Sub

    Private Sub CopyData(ByVal FromStream As Stream, ByVal ToStream As Stream)
        Dim nBytesRead As Integer
        Const m_Size As Integer = 4096
        Dim bytes(m_Size) As Byte

        nBytesRead = FromStream.Read(bytes, 0, m_Size)
        While nBytesRead > 0
            ToStream.Write(bytes, 0, nBytesRead)
            nBytesRead = FromStream.Read(bytes, 0, m_Size)
        End While
    End Sub
</Script>
```

70. Gửi Email

Chương trình này chỉ cách gửi email với SMTP sử dụng các class trong namespace System.Web.Mail.



Các chức năng chính:

- Chương trình này có một Windows Form với các control tương tự như Microsoft Outlook.
- Khi Form được load nó kiểm tra để bảo đảm người sử dụng có cài đặt SMTP Service và dịch vụ này đang hoạt động.
- Kiểm tra dữ liệu trên các trường "To" và "From" và hiển thị lỗi bằng control ErrorProvider.

Chú ý:

- SMTP Service là một thành phần trong IIS.
- Để gửi được mail, thuộc tính SmtpServer phải được gán địa chỉ của server có SMTP Server đang chạy.

Cấu hình Relay Retrictions cho SMTP Service trên máy cục bộ:

- Trong IIS, nhấn chuột phải vào Default SMTP Virtual Server và chọn Properties.
- Trên hộp thoại mở ra, chọn tab Access.
- Nhấn vào Relay.

- Trong hộp thoại Relay Restrictions, đánh dấu "All except the list below".
- Nhấn OK hai lần để chấp nhận các thay đổi và đóng các hộp thoại.

Project: **Send Mail**

frmMain.vb

```
Option Strict On
Imports System.Web.Mail
Imports System.Text
Imports System.ServiceProcess

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Sử dụng cho WndProc
    Dim WM_SYSCOMMAND As Integer = &H112
    Dim SC_CLOSE As Integer = &HF060

    Friend WithEvents btnSend As System.Windows.Forms.Button
    Friend WithEvents lblFrom As System.Windows.Forms.Label
    Friend WithEvents lblTo As System.Windows.Forms.Label
    Friend WithEvents lblSubject As System.Windows.Forms.Label
    Friend WithEvents lblBody As System.Windows.Forms.Label
    Friend WithEvents lblBCC As System.Windows.Forms.Label
    Friend WithEvents lblCC As System.Windows.Forms.Label
    Friend WithEvents erpEmailAddresses As System.Windows.Forms.ErrorProvider
    Friend WithEvents txtFrom As System.Windows.Forms.TextBox
    Friend WithEvents txtTo As System.Windows.Forms.TextBox
    Friend WithEvents txtSubject As System.Windows.Forms.TextBox
    Friend WithEvents txtBody As System.Windows.Forms.TextBox
    Friend WithEvents txtCC As System.Windows.Forms.TextBox
    Friend WithEvents txtBCC As System.Windows.Forms.TextBox
    Friend WithEvents cboPriority As System.Windows.Forms.ComboBox
    Friend WithEvents btnBrowse As System.Windows.Forms.Button
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents odlgAttachment As System.Windows.Forms.OpenFileDialog
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents lstAttachments As System.Windows.Forms.ListBox

    ' Code phát sinh form được bỏ qua
```

```
Dim arlAttachments As ArrayList

' Kiểm tra các địa chỉ email
Private Sub ValidateEmailAddress(ByVal txt As TextBox)
    ' Xác nhận có phải là chuỗi trống
  If txt.TextLength = 0 Then
    Throw New Exception("Email address is a required field")
  Else
      ' Bảo đảm là có "." và "@" trong các địa chỉ email
    If txt.Text.IndexOf(".") = -1 Or txt.Text.IndexOf("@") = -1 Then
      Throw New Exception("E-mail address must be valid e-mail " & _
        "address format." & ControlChars.Cr & "For example " & _
        "'someone@ntbooksvn.com'")
    End If
  End If
End Sub

' Thủ tục này override WndProc của Form và bắt message Close khi người sử
' dụng đóng form. Nếu không có thủ tục này, khi địa chỉ email nhập vào có lỗi thì
' ErrorProvider không cho đóng form.
Protected Overrides Sub WndProc(ByRef m As System.Windows.Forms.Message)
  If m.Msg = WM_SYSCOMMAND Then
    If m.WParam.ToInt32 = SC_CLOSE Then
        ' Tắt tính năng kiểm tra dữ liệu của các các control
      ' có CausesValidation = True
      txtTo.CausesValidation = False
      txtFrom.CausesValidation = False
    End If
  End If

    ' Truyền các message khác vào WndProc gốc
  MyBase.WndProc(m)
End Sub
```

```
' Xử lý nút Browse. Sử dụng OpenFileDialog để cho phép người sử dụng
' chọn file đính kèm.
Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrowse.Click
    With odlgAttachment
        .InitialDirectory = "C:\"
        .Filter = "All Files (*.*)|*.*|" & _
                  "HTML Files (*.htm;*.html)|*.html|" & _
                  "Microsoft Mail Documents (*.msg)|*.msg|" & _
                  "Word Documents (*.doc)|*.doc|" & _
                  "Excel Files(*.xl*)|*.xl*|" & _
                  "Excel Worksheets (*.xls)|*.xls|" & _
                  "Excel Charts (*.xlc)|*.xlc|" & _
                  "PowerPoint Presentations (*.ppt)|*.ppt|" & _
                  "Text Files (*.txt)|*.txt"
        .FilterIndex = 1

        If .ShowDialog() = DialogResult.OK Then
            If IsNothing(arlAttachments) Then
                arlAttachments = New ArrayList()

                ' Xóa chữ "(No Attachments)" trong ListView
                lstAttachments.Items.Clear()
            End If
            arlAttachments.Add(New MailAttachment(.FileName))

            Dim strFileName() As String = .FileName.Split(New Char() {CChar("\")})
            strFileName.Reverse(strFileName)
            lstAttachments.Items.Add(strFileName(0))
        End If
    End With
End Sub

' Xử lý nút Send. Thủ tục này kiểm tra tính hợp lệ của các địa chỉ; tạo phần thân
' của message bằng StringBuilder, tạo một email và thực hiện gửi nó.
Private Sub btnSend_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSend.Click
    ' Thực hiện kiểm tra các trường To và From, cả hai đều cần để gửi email
    Try
        ValidateEmailAddress(txtFrom)
```

```
    Catch ex As Exception
      txtFrom.Select(0, txtFrom.Text.Length)

      ' Gán vào ErrorProvider chuỗi thông báo để hiển thị
      erpEmailAddresses.SetError(txtFrom, ex.Message)
      Exit Sub
    End Try

    Try
      ValidateEmailAddress(txtTo)
    Catch exp As Exception
      txtTo.Select(0, txtTo.Text.Length)

      erpEmailAddresses.SetError(txtTo, exp.Message)
      Exit Sub
    End Try

    Dim sb As New StringBuilder()

    ' Tạo phần thân của email
    sb.Append("The following email was sent to you from the Send Mail " & _
      "sample application:")
    sb.Append(vbCrLf)
    sb.Append(vbCrLf)
    sb.Append("SUBJECT: ")
    sb.Append(Trim(txtSubject.Text))
    sb.Append(vbCrLf)
    sb.Append(vbCrLf)
    sb.Append("MESSAGE: ")
    sb.Append(Trim(txtBody.Text))
    sb.Append(vbCrLf)

    ' Tạo một email message cũng tương tự như là tạo một instance của
    ' một class và gán một số thuộc tính.
    Dim mailMsg As New MailMessage()
    With mailMsg
      .From = txtFrom.Text.Trim
      .To = txtTo.Text.Trim
      .Cc = txtCC.Text.Trim
      .Bcc = txtBCC.Text.Trim
```

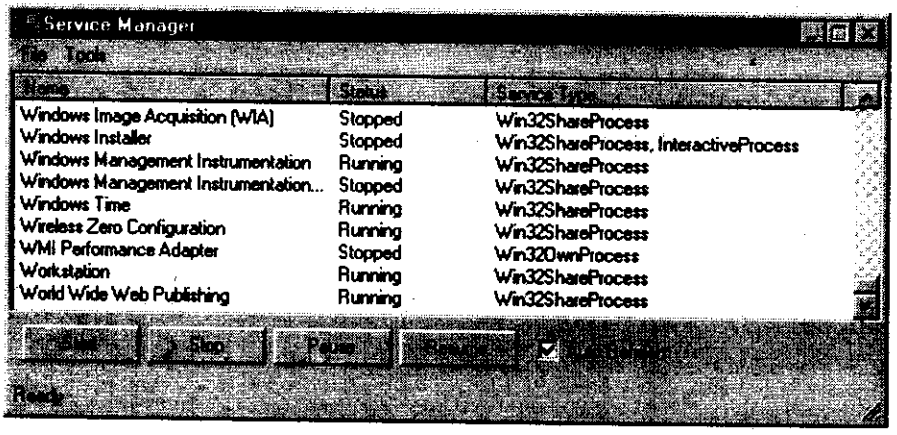
```
        .Subject = txtSubject.Text.Trim
        .Body = sb.ToString
        .Priority = CType(cboPriority.SelectedIndex, MailPriority)

        If Not IsNothing(arlAttachments) Then
          Dim mailAttachment As Object
          For Each mailAttachment In arlAttachments
            .Attachments.Add(mailAttachment)
          Next
        End If
      End With

      SmtpMail.SmtpServer = "localhost"

      Try
        SmtpMail.Send(mailMsg)
        lstAttachments.Items.Clear()
        lstAttachments.Items.Add("(No Attachments)")

        MessageBox.Show("Your email has been successfully sent!", _
          "Email Send Status", MessageBoxButtons.OK, _
          MessageBoxIcon.Information)
      Catch exp As Exception
        MessageBox.Show("The following problem occurred when attempting to " & _
          "send your email: " & exp.Message, _
          Me.Text, MessageBoxButtons.OK, MessageBoxIcon.Error)
      End Try
    End Sub

    ' Xử lý sự kiện phát sinh khi control được kiểm tra hợp lệ
    Private Sub emailAddresses_Validated(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles txtFrom.Validated, txtTo.Validated

      ' Xóa thông báo lỗi
      erpEmailAddresses.SetError(CType(sender, TextBox), "")
    End Sub
```

```
' Xử lý kiểm tra tính hợp lệ cho các TextBox
Private Sub emailAddresses_Validating(ByVal sender As System.Object, ByVal e
As System.ComponentModel.CancelEventArgs) Handles txtFrom.Validating,
txtTo.Validating
    Dim txt As TextBox = CType(sender, TextBox)

    Try
        ValidateEmailAddress(txt)
    Catch exp As Exception
        ' Bỏ qua thao tác và chọn văn bản để người sử dụng sửa lỗi
        e.Cancel = True
        txt.Select(0, txt.Text.Length)

        ' Gán chuỗi thông báo vào ErrorProvider
        erpEmailAddresses.SetError(txt, exp.Message)
    End Try
End Sub

' Kiểm tra SMTP Service được cài đặt và đang chạy
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    ' Bảo đảm SMTP Service được cài đặt
    Dim services() As ServiceController = ServiceController.GetServices
    Dim service As ServiceController
    Dim blnHasSmtpService As Boolean = False

    ' Duyệt qua tất cả các Service có trên máy tính để tìm SMTP Service
    For Each service In services
        If service.ServiceName.ToLower = "smtpsvc" Then
            blnHasSmtpService = True
            Exit For
        End If
    Next

    If Not blnHasSmtpService Then
        MessageBox.Show("You do not have SMTP Service installed on this " & _
            "machine.", Me.Text, _
            MessageBoxButtons.OK, MessageBoxIcon.Information)
    End If
```

```
        ' Bảo đảm là SMTP Service đang chạy, nếu chưa hoạt động thì cho nó hoạt động
        If Not service.Status = ServiceControllerStatus.Running Then
            Dim frmStatusMessage As New frmStatus()
            frmStatusMessage.Show("SMTP Service not currently running. " & _
                "Starting service...")
            Try
                service.Start()
                frmStatusMessage.Close()
            Catch
                MessageBox.Show("There was an error when attempting " & _
                    "to start SMTP Service. ", Me.Text, _
                    MessageBoxButtons.OK, MessageBoxIcon.Error)
            End Try
        End If

        ' Đưa các giá trị độ ưu tiên vào ComboBox Priority
        With cboPriority
            .Items.AddRange(New String() {"Normal", "Low", "High"})
            .SelectedIndex = 0
        End With
    End Sub

End Class
```

71. Service Manager

Chương trình này chỉ cách quản lý các dịch vụ (Windows Service) trên Windows NT, Windows 2000 và Windows XP.



Các chức năng chính:

- Chương trình này sử dụng class ServiceController trong namespace System.ServiceProcess. Chương trình cho phép Start, Stop, Pause và Resume các service. Một Timer sẽ chạy mỗi 5 giây để cập nhật trạng thái của các service.

Project: **Service Manager**

frmMain.vb

```
Public Class frmMain
  Inherits System.Windows.Forms.Form

  ' Sử dụng để truy xuất đến service được chọn
  Private msvc As ServiceController
  Private mcolSvcs As New Collection()
  ' Sử dụng để điều khiển cập nhật giao diện
  Private fUpdatingUI As Boolean

' Code phát sinh form được bỏ qua
  Private Sub chkAutoRefresh_CheckedChanged(ByVal sender As System.Object,
  ByVal e As System.EventArgs) Handles chkAutoRefresh.CheckedChanged
    ' Bật hoặc tắt Timer
    If Me.chkAutoRefresh.CheckState = CheckState.Checked Then
      Me.tmrStatus.Enabled = True
    Else
      Me.tmrStatus.Enabled = False
    End If
  End Sub

  Private Sub cmdPause_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdPause.Click
    Try
      msvc.Pause()
      fUpdatingUI = True
      UpdateUIForSelectedService()
      fUpdatingUI = False
```

```
    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
    End Try
  End Sub

  Private Sub cmdResume_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdResume.Click
    Try
      msvc.Continue()
      fUpdatingUI = True
      UpdateUIForSelectedService()
      fUpdatingUI = False
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                      MessageBoxIcon.Error)
    End Try
  End Sub

  Private Sub cmdStart_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdStart.Click
    Try
      msvc.Start()
      fUpdatingUI = True
      UpdateUIForSelectedService()
      fUpdatingUI = False

    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                      MessageBoxIcon.Error)
    End Try
  End Sub

  Private Sub cmdStop_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles cmdStop.Click
    Try
      msvc.Stop()
      fUpdatingUI = True
      UpdateUIForSelectedService()
      fUpdatingUI = False
```

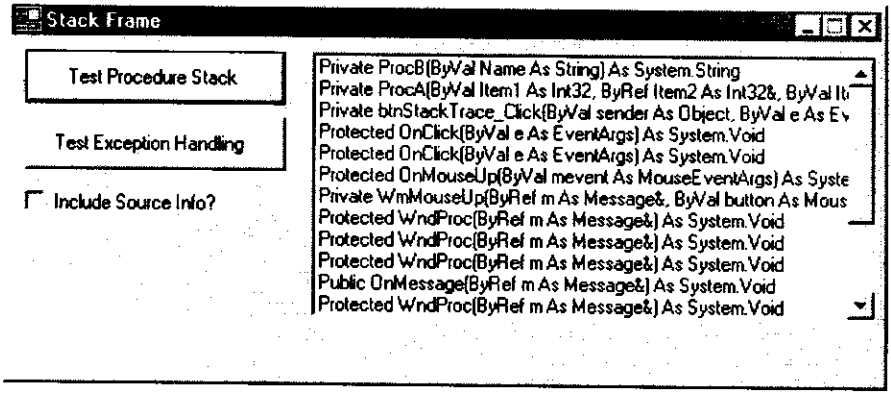
```
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                      MessageBoxIcon.Error)
    End Try
  End Sub

  Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    EnumServices()
  End Sub

  Private Sub lvServices_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles lvServices.SelectedIndexChanged
    fUpdatingUI = True
    UpdateUIForSelectedService()
    fUpdatingUI = False
  End Sub

  Private Sub tmrStatus_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrStatus.Tick
    If Not fUpdatingUI Then
      UpdateServiceStatus()
    End If
  End Sub

  Private Sub EnumServices()
    ' Lấy danh sách các service và cho thông tin của chúng vào listview
    Try
      fUpdatingUI = True
      Me.sbInfo.Text = "Loading Service List, pleasse wait"
      Me.sbInfo.Refresh()

      Me.lvServices.Items.Clear()

      If Not mcolSvcs Is Nothing Then
        mcolSvcs = New Collection()
      End If

      Dim svc As ServiceController
      Dim svcs As ServiceController() = ServiceController.GetServices()
```

```
        For Each svc In svcs
          With Me.lvServices.Items.Add(svc.DisplayName)
            .SubItems.Add(svc.Status.ToString())
            .SubItems.Add(svc.ServiceType.ToString())
          End With
          mcolSvcs.Add(svc, svc.DisplayName)
        Next svc

      Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
      Finally
        Me.sbInfo.Text = "Ready"
        fUpdatingUI = False
      End Try
    End Sub

    Private Sub UpdateServiceStatus()
      ' Kiểm tra từng service
      Try
        fUpdatingUI = True
        Me.sbInfo.Text = "Checking Service Status . . "
        Me.sbInfo.Refresh()

        Dim lvi As ListViewItem
        For Each lvi In Me.lvServices.Items
          msvc = CType(mcolSvcs.Item(lvi.Text), ServiceController)
          msvc.Refresh()

          lvi.SubItems(1).Text = msvc.Status.ToString()
        Next lvi
        UpdateUIForSelectedService()
      Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
      Finally
        Me.sbInfo.Text = "Ready"
        fUpdatingUI = False
      End Try
    End Sub
```

```
Private Sub UpdateUIForSelectedService()
    ' Cập nhật lại trạng thái các nút tương ứng với mỗi service
    Dim strName As String
    Dim i As Integer
    Try
      If Me.lvServices.SelectedItems.Count = 1 Then
        strName = Me.lvServices.SelectedItems(0).SubItems(0).Text
        msvc = CType(mcolSvcs.Item(strName), ServiceController)
        With msvc
          Me.cmdStart.Enabled = (.Status = ServiceControllerStatus.Stopped)
          Me.cmdStop.Enabled = (.CanStop AndAlso (Not .Status = _
                                        ServiceControllerStatus.Stopped))
          Me.cmdPause.Enabled = (.CanPauseAndContinue AndAlso _
                                    (Not .Status = ServiceControllerStatus.Paused))
          Me.cmdResume.Enabled = (.Status = ServiceControllerStatus.Paused)
        End With
      End If
    Catch exp As Exception
      MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                          MessageBoxIcon.Error)
    End Try
  End Sub

  Private Sub mnuRefresh_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuRefresh.Click
    If Not fUpdatingUI Then
      UpdateServiceStatus()
    End If
  End Sub

  Private Sub mnuRelist_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles mnuRelist.Click
    If Not fUpdatingUI Then
      EnumServices()
    End If
  End Sub
End Class
```

72. Stack Frame

Chương trình này chỉ cách sử dụng class System.Diagnostics.StackFrame và hai class khác là System.Diagnostics.StackTrace và System.Reflection.MethodInfo. Các class này cho phép làm việc với "call stack".



Các chức năng chính:

- Trên form chính, nhấn vào nút "**Test Procedure Stack**" để gọi một thủ tục có nhiều cấp gọi (độ sâu) (Sự kiện Click của nút này gọi ProcA, ProcA gọi ProcB, ProcB hiển thị call stack).
- Nhấn vào nút **Test Exception Handing** để xử lý Stack Trace đã kết với một exception.

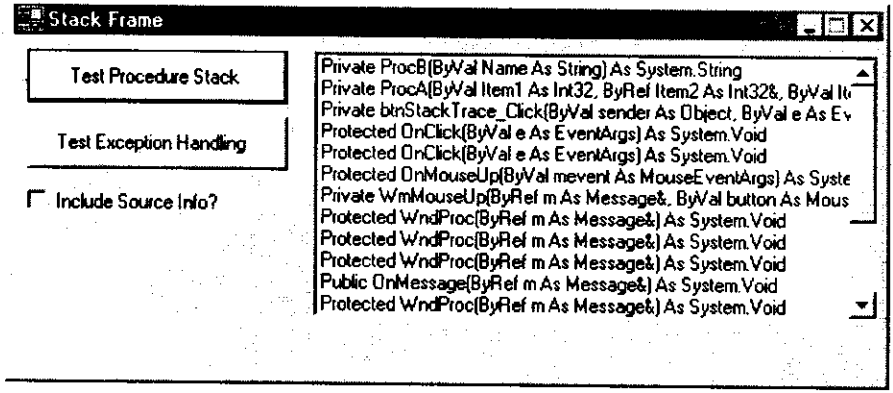
Project: **Stack Frame**

frmMain.vb

```
Option Strict On

Imports System.Diagnostics
Imports System.Reflection
Imports System.Text
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua
```

72. Stack Frame

Chương trình này chỉ cách sử dụng class System.Diagnostics.StackFrame và hai class khác là System.Diagnostics.StackTrace và System.Reflection.MethodInfo. Các class này cho phép làm việc với "call stack".



Các chức năng chính:

- Trên form chính, nhấn vào nút "**Test Procedure Stack**" để gọi một thủ tục có nhiều cấp gọi (độ sâu) (Sự kiện Click của nút này gọi ProcA, ProcA gọi ProcB, ProcB hiển thị call stack).
- Nhấn vào nút **Test Exception Handing** để xử lý Stack Trace đã kết với một exception.

Project: **Stack Frame**

frmMain.vb

```
Option Strict On

Imports System.Diagnostics
Imports System.Reflection
Imports System.Text
Imports System.IO

Public Class frmMain
    Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua
```

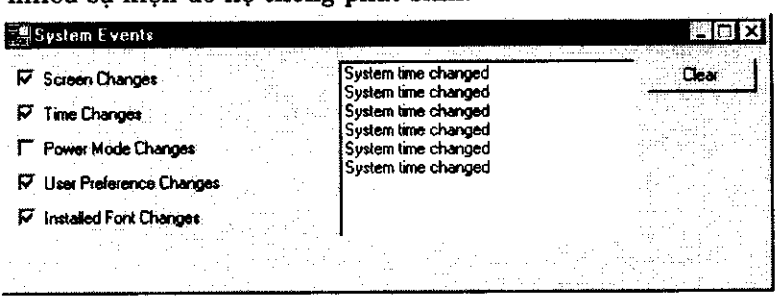
```
Private Sub btnException_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnException.Click

    ' Có thể truyền một đối tượng Exception vào constructor của class StackTrace.
    Try
        ProcException1(1, 2)
    Catch exp As Exception
        GetFullStackFrameInfo(New StackTrace(exp))
    End Try
End Sub

' Thủ tục này, cùng với ProcA và ProcB, cho phép biểu diễn cách ta có thể làm
' việc trên các stack frame tại bất kỳ vị trí nào. Trong trường hợp này, code sử
' dụng kiểu MethodInfo để hiển thị thông tin về stack frame hiện hành.
Private Sub btnStackTrace_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStackTrace.Click
    ProcA(1, 2, "Hello")
End Sub

Private Function GetFullStackFrameInfo(ByVal st As StackTrace) As String
    Dim fc As Integer = st.FrameCount
    Dim i As Integer

    ' Duyệt qua các stack frame, bắt đầu từ thủ tục hiện hành. Ta phải lấy các
    ' frame theo chỉ số, sử dụng phương thức GetFrame
    lstStackItems.Items.Clear()
    For i = 0 To fc - 1
        lstStackItems.Items.Add( _
         GetStackFrameInfo(st.GetFrame(i)))
    Next i
End Function

Private Function GetStackFrameInfo(ByVal sf As StackFrame) As String
    ' Trả về chuỗi chứa thông tin của một stack frame
    Dim pi As ParameterInfo
    Dim strParams As String
    Dim strFileName As String
    Dim i As Integer
    Dim mi As MethodInfo
    Dim typ As Type
```

```
    Dim strOut As String

    mi = CType(sf.GetMethod(), MethodInfo)

    If mi.IsPrivate Then
      strOut &= "Private "
    ElseIf mi.IsPublic Then
      strOut &= "Public "
    ElseIf mi.IsFamily Then
      strOut &= "Protected "
    ElseIf mi.IsAssembly Then
      strOut &= "Friend "
    End If

    If mi.IsStatic Then
      strOut &= "Shared "
    End If

    strOut &= mi.Name & "("

    Dim piList() As ParameterInfo = _
     sf.GetMethod.GetParameters()

    strParams = String.Empty
    For Each pi In piList
      strParams &= String.Format(", {0} {1} As {2}", _
       IIf(pi.ParameterType.IsByRef, "ByRef", "ByVal"), _
       pi.Name, pi.ParameterType.Name)
    Next pi
    ' Bỏ dấu ", " đầu tiên nếu nó tồn tại
    If strParams.Length > 2 Then
      strOut &= strParams.Substring(2)
    End If
    ' Lấy kiểu trả về của một thủ tục và nối nó vào chuỗi kết quả
    typ = mi.ReturnType
    strOut &= ") As " & typ.ToString
    Return strOut
  End Function
```

```
    ' ProcException1 đi qua bốn cấp để biểu diễn StackTrace khi xử lý exception

    Private Sub ProcException1(ByVal x As Integer, ByVal y As Integer)
        ProcException2("Mike", 12)
    End Sub

    Private Sub ProcException2(ByVal Name As String, ByVal Size As Long)
        ProcException3()
    End Sub

    Private Function ProcException3() As String
        Return ProcException4("mike@microsoft.com")
    End Function

    Private Function ProcException4(ByVal EmailAddress As String) As String
        Throw New ArgumentException("This is a fake exception!")
    End Function

    Private Sub ProcA(ByVal Item1 As Integer, ByRef Item2 As Integer, ByVal Item3
    As String)
        Dim strResults As String = ProcB(String.Concat(Item1, Item2, Item3))
    End Sub

    Private Function ProcB(ByVal Name As String) As String
        GetFullStackFrameInfo(New StackTrace())
    End Function
End Class
```

73. System Events

Tìm hiểu class SystemEvents, class này cho phép các ứng dụng tương tác với nhiều sự kiện do hệ thống phát sinh.



Các chức năng chính:

Biểu diễn năm sự kiện có sẵn của class SystemEvents. Mặc dù có nhiều sự kiện hơn để thử nhưng chương trình này chỉ các sự kiện có thể dễ dàng kiểm tra ở ứng dụng client.

- Screen changes – Thay đổi độ phân giải của màn hình.
- Time changes – Thay đổi ngày giờ hệ thống.
- Power mode changes – Nếu chạy trên Laptop, thay đổi trạng thái nguồn.
- User preference changes – Thay đổi bất kỳ cấu hình nào của Windows.
- Installed font changes – Khi font được cài đặt thêm hoặc xóa.

Project: **System Events**

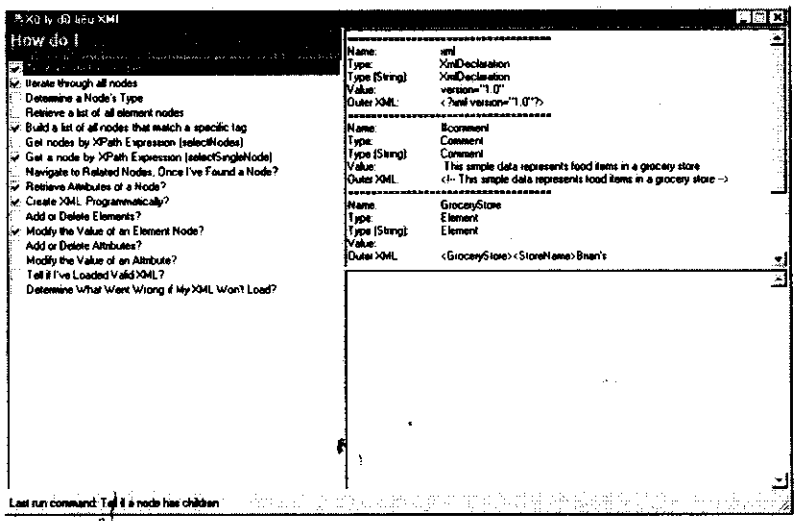
frmMain.vb

```
Option Strict On
Imports Microsoft.Win32

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnClear.Click
        lstResults.Items.Clear()
    End Sub

    Private Sub chkFontChanges_CheckedChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles chkFontChanges.CheckedChanged
        If chkFontChanges.Checked Then
            AddHandler SystemEvents.InstalledFontsChanged, _
            AddressOf FontHandler
        Else
            RemoveHandler SystemEvents.InstalledFontsChanged, _
            AddressOf FontHandler
        End If
    End Sub
```

```
Private Sub chkPowerMode_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkPowerMode.CheckedChanged
    If chkPowerMode.Checked Then
        AddHandler SystemEvents.PowerModeChanged, _
         AddressOf PowerHandler
    Else
        RemoveHandler SystemEvents.PowerModeChanged, _
         AddressOf PowerHandler
    End If
End Sub

Private Sub chkScreenChanges_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
chkScreenChanges.CheckedChanged

    If chkScreenChanges.Checked Then
        AddHandler SystemEvents.DisplaySettingsChanged, _
         AddressOf ScreenHandler
    Else
        RemoveHandler SystemEvents.DisplaySettingsChanged, _
         AddressOf ScreenHandler
    End If
End Sub

Private Sub chkTimeChanges_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkTimeChanges.CheckedChanged
    If chkTimeChanges.Checked Then
        AddHandler SystemEvents.TimeChanged, _
         AddressOf TimeHandler
    Else
        RemoveHandler SystemEvents.TimeChanged, _
         AddressOf TimeHandler
    End If
End Sub

Private Sub chkUserPreferences_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
chkUserPreferences.CheckedChanged
    If chkUserPreferences.Checked Then
        AddHandler SystemEvents.UserPreferenceChanged, _
         AddressOf PreferenceChangedHandler
```

```
    Else
        RemoveHandler SystemEvents.UserPreferenceChanged, _
        AddressOf PreferenceChangedHandler
    End If
End Sub

Private Sub AddToList(ByVal Text As String)
    lstResults.Items.Insert(0, Text)
End Sub

Private Sub FontHandler(ByVal sender As Object, ByVal e As EventArgs)
    ' Sử dụng class InstalledFontCollection để xác định danh sách các font đã cài đặt
    AddToList("Installed fonts changed.")
End Sub


Private Sub PowerHandler(ByVal sender As Object, ByVal e As
PowerModeChangedEventArgs)
    ' e.Mode trả về một trong các giá trị Microsoft.Win32.PowerModes.Resume,
    ' StatusChange, Suspend.
    ' Thay đổi trạng thái nguồn xảy ra khi đổi từ pin sang nguồn cắm hoặc từ
    ' nguồn cắm sang pin.
    AddToList("Power changed to: " & e.Mode.ToString)
End Sub

Private Sub PreferenceChangedHandler(ByVal sender As Object, ByVal e As
UserPreferenceChangedEventArgs)
' e.Category trả về một trong các giá trị của Microsoft.Win32.UserPreferenceCategory
    AddToList("You changed a setting: " & e.Category.ToString)
End Sub

Private Sub ScreenHandler(ByVal sender As Object, ByVal e As EventArgs)
    ' Sử dụng class Screen để lấy thông tin cấu hình màn hình
    AddToList("Screen resolution changed")
End Sub

Private Sub TimeHandler(ByVal sender As Object, ByVal e As EventArgs)
    AddToList("System time changed")
End Sub
End Class
```

74. Sử dụng class XMLDocument



Chương trình này cho thấy cách tận dụng các điểm mạnh của class XMLDocument. Class này đưa ra mô hình đối tượng tài liệu XML (XML Document Object Model), hay DOM, sử dụng để xử lý dữ liệu XML.

Các chức năng chính:

- Chương trình này chỉ cách sử dụng DOM bằng nhiều tác vụ khác nhau. Danh sách các checkbox liệt kê các chọn lựa để làm việc với các tài liệu XML, chọn checkbox để thực hiện công việc.
- Khi Form được load, nó kiểm tra sự tồn tại của ba file: simple.xml, new.xml và bad.xml, nếu một trong ba file không tồn tại thì chương trình không sử dụng được. Để chạy được chương trình này cần đặt ba file trên (có trong cùng thư mục của solution) vào chung thư mục với file .exe (bin).

Project: **Xu ly du lieu XML**

frmMain.vb

```
Option Strict On

' Sử dụng để truy xuất nhiều class về XML
Imports System.Xml
' Sử dụng để truy xuất đến class File dùng để kiểm tra sự tồn tại của file
Imports System.IO
```

```
Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Các biến lưu các tên file
    Private mstrModifyFile As String
    Private mstrSimpleFile As String
    Private mstrBadFile As String

    ' Dùng để phân tách phần tiêu đề với văn bản kết quả
    Private mstrLine As New String(System.Convert.ToChar("="), 35)

    ' Các biến cấp Module để làm việc với DOM
    Private mxDoc As XmlDocument
    Private mxNode As XmlNode
    Private mxNodeList As XmlNodeList

    ' StringWriter để xây dựng các khối văn bản có các CRLF (ký tự điều khiển dòng mới)
    Dim msw As New StringWriter()

    ' Các lệnh sẽ chạy. Được đưa vào ListBox khi Form_Load
    Private Const CMD_LOAD_XML_FILE As String = "Load XML File"
    Private Const CMD_LOAD_XML_STRING As String = "Load XML from Strin
    Private Const CMD_TEST_FOR_CHILD_NODES As String = _
                                            "Tell if a node has children"
    Private Const CMD_ITERATE_ALL_NODES As String = "Iterate through all nodes"
    Private Const CMD_DETERMINE_NODE_TYPE As String = _
                                            "Determine a Node's Type"
    Private Const CMD_LIST_ALL_ELEMENT_NODES As String = _
                                            "Retrieve a list of all element nodes"
    Private Const CMD_LIST_ELEMENTS_BY_TAG As String = _
                                "Build a list of all nodes that match a specific tag"
    Private Const CMD_SELECT_NODES As String = _
                                "Get nodes by XPath Expression (selectNodes)"
    Private Const CMD_SELECT_NODE As String = _
                                "Get a node by XPath Expression (selectSingleNode)"
    Private Const CMD_NAVIGATE_RELATED_NODES As String = _
                                "Navigate to Related Nodes, Once I've Found a Node?"
    Private Const CMD_RETRIEVE_ATTRIBUTES As String = _
                                "Retrieve Attributes of a Node?"
    Private Const CMD_CREATE_XML As String = "Create XML Programmatically?"
```

```
Private Const CMD_ADD_OR_DELETE_ELEMENTS As String = _
                                        "Add or Delete Elements?"
Private Const CMD_ADD_OR_DELETE_ATTRIBUTES As String = _
                                        "Add or Delete Attributes?"
Private Const CMD_MODIFY_ELEMENT As String = _
                                    "Modify the Value of an Element Node?"
Private Const CMD_MODIFY_ATTRIBUTE As String = _
                                    "Modify the Value of an Attribute?"
Private Const CMD_VALID_XML As String = "Tell if I've Loaded Valid XML?"
Private Const CMD_PARSE_ERRORS As String = _
                                "Determine What Went Wrong if My XML Won't Load?"

Private Const DEF_XPATH_EXP_FIND_NODES As String = "//Item[New]"
Private Const DEF_XPATH_EXP_FIND_NODE As String = _
                                        "//Department[@Name='Fruits']"

' Code phát sinh form được bỏ qua

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim strPath As String

    ' Lấy đường dẫn của chương trình đang chạy
    strPath = AppDomain.CurrentDomain.SetupInformation.ApplicationBase

    ' Nối đường dẫn vào tên file
    mstrSimpleFile = strPath & "Simple.xml"
    mstrModifyFile = strPath & "New.XML"
    mstrBadFile = strPath & "Bad.xml"

    Dim blnFileNotFound As Boolean = False
    Dim strMsg As String = "The file '{0}' was not found. Please place it in the same
directory as the application EXE and restart the application."

    ' Đảm bảo là các file tồn tại
    If Not File.Exists(mstrSimpleFile) Then
      MessageBox.Show(String.Format(strMsg, mstrSimpleFile), Me.Text, _
                      MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
      blnFileNotFound = True
    End If
```

```
    If Not File.Exists(mstrModifyFile) Then
      MessageBox.Show(String.Format(strMsg, mstrModifyFile), Me.Text, _
                      MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
      blnFileNotFound = True
    End If
    If Not File.Exists(mstrBadFile) Then
      MessageBox.Show(String.Format(strMsg, mstrModifyFile), Me.Text, _
                      MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
      blnFileNotFound = True
    End If

    ' Đưa các chức năng vào ListBox
    With Me.lstCommands.Items
      .Add(CMD_LOAD_XML_FILE, False)
      .Add(CMD_LOAD_XML_STRING, False)
      .Add(CMD_TEST_FOR_CHILD_NODES, False)
      .Add(CMD_ITERATE_ALL_NODES, False)
      .Add(CMD_DETERMINE_NODE_TYPE, False)
      .Add(CMD_LIST_ALL_ELEMENT_NODES, False)
      .Add(CMD_LIST_ELEMENTS_BY_TAG, False)
      .Add(CMD_SELECT_NODES, False)
      .Add(CMD_SELECT_NODE, False)
      .Add(CMD_NAVIGATE_RELATED_NODES, False)
      .Add(CMD_RETRIEVE_ATTRIBUTES, False)
      .Add(CMD_CREATE_XML, False)
      .Add(CMD_ADD_OR_DELETE_ELEMENTS, False)
      .Add(CMD_MODIFY_ELEMENT, False)
      .Add(CMD_ADD_OR_DELETE_ATTRIBUTES, False)
      .Add(CMD_MODIFY_ATTRIBUTE, False)
      .Add(CMD_VALID_XML, False)
      .Add(CMD_PARSE_ERRORS, False)
    End With

    ' Chỉ cho phép chọn trên ListBox nếu cả ba file đều tồn tại
    Me.lstCommands.Enabled = (Not blnFileNotFound)

  End Sub
```

```
Private Sub lstCommands_ItemCheck(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ItemCheckEventArgs) Handles lstCommands.ItemCheck

    If e.NewValue = CheckState.Checked Then

        Me.txtXMLDisplay.Text = String.Empty
        Me.txtXMLEdits.Text = String.Empty

        ' StringWriter được sử dụng tạo chuỗi kết quả để hiển thị
        msw = New StringWriter()

        Select Case Me.lstCommands.Items(e.Index).ToString()
            Case CMD_LOAD_XML_FILE
                Me.LoadAndDisplayXML()
            Case CMD_LOAD_XML_STRING
                Me.LoadXMLFromString()
            Case CMD_TEST_FOR_CHILD_NODES
                Me.HasAnyChildren()
            Case CMD_ITERATE_ALL_NODES
                Me.IterateThroughNodes()
            Case CMD_DETERMINE_NODE_TYPE
                Me.DetermineNodeType()
            Case CMD_LIST_ALL_ELEMENT_NODES
                Me.ListElementNodes()
            Case CMD_LIST_ELEMENTS_BY_TAG
                Me.ListSpecificTag()
            Case CMD_SELECT_NODES
                Me.CreateNodeList()
            Case CMD_SELECT_NODE
                Me.ReferToNode()
            Case CMD_NAVIGATE_RELATED_NODES
                Me.NavigateToNodes()
            Case CMD_RETRIEVE_ATTRIBUTES
                Me.RetrieveAttributes()
            Case CMD_CREATE_XML
                Me.CreateXML()
            Case CMD_ADD_OR_DELETE_ELEMENTS
                Me.AddDeleteNode()
            Case CMD_ADD_OR_DELETE_ATTRIBUTES
                Me.AddDeleteAttribute()
```

```
            Case CMD_MODIFY_ELEMENT
              Me.ModifyElementValue()
            Case CMD_MODIFY_ATTRIBUTE
              ModifyAttributeValue()
            Case CMD_VALID_XML
              Me.CheckIfXMLValid()
            Case CMD_PARSE_ERRORS
              Me.GetXMLErrorDetail()
          End Select

          ' Cập nhật thanh trạng thái, hiển thị lệnh cuối cùng được chọn
          Me.sbInfo.Text = "Last run command: " & _
                                Me.lstCommands.Items(e.Index).ToString()
          ' Giải phóng StringWriter
          msw.Flush()
          msw.Close()
        End If
    End Sub

    Private Sub mnuResetCheckBoxes_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs)
        ' Phục vụ cho mục đích reset các checkbox (xóa đánh dấu tất cả checkbox)
        If Me.lstCommands.CheckedItems.Count > 0 Then
          Me.txtXMLDisplay.Text = String.Empty
          Me.txtXMLEdits.Text = String.Empty

          Dim i As Integer
          Dim enumRef As IEnumerator
          enumRef = Me.lstCommands.CheckedIndices.GetEnumerator()

          While Not (enumRef.MoveNext() = False)
            i = CInt(enumRef.Current)
            Me.lstCommands.SetItemChecked(i, False)
          End While

        End If
    End Sub
```

```
Private Sub AddDeleteNode()
    ' Thủ tục này xóa một node và thêm bốn node mới
    Dim xDoc As New XmlDocument()
    xDoc.Load(mstrModifyFile)

    Me.txtXMLDisplay.Text = xDoc.OuterXml

    Dim xNode As XmlNode
    Dim xElmntFamily As XmlElement
    ' Tìm một node cụ thể
    xNode = xDoc.SelectSingleNode("//Family")

    If Not (xNode Is Nothing) Then
      If TypeOf xNode Is XmlElement Then
        xElmntFamily = CType(xNode, XmlElement)
      End If

      xElmntFamily.RemoveChild(xElmntFamily.SelectSingleNode("Father"))

      ' Thêm từng node một
      InsertTextNode(xDoc, xElmntFamily, "Person", "Gerald L. Smith")
      ' InsertTextNode thực hiện:
      'Set xNode = xDoc.createElement("father")
      'xNode.appendChild xDoc.createTextNode("Gerald L. Smith")
      'xElmntFamily.appendChild xNode
      InsertTextNode(xDoc, xElmntFamily, "Person", "Sara Ann Smith")
      InsertTextNode(xDoc, xElmntFamily, "Person", "Richard Andrew Smith")
      InsertTextNode(xDoc, xElmntFamily, "Person", "Emily Jean Smith")

      xDoc.Save(mstrModifyFile)

      Me.txtXMLEdits.Text = xDoc.OuterXml
    Else
      Me.txtXMLEdits.Text = String.Format("Family Node was not found." & _
                          " Please try the '{0}' option first.", CMD_CREATE_XML)
    End If
End Sub
```

```
Private Sub AddDeleteAttribute()
    ' Thủ tục này xóa hết các node con của node Family và sau đó cho
    ' chúng vào lại kèm với một số thuộc tính. Thủ tục này cũng cho biết cách
    ' xử lý các thuộc tính đã tồn tại.
    Dim xDoc As New XmlDocument()

    xDoc.Load(mstrModifyFile)

    Me.txtXMLDisplay.Text = xDoc.OuterXml

    Dim xNode As XmlNode
    Dim xElem As XmlElement

    Dim xElmntFamily As XmlElement

    ' Tìm một node cụ thể
    xNode = xDoc.SelectSingleNode("//Family")

    If Not (xNode Is Nothing) Then
      If TypeOf xNode Is XmlElement Then
        xElmntFamily = CType(xNode, XmlElement)
      End If

      ' Xóa tất cả các node đã tạo
      For Each xNode In xElmntFamily
        xElmntFamily.RemoveChild(xNode)
      Next

      ' Thêm các node con
      xElem = InsertTextNode(xDoc, xElmntFamily, "Person", "Gerald L. Smith")
      xElem.SetAttribute("type", "parent")
      xElem.SetAttribute("age", "70")

      xElem = InsertTextNode(xDoc, xElmntFamily, "Person", "Sara Ann Smith")
      xElem.SetAttribute("type", "mother")

      xElem = InsertTextNode(xDoc, xElmntFamily, "Person", _
                                                  "Richard Andrew Smith")
      xElem.SetAttribute("type", "son")
```

```
        xElem = InsertTextNode(xDoc, xElmntFamily, "Person", "Emily Jean Smith")
        xElem.SetAttribute("type", "daughter")

        ' Xử lý các thuộc tính.
        xNode = xDoc.SelectSingleNode("//Person[@type='parent']")
        If Not (xNode Is Nothing) Then
            ' Bỏ thuộc tính "age", và đổi giá trị của thuộc tính "type" thành "father"
          xElem = CType(xNode, XmlElement)
          xElem.Attributes.RemoveNamedItem("age")
          xElem.SetAttribute("type", "father")
        End If

        xDoc.Save(mstrModifyFile)

        Me.txtXMLEdits.Text = xDoc.OuterXml
    Else
        Me.txtXMLEdits.Text = String.Format("Family Node was not found. " & _
                              "Please try the '{0}' option first.", CMD_CREATE_XML)
    End If
End Sub

Private Sub CheckIfXMLValid()
    Dim xDoc As New XmlDocument()

    Try
        ' Nếu file XML (hoặc chuỗi chứa nội dung XML) không đúng,
        ' một exception có kiểu XmlException sẽ được phát sinh
        xDoc.Load(mstrBadFile)

        ' Ta sẽ không thấy dòng này nếu sử dụng file không đúng định dạng
        Me.txtXMLDisplay.Text = "Valid XML File: " & mstrBadFile

    Catch exp As XmlException
        Me.txtXMLDisplay.Text = "Invalid XML File: " & mstrBadFile
    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
    End Try
End Sub
```

```
Private Sub CreateNodeList()
    ' Khởi tạo XML Document
    Me.LoadXMLFromFile()

    Dim xElem As XmlElement
    Dim strTag As String

    strTag = InputBox("Enter an XPath expression to find:", _
                "Enter Search Expression", DEF_XPATH_EXP_FIND_NODES)
    If strTag.Length > 0 Then
        ' Tìm một nhóm các node dựa trên biểu thức XPath đã có
        mxNodeList = mxDoc.SelectNodes(strTag)
        With msw
            .WriteLine("All text elements matching '{0}':", strTag)
            .WriteLine(mstrLine)

            If Not mxNodeList Is Nothing Then
                For Each xElem In mxNodeList
                    .WriteLine("Name: " & xElem.Name)
                    .WriteLine("InnerText: " & xElem.InnerText)
                    .WriteLine("InnerXml: " & xElem.InnerXml)
                Next xElem
            End If

            Me.txtXMLDisplay.Text = .ToString()
        End With
    End If
End Sub

Private Sub CreateXML()
    ' Thủ tục này chỉ cách xây dựng một file XML bằng code
    Dim xDoc As New XmlDocument()

    Dim xPI As XmlProcessingInstruction
    Dim xComment As XmlComment
    Dim xElmntRoot As XmlElement
    Dim xElmntFamily As XmlElement

    xPI = xDoc.CreateProcessingInstruction("xml", "version='1.0'")
```

```
    xDoc.AppendChild(xPI)

    xComment = xDoc.CreateComment("Family Information")
    xDoc.AppendChild(xComment)

    xElmntRoot = xDoc.CreateElement("xml")
    xDoc.AppendChild(xElmntRoot)

     ' Thay vì tạo các node mới cụ thể, sử dụng AppendChild sẽ trả về một tham
     ' chiếu đến node mới vừa thêm vào
    xElmntFamily = _
            CType(xElmntRoot.AppendChild(xDoc.CreateElement("Family")), _
                                                XmlElement)
    Call xElmntFamily.AppendChild(xDoc.CreateElement("Father"))

     ' Nếu file đã tồn tại, phương thức Save sẽ ghi chồng lên, do đó ta có thể
     ' xác nhận như sau:

     'If File.Exists(mstrModifyFile) Then
     'If MessageBox.Show(String.Format("Do you want to overwrite the file {0}?", _
     '                    mstrModifyFile), Me.Text, MessageBoxButtons.YesNo, _
     '                        MessageBoxIcon.Question) = DialogResult.Yes Then
     '   xDoc.Save(mstrModifyFile)
     'End If
     'End If

    xDoc.Save(mstrModifyFile)

    Me.txtXMLDisplay.Text = xDoc.OuterXml
End Sub

Private Sub DetermineNodeType()
     ' Khởi tạo XML Document
    Me.LoadXMLFromFile()
     ' Sử dụng một hàm đệ quy để ghé thăm tất cả các node
    TraverseTreeType(msw, mxDoc, 0)

    Me.txtXMLDisplay.Text = msw.ToString
End Sub
```

```
Private Sub GetXMLErrorDetail()
    Dim xDoc As New XmlDocument()

    Try
        ' Nếu file XML có lỗi thì sẽ phát sinh một Exception
        xDoc.Load(mstrBadFile)

        Me.txtXMLDisplay.Text = "Valid XML File: " & mstrBadFile

    Catch exp As XmlException
        With msw
            .WriteLine(exp.Message)
            Me.txtXMLDisplay.Text = .ToString
        End With
    Catch exp As Exception
        MessageBox.Show(exp.Message, exp.Source, MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
    End Try
End Sub

Private Sub HasAnyChildren()
    Me.LoadXMLFromFile()
    Dim tab2x As String = vbTab & vbTab

    ' Nếu Docement có các node con, duyệt qua tất cả chúng
    If mxDoc.HasChildNodes Then
        For Each mxNode In mxDoc.ChildNodes
            With msw
                .WriteLine(mstrLine)
                .WriteLine("Name: {0}{1}", tab2x, mxNode.Name)
                .WriteLine("Type: {0}{1}", tab2x, mxNode.NodeType)
                .WriteLine("Type (String): {0}{1}", vbTab, mxNode.NodeType.ToString())
                .WriteLine("Value: {0}{1}", tab2x, mxNode.Value)
                .WriteLine("Outer XML: {0}{1}", vbTab, mxNode.OuterXml)
            End With
        Next mxNode
        Me.txtXMLDisplay.Text = msw.ToString()
    End If
End Sub
```

```
Private Function InsertTextNode(ByVal xDoc As XmlDocument, ByVal xNode As
XmlNode, ByVal strTag As String, ByVal strText As String) As XmlElement
    ' Thêm một text node làm node con của xNode.
    Dim xNodeTemp As XmlNode
    xNodeTemp = xDoc.CreateElement(strTag)
    xNodeTemp.AppendChild(xDoc.CreateTextNode(strText))
    xNode.AppendChild(xNodeTemp)
    Return CType(xNodeTemp, XmlElement)
End Function

Private Sub IterateThroughNodes()
    Me.LoadXMLFromFile()
    ' Sử dụng hàm đệ quy để ghé thăm tất cả các node
    TraverseTree(msw, mxDoc, 0)
    Me.txtXMLDisplay.Text = msw.ToString
End Sub

Private Sub ListElementNodes()
    Me.LoadXMLFromFile()
    ' Tìm các elements theo tên thẻ (tag)
    ' Ví dụ này tìm tất cả (*) các element
    mxNodeList = mxDoc.GetElementsByTagName("*")

    With msw
        .WriteLine("Elements matching '*':")
        .WriteLine(mstrLine)
        For Each mxNode In mxNodeList
            .WriteLine(mxNode.Name)
        Next mxNode

        Me.txtXMLDisplay.Text = .ToString()
    End With
End Sub

Private Sub ListSpecificTag()
    Me.LoadXMLFromFile()

    Dim strTag As String
    Dim xnt As XmlNode
```

```
    ' Hàm này cho phép tìm các element cụ thể theo tên tag
    strTag = InputBox("Enter a Tag to find:", "Enter Tag", "")
    If Len(strTag) > 0 Then
      mxNodeList = mxDoc.GetElementsByTagName(strTag)

      With msw
        .WriteLine("All text elements matching '{0}':", strTag)
        .WriteLine(mstrLine)

        For Each mxNode In mxNodeList
          For Each xnt In mxNode.ChildNodes
            If xnt.NodeType = XmlNodeType.Text Then
              .WriteLine(mxNode.Name & ": " & xnt.Value)
            End If
          Next xnt
        Next mxNode

        Me.txtXMLDisplay.Text = .ToString()
      End With
    End If
  End Sub

  Private Sub LoadXMLFromFile()
    ' Gọi một phiên bản phức tạp hơn
    Me.LoadXMLFromFile(False)
  End Sub

  Private Sub LoadXMLFromFile(ByVal ForceReload As Boolean)
    If ForceReload OrElse (mxDoc Is Nothing) Then
      mxDoc = New XmlDocument()
      mxDoc.Load(mstrSimpleFile)
    End If
  End Sub

  Private Sub LoadAndDisplayXML()
    Me.LoadXMLFromFile()

    Me.txtXMLDisplay.Text = mxDoc.OuterXml()
  End Sub
```

```
Private Sub LoadXMLFromString()
    ' Tạo một chuỗi tùy ý có các tag XML

    ' Sử dụng class StringWriter
    With msw
        .WriteLine("<xml>")
        .WriteLine("<Family>")
        .WriteLine("    <Person type='father'>Gerald L. Smith</Person>")
        .WriteLine("    <Person type='mother'>Sara Ann Smith</Person>")
        .WriteLine("    <Person type='child' gender='male'>Richard Andrew
Smith</Person>")
        .WriteLine("    <Person type='child' gender='female'>Emily Jean
Smith</Person>")
        .WriteLine("</Family>")
        .WriteLine("</xml>")
    End With

    ' Tạo một instance XML Document cục bộ
    Dim xDoc As New XmlDocument()
    ' Báo cho trình phân tích XML để lại các CRLF trong document
    xDoc.PreserveWhitespace = True
    xDoc.LoadXml(msw.ToString())

    Me.txtXMLDisplay.Text = xDoc.OuterXml()
End Sub

Private Sub ModifyElementValue()
    ' Cho thấy cách thay đổi một giá trị của một element
    Dim xDoc As New XmlDocument()

    xDoc.Load(mstrModifyFile)

    Me.txtXMLDisplay.Text = xDoc.OuterXml

    Dim xNode As XmlNode
    Dim xElem As XmlElement

    Dim xElmntFamily As XmlElement
    xNode = xDoc.SelectSingleNode("//Person")
```

```
    If Not (xNode Is Nothing) Then
        xElem = CType(xNode, XmlElement)
        ' Đổi "Gerald L. Smith" thành "Jerry Smith"
        xElem.InnerText = "Jerry Smith"

        Me.txtXMLEdits.Text = xDoc.OuterXml
    Else
        Me.txtXMLEdits.Text = String.Format("Family Node was not found. " & _
                              "Please try the '{0}' option first.", CMD_CREATE_XML)
    End If
End Sub

Private Sub ModifyAttributeValue()
    ' Thay đổi các giá trị của cá thuộc tính
    Dim xDoc As New XmlDocument()

    xDoc.Load(mstrModifyFile)

    Me.txtXMLDisplay.Text = xDoc.OuterXml

    Dim xNode As XmlNode
    Dim xElem As XmlElement

    For Each xNode In xDoc.SelectNodes("//Person")
        xElem = CType(xNode, XmlElement)

        Select Case xElem.GetAttribute("type")
            Case "father"
                xElem.SetAttribute("type", "parent")
                xElem.SetAttribute("gender", "male")

            Case "mother"
                xElem.SetAttribute("type", "parent")
                xElem.SetAttribute("gender", "female")

            Case "son"
                xElem.SetAttribute("type", "child")
                xElem.SetAttribute("gender", "male")
```

```
            Case "daughter"
               xElem.SetAttribute("type", "child")
               xElem.SetAttribute("gender", "female")
            End Select
         Next

         Me.txtXMLEdits.Text = xDoc.OuterXml
      End Sub

      Private Sub NavigateToNodes()
         Me.LoadXMLFromFile()

         Dim xElem As XmlElement
         Dim xnodTemp As XmlNode

         ' In ra tên của mỗi phần tử mới
         mxNodeList = mxDoc.SelectNodes("//New")

         If Not (mxNodeList Is Nothing) Then
            With msw
               .WriteLine("All New Items:")
               .WriteLine(mstrLine)
               For Each mxNode In mxNodeList
                  xnodTemp = mxNode.ParentNode.SelectSingleNode("Name")
                  If Not (xnodTemp Is Nothing) Then
                     If TypeOf xnodTemp Is XmlElement Then
                        xElem = CType(xnodTemp, XmlElement)
                        .WriteLine(xElem.InnerText)
                     End If
                  End If
               Next mxNode
               Me.txtXMLDisplay.Text = .ToString()
            End With
         End If
      End Sub

      Private Sub ReferToNode()
         Me.LoadXMLFromFile()
         Dim xElem As XmlElement
         Dim strTag As String
```

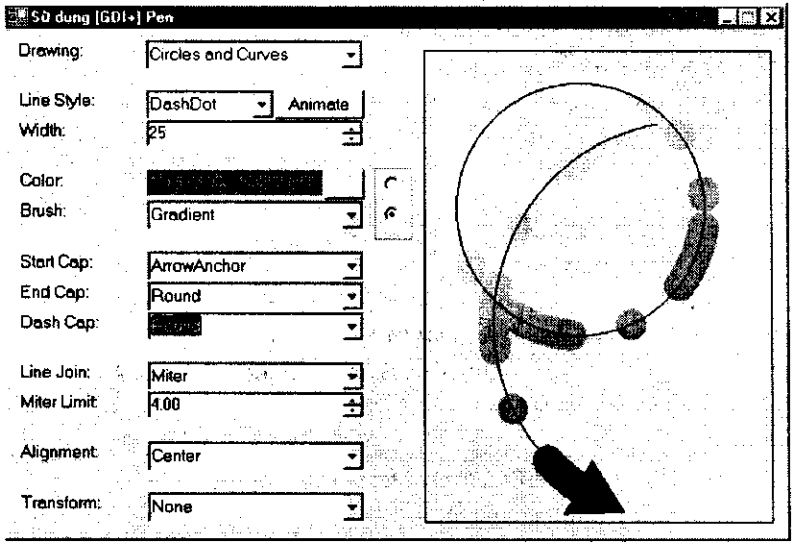
```
    strTag = InputBox("Enter an XPath expression to find:", _
                      "Enter Search Expression", DEF_XPATH_EXP_FIND_NODE)
    If strTag.Length > 0 Then
      With msw
        .WriteLine("All elements matching '{0}':", strTag)
        .WriteLine(mstrLine)

        mxNode = mxDoc.SelectSingleNode(strTag)
        If Not (mxNode Is Nothing) Then
          If TypeOf mxNode Is XmlElement Then
            xElem = CType(mxNode, XmlElement)
            .WriteLine(xElem.InnerText)
          End If
        End If

        Me.txtXMLDisplay.Text = .ToString()
      End With
    End If
  End Sub

  Private Sub RetrieveAttributes()
    Me.LoadXMLFromFile()

    Dim xAttr As XmlAttribute
    Dim xTmpNode As XmlNode

    With msw
      mxNodeList = mxDoc.SelectNodes("//Item")
      If Not (mxNodeList Is Nothing) Then
        .WriteLine("All Item Attributes:")
        .WriteLine(mstrLine)

        For Each mxNode In mxNodeList
          .WriteLine("{0} ({1})", mxNode.Name, mxNode.InnerText)
          For Each xAttr In mxNode.Attributes
            .WriteLine("   {0}: {1}", xAttr.Name, xAttr.Value)
          Next
        Next
      End If
```

```
        .WriteLine()
        .WriteLine()

        mxNodeList = mxDoc.SelectNodes("//Department")

        If Not (mxNodeList Is Nothing) Then
          .WriteLine("Departments:")
          .WriteLine(mstrLine)

          For Each mxNode In mxNodeList
            xTmpNode = mxNode.Attributes.GetNamedItem("Name")
            If Not (xTmpNode Is Nothing) Then
              xAttr = CType(xTmpNode, XmlAttribute)
              .WriteLine(xAttr.Value)
            End If
          Next
        End If

        Me.txtXMLDisplay.Text = .ToString()
    End With
End Sub

Private Sub TraverseTree(ByVal sw As StringWriter, ByVal xNode As XmlNode,
ByVal intLevel As Integer)
    Dim xNodeLoop As XmlNode

    ' In ra tên node của node hiện hành
    Dim s As New String(System.Convert.ToChar(vbTab), intLevel)
    sw.WriteLine(s & xNode.Name)
    ' Nếu node hiện hành có các node con, gọi lại thủ tục này và truyền node con vào.
    If xNode.HasChildNodes Then
      For Each xNodeLoop In xNode.ChildNodes
        Call TraverseTree(sw, xNodeLoop, intLevel + 1)
      Next xNodeLoop
    End If

End Sub
```

```
Private Sub TraverseTreeType(ByVal sw As StringWriter, ByVal xNode As
XmlNode, ByVal intLevel As Integer)
    Dim xNodeLoop As XmlNode

     ' In tên node của node hiện hành
    Dim s As New String(System.Convert.ToChar(vbTab), intLevel)
    Dim strValues() As String = {s, xNode.Name, xNode.NodeType.ToString()}
    sw.WriteLine("{0}{1} ({2})", strValues)

    If xNode.HasChildNodes Then
      For Each xNodeLoop In xNode.ChildNodes
        Call TraverseTreeType(sw, xNodeLoop, intLevel + 1)
      Next xNodeLoop
    End If
  End Sub
End Class
```

75. Sử dụng [GDI+] Pen

Chương trình này cho biết nhiều tính năng chính của đối tượng Pen khi sử dụng GDI+.



Các chức năng chính:

- Người sử dụng có thể chọn một số các tham số khác nhau để điều khiển cách vẽ của Pen.

- Sau mỗi lần chọn, một nhóm các đường thẳng được vẽ lại để thấy tác dụng.

Một số tính năng cao cấp gồm:

- Thiết lập đối tượng Pen trên đối tượng Brush để có thể vẽ các đường thẳng cao cấp.
- Làm hoạt hình một đường đứt nét.
- Thay đổi độ rộng của Pen để trông giống bút viết.

Project: **Sung dung [GDI+] Pen**

frmMain.vb

```
Option Strict On
Imports System.Drawing.Drawing2D

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Khai báo các biến cần thiết
    Dim m_Pen As New Pen(Color.Black)
    Dim m_penColor As Color = Color.BurlyWood
    Dim m_penBrush As Brush = New SolidBrush(Color.Black)
    Dim m_BlackThinPen As New Pen(Color.Black)

    Dim graphic As Graphics

' Code phát sinh form được bỏ qua

    ' Dùng để bật hoặc tắt Timer thực hiện hoạt hình
    Private Sub btnCycle_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCycle.Click
        timerCycle.Interval = 333
        timerCycle.Enabled = Not timerCycle.Enabled
        If timerCycle.Enabled Then
            btnCycle.Text = "Stop!"
        Else
            btnCycle.Text = "Animate"
        End If
    End Sub
```

```
' Thủ tục này chọn màu dựa trên màu do người sử dụng chọn từ ColorDialog
Private Sub btnSetColor_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSetColor.Click
    Dim cdlg As New ColorDialog()

    If cdlg.ShowDialog() = DialogResult.OK Then
        m_penColor = cdlg.Color
        txtColor.Text = cdlg.Color.ToString()
        txtColor.BackColor = cdlg.Color
    End If
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    Dim i As Integer

    ' Thiết lập các ComboBox trên form
    ' StartCap
    comboStartCap.Items.Add(LineCap.DiamondAnchor)
    comboStartCap.Items.Add(LineCap.ArrowAnchor)
    comboStartCap.Items.Add(LineCap.DiamondAnchor)
    comboStartCap.Items.Add(LineCap.Flat)
    comboStartCap.Items.Add(LineCap.Round)
    comboStartCap.Items.Add(LineCap.RoundAnchor)

    ' EndCap
    comboEndCap.Items.Add(LineCap.DiamondAnchor)
    comboEndCap.Items.Add(LineCap.ArrowAnchor)
    comboEndCap.Items.Add(LineCap.DiamondAnchor)
    comboEndCap.Items.Add(LineCap.Flat)
    comboEndCap.Items.Add(LineCap.Round)
    comboEndCap.Items.Add(LineCap.RoundAnchor)

    ' Dash Cap
    comboDashCap.Items.Add(DashCap.Flat)
    comboDashCap.Items.Add(DashCap.Round)
    comboDashCap.Items.Add(DashCap.Triangle)
```

```
        ' Line Join
        comboLineJoin.Items.Add(LineJoin.Bevel)
        comboLineJoin.Items.Add(LineJoin.Miter)
        comboLineJoin.Items.Add(LineJoin.MiterClipped)
        comboLineJoin.Items.Add(LineJoin.Round)


        ' Line Style
        comboLineStyle.Items.Add(DashStyle.Solid)
        comboLineStyle.Items.Add(DashStyle.Dash)
        comboLineStyle.Items.Add(DashStyle.DashDot)
        comboLineStyle.Items.Add(DashStyle.DashDotDot)
        comboLineStyle.Items.Add(DashStyle.Dot)
        comboLineStyle.Items.Add(DashStyle.Custom)

        ' Alignment
        comboAlignment.Items.Add(PenAlignment.Center)
        comboAlignment.Items.Add(PenAlignment.Inset)
        comboAlignment.Items.Add(PenAlignment.Left)
        comboAlignment.Items.Add(PenAlignment.Outset)
        comboAlignment.Items.Add(PenAlignment.Right)

    End Sub

    Private Sub radioColor_CheckedChanged(ByVal sender As System.Object, ByVal
    e As System.EventArgs) Handles radioColor.CheckedChanged
        txtColor.Enabled = radioColor.Checked
        btnSetColor.Enabled = radioColor.Checked
        comboBrush.Enabled = radioBrush.Checked
    End Sub

    ' Thủ tục này thu thập tất cả thông tin do người sử dụng chọn và sử dụng các
    ' thông tin này để tạo một đối tượng Pen.
    Private Sub RedrawPicture(ByVal sender As System.Object, ByVal e As
    System.EventArgs) _
        Handles MyBase.Activated, comboShape.SelectedIndexChanged, _
        updownWidth.ValueChanged, txtColor.TextChanged, _
        comboAlignment.SelectedIndexChanged, _
        comboStartCap.SelectedIndexChanged, _
        comboEndCap.SelectedIndexChanged, _
```

```
    comboDashCap.SelectedIndexChanged, _
    comboLineJoin.SelectedIndexChanged, _
    comboLineStyle.SelectedIndexChanged, _
    updownMiterLimit.ValueChanged, _
    comboTransform.SelectedIndexChanged, _
    comboBrush.SelectedIndexChanged

    ' Reset PictureBox
    pbLines.CreateGraphics().Clear(pbLines.BackColor)
    pbLines.Refresh()
    ' Bỏ tất cả các biến đổi trên Pen
    m_Pen.ResetTransform()
    ' Chọn DashPartern để sử dụng nếu chọn Line Style là Custom
    m_Pen.DashPattern = New Single() {0.5, 0.25, 0.75, 1.5}

    ' Một đối tượng Pen thường kết với một đối tượng Color hoặc Brush
    If radioColor.Checked Then
        m_Pen.Color = m_penColor
    Else
        Select Case comboBrush.Text
            Case "Solid"
                ' Tương tự như gán Color cho Pen
                m_penBrush = New SolidBrush(m_penColor)
            Case "Hatch"
                ' Định nghĩa một HatchBrush sẽ sử dụng, ta sử dụng Plaid
                m_penBrush = New HatchBrush(HatchStyle.Plaid, m_penColor)
            Case "Texture"
                ' Sử dụng một hình ảnh làm Brush
                m_penBrush = New TextureBrush( _
                    New Bitmap("..\WaterLilies.jpg"), WrapMode.Tile)
            Case "Gradient"
                ' Có thể sử dụng các Gradient Brush khác ở đây
                m_penBrush = New LinearGradientBrush( _
                    New Point(0, 0), _
                    New Point(pbLines.Width, pbLines.Height), _
                    Color.AliceBlue, Color.DarkBlue)
        End Select
        m_Pen.Brush = m_penBrush
    End If
```

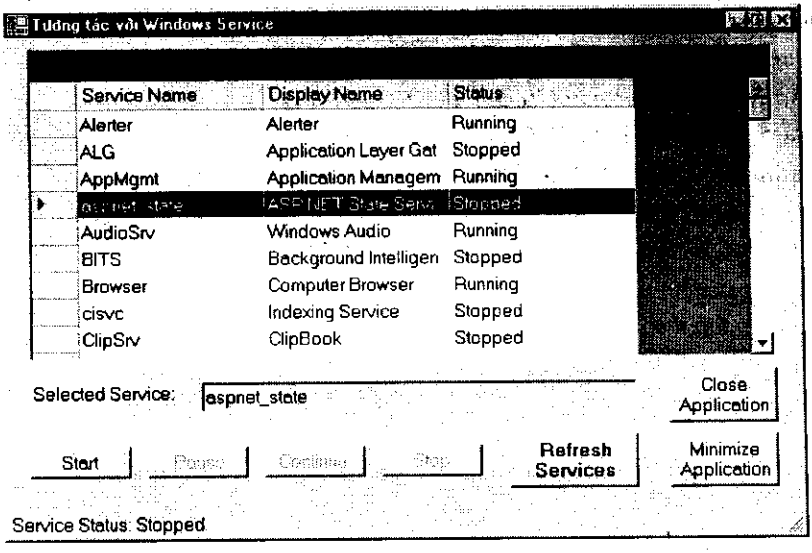
```
    ' Gán các giá trị do người sử dụng chọn vào đối tượng Pen
    ' Bề rộng của Pen, tính bằng pixel
    m_Pen.Width = updownWidth.Value

    ' DashStyle quyết định kiểu của đường thẳng
    m_Pen.DashStyle = CType(comboLineStyle.SelectedItem, DashStyle)
    ' MiterLimit quyết định khi nào các cạnh ở góc được cắt. Mặc nhiên là 10.0
    m_Pen.MiterLimit = updownMiterLimit.Value
    ' StartCap quyết định kiểu của đầu bắt đầu của đường thẳng
    m_Pen.StartCap = CType(comboStartCap.SelectedItem, LineCap)
    ' EndCap quyết định kiểu của đầu kết thúc của đường thẳng
    m_Pen.EndCap = CType(comboEndCap.SelectedItem, LineCap)
    ' DashCap quyết định kiểu của hai đầu của đường gạch trên đường thẳng
    m_Pen.DashCap = CType(comboDashCap.SelectedItem, DashCap)
    ' LineJoin quyết định kiểu giao của hai đường thẳng
    m_Pen.LineJoin = CType(comboLineJoin.SelectedItem, LineJoin)

    m_Pen.Alignment = CType(comboAlignment.SelectedItem, PenAlignment)

' Transform được sử dụng cho một số tính năng cao cấp. Ví dụ, tạo Pen giống bút vẽ
    Select Case comboTransform.Text
      Case "None"
        m_Pen.ResetTransform()
      Case "Scale"
          ' ScaleTransform  thay đổi bề rộng và chiều cao của Pen
        m_Pen.ScaleTransform(0.5, 2)
      Case "Rotate"
          ' RotateTransform xoay Brush theo một góc tính bằng độ
        m_Pen.RotateTransform(45)
      Case "Translate"
          ' TranslateTransform dịch Brush
        m_Pen.TranslateTransform(2, 4)
    End Select

    graphic = pbLines.CreateGraphics()

    If Me.comboShape.Text = "Lines" Then
        ' Vẽ ba đường cơ bản
```

```
        ' Vẽ theo các thông số được chọn
        graphic.DrawLine(m_Pen, 35, 35, pbLines.Width - 35, 35)
        graphic.DrawLine(m_Pen, 35, 80, 35, pbLines.Height - 35)
        graphic.DrawLine(m_Pen, 90, 90, pbLines.Width - 35, _
          pbLines.Height - 35)

        ' Vẽ với pen màu đen
        graphic.DrawLine(m_BlackThinPen, 35, 35, pbLines.Width - 35, 35)
        graphic.DrawLine(m_BlackThinPen, 35, 80, 35, pbLines.Height - 35)
        graphic.DrawLine(m_BlackThinPen, 90, 90, pbLines.Width - 35, _
            pbLines.Height - 35)

    ElseIf Me.comboShape.Text = "Intersecting Lines" Then
        ' Vẽ một đường kết hợp

        ' Vẽ một đường phức tạp sử dụng một mảng các Point.
        ' Với mảng này ta chỉ có hai đầu của đường vẽ.
        Dim ptArray(5) As PointF
        ptArray(0) = New PointF(35, 35)
        ptArray(1) = New PointF(70, pbLines.Height - 75)
        ptArray(2) = New PointF(100, 35)
        ptArray(3) = New PointF(pbLines.Width - 40, pbLines.Height \ 2)
        ptArray(4) = New PointF(pbLines.Width \ 2, pbLines.Height \ 2)
        ptArray(5) = New PointF(pbLines.Width - 25, 25)

        ' Vẽ đường theo thông số Pen được chọn
        graphic.DrawLines(m_Pen, ptArray)

        ' Vẽ đường sử dụng Pen màu đen
        graphic.DrawLines(m_BlackThinPen, ptArray)

    ElseIf Me.comboShape.Text = "Circles and Curves" Then
        ' Vẽ một vòng cung và một vòng tròn

        graphic.DrawEllipse(m_Pen, 25, 25, 200, 200)
        graphic.DrawArc(m_Pen, 55, 55, pbLines.Height - 55, _
          pbLines.Height - 55, 110, 150)

        graphic.DrawEllipse(m_BlackThinPen, 25, 25, 200, 200)
```

```
        graphic.DrawArc(m_BlackThinPen, 55, 55, pbLines.Height - 55, _
          pbLines.Height - 55, 110, 150)

      End If
    End Sub

    ' Làm hoạt hình cho các đường vẽ bằng cách tăng giá trị của DashOffset
    Private Sub timerCycle_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles timerCycle.Tick
      m_Pen.DashOffset = (m_Pen.DashOffset + 0.5F) Mod 30
      RedrawPicture(Me, New System.EventArgs())
    End Sub
End Class
```

76. Tương tác với Windows Service

Chương trình này cho thấy cách điều khiển một Windows Service đã được cài đặt.



Các chức năng chính:

- Chương trình này liệt kê tất cả và cho phép điều khiển bất kỳ Windows Service nào đã được cài đặt trên máy cục bộ. Bao gồm các chức năng Start, Stop, Pause và Continue. Ngoài ra, chương trình còn biểu diễn cách tạo một trình điều khiển Service giống như SQL Server Service Manager. Chức năng này sử dụng các control NotifyIcon và ContextMenu.

Project: **Tuong tac voi Windows Service**

frmMain.vb

```
Option Strict On
' Phải có tham chiếu đến "System.ServiceProcess.dll"
' Nhấn chuột phải vào References trong Solution Explorer để thêm tham chiếu
Imports System.ServiceProcess

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Biến để lưu Service được chọn
    Dim m_SelectedService As ServiceController
    ' Biến lưu tên của Service được chọn
    Dim m_SelectedServiceName As String = "No Service Selected"
    ' Bởi vì nút X của form được dùng để thu nhỏ form (Minimize), để tránh việc đóng
    ' từ nút X ta sử dụng biến cờ để thông báo cho ứng dụng biết là ta sử dụng
    ' Button "Close Application" để thoát ứng dụng.
    Dim m_isCloseButtonPushed As Boolean = False

    Friend WithEvents btnStop As System.Windows.Forms.Button
    Friend WithEvents btnContinue As System.Windows.Forms.Button
    Friend WithEvents btnPause As System.Windows.Forms.Button
    Friend WithEvents sbrServiceStatus As System.Windows.Forms.StatusBar
    Friend WithEvents btnStart As System.Windows.Forms.Button
    Friend WithEvents grdServices As System.Windows.Forms.DataGrid
    Friend WithEvents txtSelectedService As System.Windows.Forms.TextBox
    Friend WithEvents tmrCheckStatus As System.Windows.Forms.Timer
    Friend WithEvents btnRefreshServices As System.Windows.Forms.Button
    Friend WithEvents cmnuServiceStatus As System.Windows.Forms.ContextMenu
    Friend WithEvents nicoSvcMgrApp As System.Windows.Forms.NotifyIcon
    Friend WithEvents miServiceName As System.Windows.Forms.MenuItem
    Friend WithEvents miLineBreak1 As System.Windows.Forms.MenuItem
    Friend WithEvents miStart As System.Windows.Forms.MenuItem
    Friend WithEvents miPause As System.Windows.Forms.MenuItem
    Friend WithEvents miContinue As System.Windows.Forms.MenuItem
    Friend WithEvents miStop As System.Windows.Forms.MenuItem
    Friend WithEvents miLineBreak2 As System.Windows.Forms.MenuItem
    Friend WithEvents miSelectService As System.Windows.Forms.MenuItem
```

```
    Friend WithEvents btnCloseApp As System.Windows.Forms.Button
    Friend WithEvents btnMinimizeApp As System.Windows.Forms.Button
    Friend WithEvents lblSelectedService As System.Windows.Forms.Label
    Friend WithEvents mnuExitApp As System.Windows.Forms.MenuItem

' Code phát sinh form được bỏ qua
    ' Thủ tục này dùng để thoát ứng dụng khi nhấn nút "Close Application"
    Private Sub btnCloseApp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCloseApp.Click
        m_isCloseButtonPushed = True
        Me.Close()
    End Sub

    ' Thủ tục này ra lệnh service tiếp tục hoạt động (Continue)
    ' Chú ý: xử lý cho cả Button và MenuItem
    Private Sub btnContinue_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnContinue.Click, miContinue.Click
        m_SelectedService.Continue()
        UpdateServiceStatus()
    End Sub
    ' Minimize cửa sổ ứng dụng
    Private Sub btnMinimizeApp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnMinimizeApp.Click
        Me.WindowState = FormWindowState.Minimized
    End Sub

    ' Ra lệnh cho service tạm dừng (Pause)
    Private Sub btnPause_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnPause.Click, miPause.Click
        m_SelectedService.Pause()
        UpdateServiceStatus()
    End Sub

    ' Ra lệnh cho service dừng hẳn (Stop)
    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStop.Click, miStop.Click
        m_SelectedService.Stop()
        UpdateServiceStatus()
    End Sub
```

```
' Ra lệnh cho service hoạt động (Start)
Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click, miStart.Click
    m_SelectedService.Start()
    UpdateServiceStatus()
End Sub

' Thủ tục này sử dụng để tìm các Service đã được cài đặt, và ép cho DataGrid
' cập nhật thông tin mới nhất.
Private Sub btnRefreshServices_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRefreshServices.Click
    grdServices.DataSource = GetServicesTable()
    grdServices.Refresh()
End Sub

' Thủ tục này bảo đảm là MenuItem đầu tiên là tên của Service được chọn
Private Sub cmnuServiceStatus_Popup(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles cmnuServiceStatus.Popup
    Me.cmnuServiceStatus.MenuItems(0).Text = Me.m_SelectedServiceName
End Sub

' Xử lý cho sự kiện Closing (khi nhấn nút X)
Private Sub frmMain_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    If Not m_isCloseButtonPushed Then
        Me.WindowState = FormWindowState.Minimized
        e.Cancel = True
    End If
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    grdServices.PreferredColumnWidth = grdServices.Width \ 4
    ' Lấy các Windows Service và hiển thị chúng trên DataGrid
    grdServices.DataSource = GetServicesTable()
    grdServices.Refresh()
    ' Cho timer hoạt động để nó cập nhật thôngtin theo thời gian
    tmrCheckStatus.Enabled = True
End Sub
```

```
' Xác định Service do người sử dụng chọn và gán nó vào biến Service được chọn
Private Sub grdServices_MouseDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.MouseEventArgs) Handles grdServices.MouseDown

    ' Tạo một biến HitTestInfo để xác định dòng được chọn
    Dim hit As System.Windows.Forms.DataGrid.HitTestInfo = _
      grdServices.HitTest(e.X, e.Y)
    ' Tìm service dựa trên dòng được chọn
    If hit.Row >= 0 Then
      Me.grdServices.Select(hit.Row)
      m_SelectedServiceName = Me.grdServices.Item(hit.Row, 0).ToString()
      RefreshSelectedService()
      UpdateServiceStatus()
    End If
End Sub

' Hiển thị cửa sổ chính  khi người sử dụng chọ Service trong ContextMenu
Private Sub miSelectService_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles miSelectService.Click
    Me.WindowState = FormWindowState.Normal
End Sub

' Hiển thị cửa sổ chính khi người sử dụng nhấp đúp vào NotifyIcon
Private Sub nicoSvcMgrApp_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles nicoSvcMgrApp.DoubleClick
    Me.WindowState = FormWindowState.Normal
End Sub

' Thủ tục này cập nhật thông tin sau mỗi 1/2 giây
Private Sub tmrCheckStatus_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles tmrCheckStatus.Tick
    RefreshSelectedService()
    UpdateServiceStatus()
End Sub

Private Sub mnuExitApp_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuExitApp.Click
    Application.Exit()
End Sub
```

```
' Hàm này tạo một DataTable để lưu tất cả các Windows Service trên máy tính
Private Function CreateDataTable() As DataTable
    Dim myTable As New DataTable("WindowsServiceTable")
    Dim myColumn As DataColumn
    Dim typeString As Type = "".GetType()

    ' Tạo các cột cần thiết
    myTable.Columns.Add("Service Name", typeString)
    myTable.Columns.Add("Display Name", typeString)
    myTable.Columns.Add("Status", typeString)

    Return myTable
End Function

' Hàm này trả về một DataTable đã chứa thông tin tất cả các Service
Private Function GetServicesTable() As DataTable
    Dim myTable As DataTable
    Dim myServicesArray() As ServiceController
    Dim myService As ServiceController
    Dim myRow As DataRow

    myTable = CreateDataTable()
    ' Lấy một mảng chứa tất cả Service đã cài đặt trên máy tính cục bộ
    myServicesArray = ServiceController.GetServices()

    ' Cho thông tin của các Windows Service vào DataTable
    For Each myService In myServicesArray
        myRow = myTable.NewRow()
        myRow.Item("Service Name") = myService.ServiceName
        myRow.Item("Display Name") = myService.DisplayName
        myRow.Item("Status") = myService.Status.ToString()

        myTable.Rows.Add(myRow)
    Next

    ' Trả về bảng đã có dữ liệu
    Return myTable
End Function
```

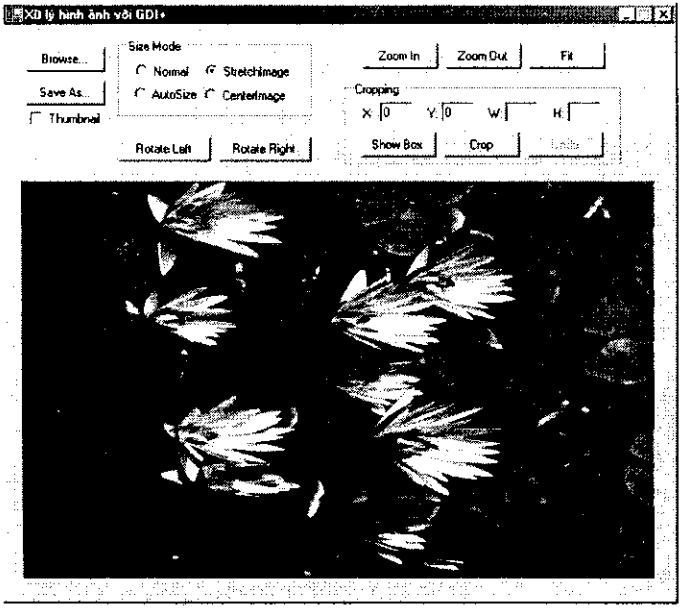
```
Private Sub RefreshSelectedService()
    Dim installedServices() As ServiceController
    Dim tmpService As ServiceController
    Dim isServiceInstalled As Boolean = False
    Dim i As Integer = 0

    Me.tmrCheckStatus.Enabled = False

    ' Lấy danh sách của các service đang có
    installedServices = ServiceController.GetServices()

    For Each tmpService In installedServices
        If tmpService.ServiceName = m_SelectedServiceName Then
            isServiceInstalled = True
            m_SelectedService = tmpService
        End If
    Next tmpService
    Me.tmrCheckStatus.Enabled = True
End Sub

' Thủ tục này cập nhật lại trạng thái của các control, ngoại trừ DataGrid
Private Sub UpdateServiceStatus()
    Me.tmrCheckStatus.Enabled = False

    If Not (m_SelectedService Is Nothing) Then
        RefreshSelectedService()

        Select Case m_SelectedService.Status
            Case ServiceControllerStatus.Stopped
                Me.btnContinue.Enabled = False
                Me.btnPause.Enabled = False
                Me.btnStart.Enabled = True
                Me.btnStop.Enabled = False
                Me.miContinue.Enabled = False
                Me.miPause.Enabled = False
                Me.miStart.Enabled = True
                Me.miStop.Enabled = False
```

```
        Case ServiceControllerStatus.Running
            Me.btnContinue.Enabled = False
            If m_SelectedService.CanPauseAndContinue Then
                Me.btnPause.Enabled = True
            End If
            Me.btnStart.Enabled = False
            Me.btnStop.Enabled = True
            Me.miContinue.Enabled = False
            If m_SelectedService.CanPauseAndContinue Then
                Me.miPause.Enabled = True
            End If
            Me.miStart.Enabled = False
            Me.miStop.Enabled = True

        Case ServiceControllerStatus.Paused
            If m_SelectedService.CanPauseAndContinue Then
                Me.btnContinue.Enabled = True
            End If
            Me.btnPause.Enabled = False
            Me.btnStart.Enabled = False
            Me.btnStop.Enabled = True

            If m_SelectedService.CanPauseAndContinue Then
                Me.miContinue.Enabled = True
            End If
            Me.miPause.Enabled = False
            Me.miStart.Enabled = False
            Me.miStop.Enabled = True
        Case Else
            ' Trường hợp này xảy ra khi một thao tác đang thực hiện. Trong
            ' trường hợp này không cho người sử dụng làm gì cả
            Me.btnContinue.Enabled = False
            Me.btnPause.Enabled = False
            Me.btnStart.Enabled = False
            Me.btnStop.Enabled = False
            Me.miContinue.Enabled = False
            Me.miPause.Enabled = False
            Me.miStart.Enabled = False
            Me.miStop.Enabled = False
    End Select
```

```
            ' Xuất thông tin trạng thái ra Status Bar
            Me.sbrServiceStatus.Text = "Service Status: " + _
                m_SelectedService.Status.ToString()
        Else
            ' Service không chạy được
            Me.btnContinue.Enabled = False
            Me.btnPause.Enabled = False
            Me.btnStart.Enabled = False
            Me.btnStop.Enabled = False
            Me.miContinue.Enabled = False
            Me.miPause.Enabled = False
            Me.miStart.Enabled = False
            Me.miStop.Enabled = False
        End If
        ' Đưa tên của service được chọn vào TextBox
        Me.txtSelectedService.Text = m_SelectedServiceName
        Me.tmrCheckStatus.Enabled = True
    End Sub
End Class
```

77. Xử lý hình ảnh với GDI+

Chương trình này cho biết cách xử lý các hình ảnh bằng cách sử dụng GDI+.



Các chức năng chính:

- Chọn, mở và hiển thị hình ảnh vào một PictureBox.
- Gán thuộc tính SizeMode của PictureBox.
- Phóng to và thu nhỏ, và định kích thước hình ảnh vừa bằng PictureBox.
- Quay hình ảnh sang trái, sang phải một góc 90°.
- Cắt hình ảnh.
- Lưu hình ảnh trong PictureBox vào một file, cả kích thước bình thường và Thumbnail (hình ảnh nhỏ).

Project: **Xu ly hinh anh voi GDI+**

frmMain.vb

```
Option Strict On
Imports System.Drawing.Imaging

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Const PICTUREBOX_WIDTH As Int32 = 608
    Const PICTUREBOX_HEIGHT As Int32 = 380
    Const THUMBNAIL_MIN_SIZE As Int32 = 64
    Private IsFitForZoomIn As Boolean = False
    Private imgUndo As Image
    Private intVal As Int32

    ' Chọn hình ảnh và hiển thị trong PictureBox
    Private Sub btnBrowse_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnBrowse.Click
        With odlgImage
            .InitialDirectory = "C:\"
            .Filter = "All Image Formats (*.bmp;*.jpg;*.jpeg;*.gif;*.tif)|" & _
                "*.bmp;*.jpg;*.jpeg;*.gif;*.tif|Bitmaps (*.bmp)|*.bmp|" & _
                "GIFs (*.gif)|*.gif|JPEGs (*.jpg)|*.jpg;*.jpeg|TIFs (*.tif)|*.tif"
            .FilterIndex = 1
        End With
```

```
        ' Khi người sử dụng nhấn Open, hiển thị hình ảnh trong PictureBox và
        ' đường dẫn đầy đủ của hình ảnh.
        If odlgImage.ShowDialog() = DialogResult.OK Then
            With picImage
                ' Giữ nguyên kích thước của PictureBox
                .Width = PICTUREBOX_WIDTH
                .Height = PICTUREBOX_HEIGHT
                .Image = Image.FromFile(odlgImage.FileName)
                .SizeMode = PictureBoxSizeMode.Normal
            End With
            optNormal.Checked = True
        End If
    End Sub

    ' Xử lý nút Fit
    Private Sub btnFit_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnFit.Click
        grpSizeMode.Enabled = True

        With picImage
            .Width = PICTUREBOX_WIDTH
            .Height = PICTUREBOX_HEIGHT
            .SizeMode = PictureBoxSizeMode.StretchImage
        End With
        ' Cập nhật giao diện
        optStretchImage.Checked = True
        Fit()
    End Sub

    ' Xử lý nút Crop
    Private Sub btnCrop_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCrop.Click
        If IsValidCropValues() Then
            ' Khởi tạo một biến chứa một bản sao của bitmap trước khi thực hiện
            ' cắt để có thể undo.
            imgUndo = picImage.Image
            btnUndo.Enabled = True
```

```
        ' Tạo một vùng hình chữ nhật theo tọa độ do người sử dụng chọn
        Dim recSource As New Rectangle(CInt(txtXCoord.Text), _
                    CInt(txtYCoord.Text), CInt(txtWidth.Text), CInt(txtHeight.Text))

        ' Tạo một Bitmap mới để vẽ hình ảnh được cắt.

        Dim bmpCropped As New Bitmap(CInt(txtWidth.Text), CInt(txtHeight.Text))
        ' Lấy đối tượng Graphics từ Bitmap để vẽ
        Dim grBitmap As Graphics = Graphics.FromImage(bmpCropped)
        grBitmap.DrawImage(picImage.Image, 0, 0, recSource, GraphicsUnit.Pixel)
        ' Đưa hình ảnh đã được cắt vào PictureBox
        picImage.Image = bmpCropped
    End If
End Sub

' Xử lý nút Rotate Left
Private Sub btnRotateLeft_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRotateLeft.Click
    ' Quay 270 độ tương đương với quay -90 độ
    picImage.Image.RotateFlip(RotateFlipType.Rotate270FlipNone)
    picImage.Refresh()
End Sub

' Xử lý nút Rotate Right
Private Sub btnRotateRight_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRotateRight.Click
    picImage.Image.RotateFlip(RotateFlipType.Rotate90FlipNone)
    picImage.Refresh()
End Sub

' Xử lý nút Save, lưu hình ảnh hiển thị trong PictureBox vào một file
Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    With sdlgImage
        .InitialDirectory = "C:\"
        .FileName = "myimage"
        .Filter = "Bitmap (*.bmp)|*.bmp|GIF (*.gif)|*.gif|" & _
            "JPEG (*.jpg)|*.jpg;*.jpeg|TIF (*.tif)|*.tif"
        .FilterIndex = 3
    End With
```

```
If sdlgImage.ShowDialog() = DialogResult.OK Then
   Dim strMsg As String

   Try
      If chkThumbnail.Checked Then
         ' Lưu hình ảnh là một Thumbnail
         With picImage
            Dim ratio As Double = CalculateAspectRatioAndSetDimensions()
            Dim intThumbWidth As Int32 = THUMBNAIL_MIN_SIZE
            Dim intThumbHeight As Int32 = THUMBNAIL_MIN_SIZE

            Dim imgSource As Image = .Image
            Dim imgThumbnail As Image

            If .Image.Width > .Image.Height Then
               intThumbWidth = CInt(CDbl(intThumbWidth * ratio))
               imgThumbnail = _
                  imgSource.GetThumbnailImage(intThumbWidth, _
                  intThumbHeight, Nothing, Nothing)
            Else
               intThumbHeight = CInt(CDbl(intThumbHeight * ratio))
               imgThumbnail = _
                  imgSource.GetThumbnailImage(intThumbWidth, _
                  intThumbHeight, Nothing, Nothing)
            End If

            imgThumbnail.Save(sdlgImage.FileName, GetImageFormat())
         End With

         strMsg = "Image successfully saved as thumbnail to " & _
            sdlgImage.FileName
      Else

         ' Lưu hình ảnh vào file có tên nhập vào trên SaveFileDialog
         picImage.Image.Save(sdlgImage.FileName, GetImageFormat())
         strMsg = "Image successfully saved to " & sdlgImage.FileName
      End If
```

```
            MessageBox.Show("Image successfully saved to " & _
              sdlgImage.FileName, Me.Text, _
              MessageBoxButtons.OK, MessageBoxIcon.Information)
        Catch exp As Exception
            MessageBox.Show("The following error occurred while trying to " & _
              "save the image: " & exp.Message, Me.Text, _
               MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End If
End Sub

' Xử lý nút Show. Hiển thị hình chữ nhật sẽ cắt
Private Sub btnShowBox_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShowBox.Click
    If IsValidCropValues() Then
        ' Gọi Refresh để xóa hình chữ nhật cũ, nếu có
        picImage.Refresh()
        ' Hiển thị hình chữ nhật đỏ để thấy được ảnh sẽ được cắt
        Dim recCropBox As New Rectangle(CInt(txtXCoord.Text), _
          CInt(txtYCoord.Text), CInt(txtWidth.Text), CInt(txtHeight.Text))
        picImage.CreateGraphics.DrawRectangle(Pens.Red, recCropBox)
    End If
End Sub

' Xử lý nút Undo
Private Sub btnUndo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnUndo.Click
    picImage.Image = imgUndo
    picImage.Refresh()
    btnUndo.Enabled = False
End Sub

' Xử lý nút Zoom In
Private Sub btnZoomIn_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnZoomIn.Click
    If grpSizeMode.Enabled Then
        picImage.SizeMode = PictureBoxSizeMode.AutoSize
    End If

    grpSizeMode.Enabled = False
```

```
        btnFit.Enabled = True
        IsFitForZoomIn = True

        picImage.SizeMode = PictureBoxSizeMode.StretchImage

        Fit()

        ' Tăng hình ảnh lên 25%
        picImage.Width = CInt(picImage.Width * 1.25)
        picImage.Height = CInt(picImage.Height * 1.25)
    End Sub

    ' Xử lý nút Zoom Out
    Private Sub btnZoomOut_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnZoomOut.Click
        grpSizeMode.Enabled = False
        btnFit.Enabled = True

        Fit()
        picImage.SizeMode = PictureBoxSizeMode.StretchImage

        ' Giảm kích thước nhỏ hơn 25%
        picImage.Width = CInt(picImage.Width / 1.25)
        picImage.Height = CInt(picImage.Height / 1.25)
    End Sub

    ' Xử lý sự kiện CheckedChanged trên nhóm các RadioButton
    Private Sub SizeModeRadioButtons_CheckedChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    optAutoSize.CheckedChanged, optCenterImage.CheckedChanged,
    optNormal.CheckedChanged, optStretchImage.CheckedChanged
        Dim opt As RadioButton = CType(sender, RadioButton)
        ' Mỗi RadioButton lưu một giá trị của enum PictureBoxSizeMode trong
        ' thuộc tính Tag của nó.
        Dim sm As PictureBoxSizeMode = CType(opt.Tag, PictureBoxSizeMode)

        If opt.Checked Then
            picImage.SizeMode = sm
            If sm = PictureBoxSizeMode.AutoSize Then
                btnFit.Enabled = False
```

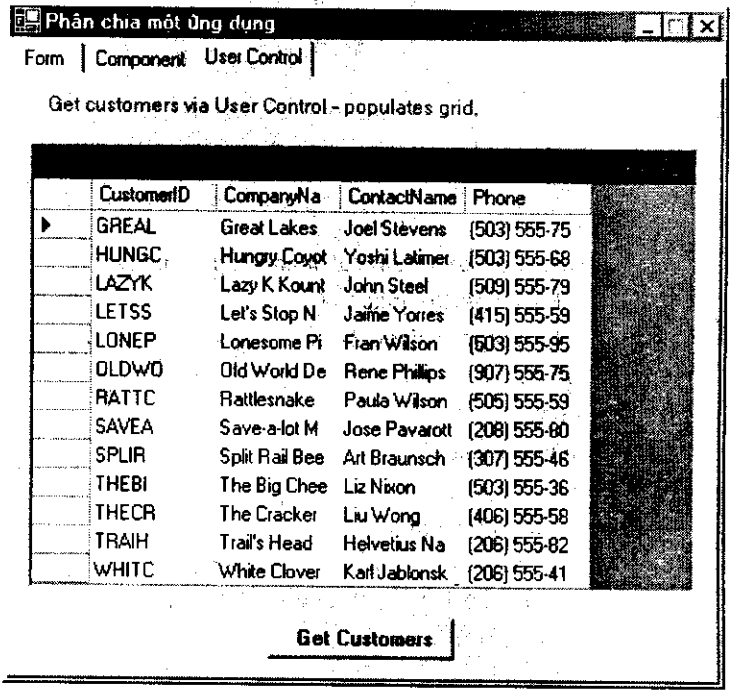
```
        Else
            btnFit.Enabled = True
            picImage.Width = PICTUREBOX_WIDTH
            picImage.Height = PICTUREBOX_HEIGHT
        End If
    End If
End Sub

' Thủ tục này làm cho hình ảnh vừa với PictureBox.
Private Sub Fit()
    If picImage.Image.Width < picImage.Width And _
        picImage.Image.Height < picImage.Height Then
        If Not IsFitForZoomIn Then
            picImage.SizeMode = PictureBoxSizeMode.CenterImage
        End If
    End If
    CalculateAspectRatioAndSetDimensions()
End Sub

' Tính toán và trả về tỉ lệ các cạnh của hình ảnh.
Private Function CalculateAspectRatioAndSetDimensions() As Double
    Dim ratio As Double
    If picImage.Image.Width > picImage.Image.Height Then
        ratio = picImage.Image.Width / _
            picImage.Image.Height
        picImage.Height = CInt(CDbl(picImage.Width) / ratio)
    Else
        ratio = picImage.Image.Height / _
            picImage.Image.Width
        picImage.Width = CInt(CDbl(picImage.Height) / ratio)
    End If
    Return ratio
End Function

' Gán định dạng tương ứng khi lưu hình ảnh bằng cách tham chiếu đến FilterIndex
' của hộp thoại SaveFileDialog.
Private Function GetImageFormat() As ImageFormat
    Select Case sdlgImage.FilterIndex
        Case 1
            Return ImageFormat.Bmp
```

```
        Case 2
            Return ImageFormat.Jpeg
        Case 3
            Return ImageFormat.Gif
        Case Else
            Return ImageFormat.Tiff
    End Select
End Function

' Kiểm tra dữ liệu nhập vào trong các TextBox Cropping
Private Function IsValidCropValues() As Boolean
    Dim objControl As Object
    For Each objControl In grpCropping.Controls
        If objControl.GetType.ToString = "System.Windows.Forms.TextBox" Then
            Dim txt As TextBox = CType(objControl, TextBox)
            Dim ValidationMsg As String = _
                txt.Tag.ToString & " must be a positive integer"

            With txt
                If .Name = "txtXCoord" Or .Name = "txtYCoord" Then
                    ValidationMsg &= " or zero."
                Else
                    ValidationMsg &= "."
                End If
                If .Text.Trim = "" Then
                    MessageBox.Show(ValidationMsg, Me.Text, _
                        MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
                    .SelectAll()
                    .Focus()
                    Return False
                End If
                If Not IsNumeric(.Text.Trim) Then
                    MessageBox.Show(ValidationMsg, Me.Text, _
                        MessageBoxButtons.OK, _
                        MessageBoxIcon.Error)
                    .SelectAll()
                    .Focus()
                    Return False
                Else
```

```
            Try
              intVal = CInt(.Text.Trim)
              If txt.Name = "txtXCoord" Or txt.Name = "txtYCoord" Then
                If intVal < 0 Then
                  MessageBox.Show(ValidationMsg, Me.Text, _
                    MessageBoxButtons.OK, _
                    MessageBoxIcon.Error)
                  .SelectAll()
                  .Focus()
                  Return False
                End If
              ElseIf intVal <= 0 Then
                MessageBox.Show(ValidationMsg, Me.Text, _
                  MessageBoxButtons.OK, _
                  MessageBoxIcon.Error)
                .SelectAll()
                .Focus()
                Return False
              End If
            Catch Exp As Exception
              MessageBox.Show("Value must be a positive " & _
                "integer.", Me.Text, MessageBoxButtons.OK, _
                MessageBoxIcon.Error)
              .SelectAll()
              .Focus()
              Return False
            End Try
          End If
        End With
      End If
    Next
    Return True
  End Function
End Class
```

78. Phân chia một ứng dụng

Biểu diễn cách phân chia chức năng trong một ứng dụng. Điều này cho phép người lập trình làm việc có hiệu quả hơn bởi vì các chi tiết được đặt trong một component và chỉ cần nâng cấp ở một nơi.



Các chức năng chính:

- Đọc nội dung một file .CSV vào một DataTable.
- Xây dựng một Class Library như là một component truy xuất dữ liệu.
- Xây dựng một User Control dựa trên một DataGrid.

Solution: **Phan chia mot ung dung**

Project: **DataAccessComponent (Class Library)**

Customer.vb

```
Option Strict On
Imports System.IO
```

```
Public Class Business
    Protected m_FileName As String

    Public Property FileName() As String
        ' Thuộc tính FileName sử dụng cho các thủ tục truy xuất dữ liệu
        Get
            Return m_FileName
        End Get
        Set(ByVal Value As String)
            m_FileName = Value
        End Set
    End Property

    Public Function GetCustomers() As DataTable
        ' Lấy các record trong file và trả về một DataTable

        If m_FileName = "" Then
            Throw (New ArgumentNullException("FileName", "No value for FileName"))
        End If

        Dim objStreamReader As StreamReader = New StreamReader(m_FileName)
        Dim strLine As String
        Dim strColumns() As String

        Dim dtCustomers As New DataTable()
        Dim drCustomer As DataRow

        dtCustomers.Columns.Add("CustomerID")
        dtCustomers.Columns.Add("CompanyName")
        dtCustomers.Columns.Add("ContactName")
        dtCustomers.Columns.Add("Phone")

        ' Lấy một dòng từ file
        strLine = objStreamReader.ReadLine

        ' Vào vòng lặp nếu có dữ liệu
        Do While (Not strLine Is Nothing)
            ' Tạo một DataRow để chứa một dòng
            drCustomer = dtCustomers.NewRow
```

```
            ' Lấy dữ liệu cho các cột, phân tách bằng dấu phẩy
            strColumns = Split(strLine, ",")
            drCustomer.ItemArray = strColumns
            ' Cho DataRow vào DataTable
            dtCustomers.Rows.Add(drCustomer)
            ' Lấy dòng kết tiếp
            strLine = objStreamReader.ReadLine
        Loop
        ' Trả về kết quả
        Return dtCustomers
    End Function

    Public Sub New()
        MyBase.new()
    End Sub

    Public Sub New(ByVal FileName As String)
        MyBase.new()

        m_FileName = FileName
    End Sub
End Class
```

Project: **DataGridControl (User Control)**

DataGridControl.vb

```
Option Strict On
Public Class DataGridControl
    Inherits System.Windows.Forms.UserControl
    Friend WithEvents dgData As System.Windows.Forms.DataGrid

' Code phát sinh giao diện được bỏ qua

    Protected m_FileName As String

    Public Property FileName() As String
        Get
            FileName = m_FileName
        End Get
```

```
    Set(ByVal Value As String)
      m_FileName = Value
    End Set
  End Property

  Sub BindCustomers()
    ' Truy xuất trực tiếp DataAccessComponent, thiết lập DataGrid với tập kết quả.
    ' Ẩn chi tiết cụ thể đối với người lập trình sử dụng control này.
    Dim dtCustomers As DataTable
    ' Tạo instance đối tượng DataAccess
    Dim oBusiness As DataAccessComponent.Business = _
                                    New DataAccessComponent.Business()
    ' Truyền vào tên file sẽ mở
    oBusiness.FileName = m_FileName

    ' Lấy bảng đã chứa dữ liệu
    dtCustomers = oBusiness.GetCustomers

    ' Hiển thị kết quả
    dgData.SetDataBinding(dtCustomers, "")
  End Sub
End Class
```

Project: **Phan chia mot ung dung**

frmMain.vb

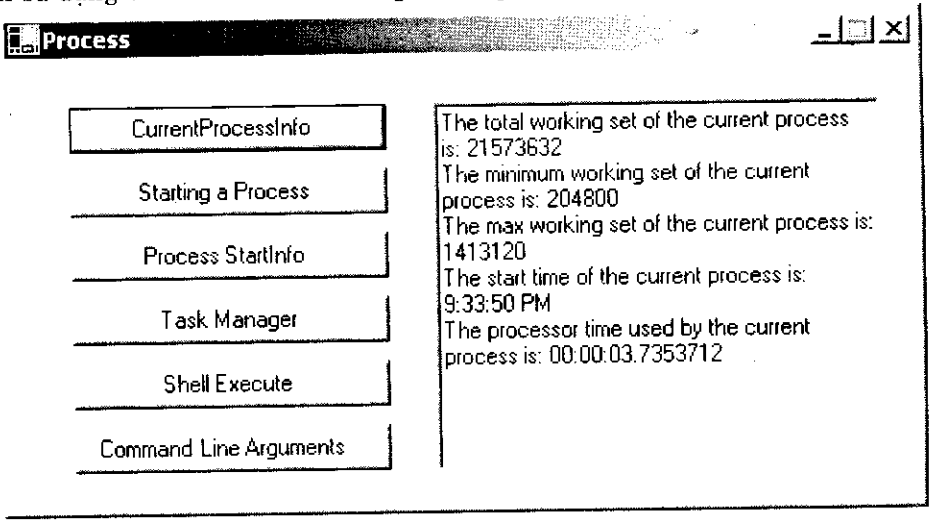
```
Option Strict On
Imports System.IO

Public Class frmMain
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

  Protected strCustomersFile As String = "../Customers.csv"
```

```
Private Sub bnLoadDataGridComponent_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles bnLoadDataGridComponent.Click
    ' Truy xuất trực tiếp đến DataAccessComponent, thiết lập DataGrid với
    ' tập kết quả. Ẩn chi tiết truy xuất file.
    Dim dtCustomers As DataTable

    ' Tạo instance của đối tượng DataAccessComponent
    Dim oBusiness As DataAccessComponent.Business = _
                            New DataAccessComponent.Business()
    ' Truyền vào tên file
    oBusiness.FileName = strCustomersFile

    ' Nhận về một DataTable có dữ liệu
    dtCustomers = oBusiness.GetCustomers()
    ' Hiển thị kết quả
    dgCustomersComponent.SetDataBinding(dtCustomers, "")
End Sub

Private Sub bnLoadDataGridForm_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles bnLoadDataGridForm.Click
    ' Truy xuất file .CSV trực tiếp, thiết lập DataGrid với tập kết quả. Nếu đọc code
    ' thì dễ dàng biết được nó làm gì, tuy nhiên không sử dụng lại được và người lập
    ' trình phải biết tất cả chi tiết.

    Dim objStreamReader As StreamReader = _
                            New StreamReader(strCustomersFile)
    Dim strLine As String
    Dim strColumns() As String

    Dim dtCustomers As New DataTable()
    Dim drCustomer As DataRow

    dtCustomers.Columns.Add("CustomerID")
    dtCustomers.Columns.Add("CompanyName")
    dtCustomers.Columns.Add("ContactName")
    dtCustomers.Columns.Add("Phone")

    strLine = objStreamReader.ReadLine
```

```
    Do While (Not strLine Is Nothing)
      drCustomer = dtCustomers.NewRow

      strColumns = Split(strLine, ",")
      drCustomer.ItemArray = strColumns
      dtCustomers.Rows.Add(drCustomer)
      strLine = objStreamReader.ReadLine
    Loop

    dgCustomersForm.SetDataBinding(dtCustomers, "")
  End Sub

  Private Sub bnLoadUserControl_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles bnLoadDataGridUserControl.Click
    ' User Control thực hiện tất cả công việc, rất dễ sử dụng.
    ' truyền vào tên file để truy xuất
    dgCustomersUserControl.FileName = strCustomersFile
    ' Hiển thị kết quả
    dgCustomersUserControl.BindCustomers()
  End Sub
End Class
```

79. Class Process

Chương trình này chỉ một số cách để gọi và giám sát các process bằng cách sử dụng class Process có trong namespace System.Diagnostics.



Project: **Class Process**

frmMain.vb

```
Option Strict On

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub CurrentProcessInfo_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCurrentProcessInfo.Click
        ' Lấy thông tin về Process hiện hành
        Dim curProc As Process = Process.GetCurrentProcess()

        Dim s As String = "The total working set of the current process is: " + _
            curProc.WorkingSet.ToString() + vbCrLf

        s += "The minimum working set of the current process is: " + _
            curProc.MinWorkingSet.ToString() + vbCrLf
        s += "The max working set of the current process is: " + _
            curProc.MaxWorkingSet.ToString() + vbCrLf
        s += "The start time of the current process is: " + _
            curProc.StartTime.ToLongTimeString() + vbCrLf
        s += "The processor time used by the current process is: " + _
        curProc.TotalProcessorTime.ToString() + vbCrLf

        DisplayText.Text = s
    End Sub

    Private Sub btnStartProcess_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnStartProcess.Click

        ' Gọi một process bằng class Process
        Process.Start("notepad.exe")
    End Sub
```

```
Private Sub btnProcessStartInfo_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnProcessStartInfo.Click
    ' Đối tượng StartInfo cho phép truyền vào các tham số cho ứng dụng trước khi
    ' chạy nó. Trong trường hợp này ta thay đổi trạng thái cửa sổ
    Dim startInfo As New ProcessStartInfo("notepad.exe")
    startInfo.WindowStyle = ProcessWindowStyle.Maximized
    Process.Start(startInfo)
End Sub

Private Sub btnTaskManager_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnTaskManager.Click
    ' Sử dụng class Process ta có thể lấy nhiều thông tin hơn về process
    Dim f As New frmTaskManager()
    f.Show()
End Sub

Private Sub btnShellExecute_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShellExecute.Click

    If Not System.IO.File.Exists("demofile_shell.txt") Then
        Dim sw As New System.IO.StreamWriter("..\demofile_shell.txt")
        sw.WriteLine("Shell Execute Demo")
        sw.Close()
    End If

    ' Class StartInfo cũng có thể sử dụng để gọi Shell mặc nhiên của hệ điều hành.
    Dim startInfo As New ProcessStartInfo("..\demofile_shell.txt")
    startInfo.UseShellExecute = True
    Process.Start(startInfo)
End Sub

Private Sub btnCommandLine_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCommandLine.Click
    ' Chạy Windows Explorer với tham số dòng lệnh
    Dim startInfo As New ProcessStartInfo("explorer.exe")
    startInfo.Arguments = "/n"
    Process.Start(startInfo)
End Sub
End Class
```

frmTaskManager.vb

```
Public Class frmTaskManager
  Inherits System.Windows.Forms.Form
' Code phát sinh form được bỏ qua

  ' Danh sách các Process hiện đang chạy
  Dim ProcessList As New ArrayList()
  Private Sub btnModules_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnModules.Click

    ' Lấy item đang được chọn trong ComboBox. Sau đó hiển thị tất cả các
    ' modules của process đó trên frmModules.
    Dim ProcID As Integer

    Dim ProcessListIndex As Int32 = cboCurrentProcesses.SelectedIndex
    ProcID = CInt(ProcessList(ProcessListIndex))
    Dim frmMod As New frmModules()
    frmMod.ProcessID = ProcID
    frmMod.Show()
  End Sub

  Private Sub frmTaskManager_Load(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles MyBase.Load
    Try
      ' Duyệt qua các process đang chạy và cho chúng vào ComboBox
      Dim ProcessInfo As Process
      For Each ProcessInfo In Process.GetProcesses()
        If ProcessInfo.ProcessName = "EXPLORER" Or _
                                ProcessInfo.ProcessName = "devenv" Then
          cboCurrentProcesses.Items.Add(ProcessInfo.ProcessName)
          ProcessList.Add(ProcessInfo.Id)
        End If
      Next

      cboCurrentProcesses.SelectedIndex = 0
      If btnModules.Enabled = False Then
        btnModules.Enabled = True
      End If
```

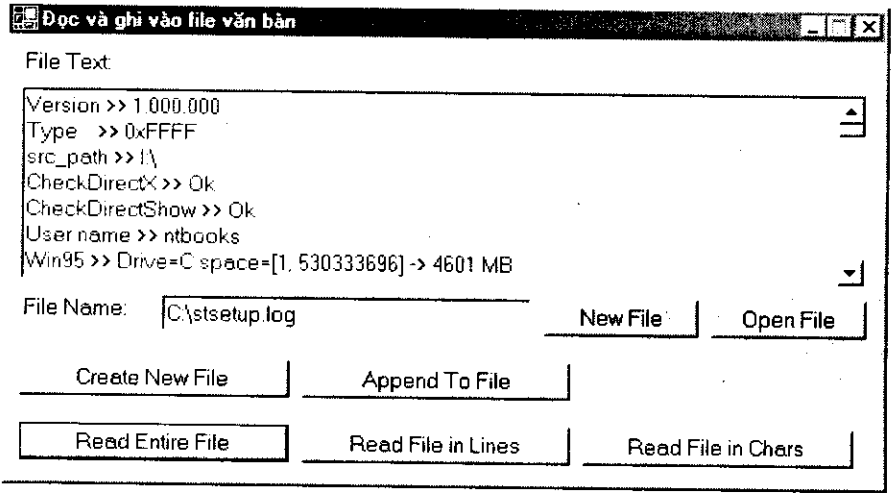
```
    Catch exc As Exception
      MessageBox.Show("Unable to load process names:" & vbCrLf & _
            exc.Message & vbCrLf & "Please choose another process.", _
           "frmTaskManager", _
            MessageBoxButtons.OK, _
            MessageBoxIcon.Error)
      btnModules.Enabled = False
    End Try
  End Sub

  Private Sub cboCurrentProcesses_SelectedIndexChanged(ByVal sender As
  Object, ByVal e As System.EventArgs) Handles
  cboCurrentProcesses.SelectedIndexChanged
    Dim ProcessID As Int32

    Try

      ' Lấy thông tin của process được chọn
      Dim ProcessListIndex As Int32 = cboCurrentProcesses.SelectedIndex
      ProcessID = CInt(ProcessList(ProcessListIndex))
      Dim ProcessInfo As Process = _
        Process.GetProcessById(ProcessID)

      ' Hiển thị thông tin
      txtPriority.Text = ProcessInfo.BasePriority.ToString()
      txtNumberOfThreads.Text = ProcessInfo.Threads.Count.ToString()
      txtMaxWorkingSet.Text = ProcessInfo.MaxWorkingSet.ToString()
      txtMinWorkingSet.Text = ProcessInfo.MinWorkingSet.ToString()
      txtStartTime.Text = ProcessInfo.StartTime.ToLongTimeString()
      txtTotalProcessorTime.Text = ProcessInfo.TotalProcessorTime.ToString()
    Catch exc As Exception
      MessageBox.Show( _
        "Unable to retrieve information for ProcessID : " & ProcessID & _
        vbCrLf & exc.Message, _
        "frmTaskManager", _
        MessageBoxButtons.OK, _
        MessageBoxIcon.Error)
    End Try
  End Sub
End Class
```

frmModules.vb

```
Public Class frmModules
    Inherits System.Windows.Forms.Form
    Private _processID As Integer

' Code phát sinh form được bỏ qua

    Public Property ProcessID() As Integer
        Get
            Return (_processID)
        End Get
        Set(ByVal Value As Integer)
            _processID = Value
        End Set
    End Property

    Private Sub frmModules_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        Dim ProcessInfo As Process = _
            Process.GetProcessById(ProcessID)
        Dim modl As ProcessModuleCollection = ProcessInfo.Modules
        Dim strMod As New System.Text.StringBuilder()
        Dim proMod As ProcessModule
        For Each proMod In modl
            strMod.Append("Module Name: " + proMod.ModuleName + vbCrLf)
        Next proMod
        rchText.Text = strMod.ToString
    End Sub

End Class
```

80. Đọc và ghi file Text

Chương trình này chỉ cách sử dụng các class StreamReader và StreamWriter để đọc và ghi vào các file văn bản.



Các chức năng chính:

- Tạo một file văn bản mới, sẵn sàng để ghi văn bản vào.
- Ghi thêm vào một file đã tồn tại.
- Đọc toàn bộ nội dung file với một lệnh.
- Đọc từng dòng của file.
- Đọc từng ký tự của file.

Project: **Doc va ghi file van ban**

frmMain.vb

```
Imports System.IO
Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua
```

```
    ' Chỉ định một tên file mới và đường dẫn để lưu file.
    Private Sub btnNewFileName_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnNewFileName.Click
        ' Sử dụng SaveFileDialog và đưa tên và đường dẫn của file vào txtFileName
        If Me.sdlgFile.ShowDialog = DialogResult.OK Then
            Me.txtFileName.Text = sdlgFile.FileName
        End If
    End Sub

    ' Chỉ định đường dẫn và tên file của một file đã tồn tại.
    Private Sub btnOpenFileName_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnOpenFileName.Click
        ' Tạo đối tượng OpenFileDialog
        Dim myOpenFileDialog As New OpenFileDialog()

        ' Gán các thuộc tính thích hợp
        myOpenFileDialog.CheckFileExists = True
        myOpenFileDialog.DefaultExt = "txt"
        myOpenFileDialog.InitialDirectory = "C:\"
        myOpenFileDialog.Multiselect = False

        ' Sử dụng hộp thoại OpenFileDialog và đưa đường dẫn và tên file vào txtFileName
        If myOpenFileDialog.ShowDialog = DialogResult.OK Then
            Me.txtFileName.Text = myOpenFileDialog.FileName
        End If

    End Sub

    ' Sử dụng đối tượng StreamWriter để ghi thêm văn bản vào một file đã tồn tại.
    ' Nếu file không tồn tại thì file sẽ được tạo.
    Private Sub btnStreamWriterAppendToFile_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles btnStreamWriterAppendToFile.Click

        ' StreamWriter phải được định nghĩa ngoài khối Try-Catch để có thể tham
        ' chiếu đến nó trong phần Finally.
        Dim myStreamWriter As StreamWriter

        ' Đặt StreamWriter vào khối Try-Catch vì có thể người sử dụng nhập
        ' vào tên file không đúng.
```

```
    Try
        ' Tạo một StreamWriter sử dụng phương thức Shared của class File
        myStreamWriter = File.AppendText(txtFileName.Text)

        ' Ghi toàn bộ nội dung của txtFileText vào StreamWriter
        myStreamWriter.Write(txtFileText.Text)
        myStreamWriter.Flush()
    Catch exc As Exception
        MsgBox("File could not be opened or written to." + vbCrLf + _
          "Please verify that the filename is correct, " + _
          "and that you have read permissions for the desired " + _
          "directory." + vbCrLf + vbCrLf + "Exception: " + exc.Message)
    Finally
        ' Đóng StreamWriter nếu nó đã được tạo
        If Not myStreamWriter Is Nothing Then
            myStreamWriter.Close()
        End If
    End Try
End Sub

' Sử dụng StreamWriter để tạo một file mới và ghi vào nó nội dung trong txtFileText
Private Sub btnStreamWriterCreateFile_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnStreamWriterCreateFile.Click

    ' Kiểm tra và xác nhận người sử dụng có muốn ghi chồng lên
    If File.Exists(txtFileName.Text) Then
        If MsgBox("That file exists. Would you like to overwrite it?", _
            MsgBoxStyle.YesNo) <> MsgBoxResult.Yes Then
            ' Thoát thủ tục
            Return
        End If
    End If

    Dim myStreamWriter As StreamWriter

    Try
        myStreamWriter = File.CreateText(txtFileName.Text)
        myStreamWriter.Write(txtFileText.Text)
        myStreamWriter.Flush()
```

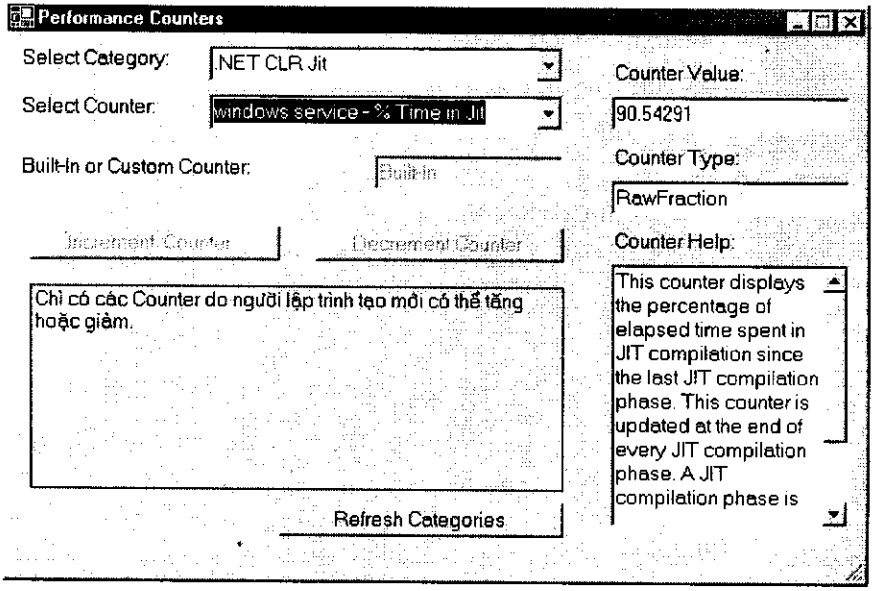
```
    Catch exc As Exception
      MsgBox("File could not be created or written to." + vbCrLf + _
        "Please verify that the filename is correct, " + _
        "and that you have write permissions for the desired " + _
        "directory." + vbCrLf + vbCrLf + "Exception: " + exc.Message)
    Finally
      If Not myStreamWriter Is Nothing Then
        myStreamWriter.Close()
      End If
    End Try
  End Sub

  ' Sử dụng StreamWriter để mở một file và đọc từng dòng.
  Private Sub btnStringReaderReadFileInLines_Click(ByVal sender As
  System.Object, ByVal e As System.EventArgs) Handles
  btnStringReaderReadFileInLines.Click

    Dim myStreamReader As StreamReader
    Dim myInputString As String
    Dim rowCount As Integer = 0

    Try
      myStreamReader = File.OpenText(txtFileName.Text)

      ' Xóa nội dung trong TextBox
      txtFileText.Clear()

      ' Bắt đầu đọc một dòng
      myInputString = myStreamReader.ReadLine()

      ' Tiếp tục đọc trong khi file vẫn còn
      While Not myInputString Is Nothing
        ' xuất nội dung kèm với số thứ tự dòng
        txtFileText.Text += rowCount.ToString() + ": " + _
          myInputString + vbCrLf
        rowCount += 1
        ' Đọc dòng kết tiếp
        myInputString = myStreamReader.ReadLine()
      End While
```

```
    Catch exc As Exception
        ' Thông báo lỗi cho người sử dụng
        MsgBox("File could not be opened or read." + vbCrLf + _
          "Please verify that the filename is correct, " + _
          "and that you have read permissions for the desired " + _
          "directory." + vbCrLf + vbCrLf + "Exception: " + exc.Message)
    Finally
        If Not myStreamReader Is Nothing Then
            myStreamReader.Close()
        End If
    End Try
End Sub

' Mở một file đã tồn tại và đọc nội dung của nó với một dòng lệnh
Private Sub btnStreamReaderReadFromFile_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btnStreamReaderReadFromFile.Click
    Dim myStreamReader As StreamReader

    Try
        myStreamReader = File.OpenText(txtFileName.Text)
        ' Đọc toàn bộ nội dung của file và xuất ra txtFileText
        Me.txtFileText.Text = myStreamReader.ReadToEnd()
    Catch exc As Exception
        MsgBox("File could not be opened or read." + vbCrLf + _
          "Please verify that the filename is correct, " + _
          "and that you have read permissions for the desired " + _
          "directory." + vbCrLf + vbCrLf + "Exception: " + exc.Message)
    Finally
        If Not myStreamReader Is Nothing Then
            myStreamReader.Close()
        End If
    End Try
End Sub

' Mở file và đọc từng ký tự của file.
Private Sub btnStreamReaderReadInChars_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnStreamReaderReadInChars.Click
    Dim myStreamReader As StreamReader
    Dim myNextInt As Integer
```

```
    Try
      myStreamReader = File.OpenText(txtFileName.Text)
      txtFileText.Clear()

       ' Phương thức Read() trả về một số nguyên
      myNextInt = myStreamReader.Read()
       ' Phương thức Read() trả về -1 khi đọc đến cuối file
      While myNextInt <> -1
          ' Chuyển số nguyên đó thành ký tự unicode và cho nó vào TextBox
        txtFileText.Text += ChrW(myNextInt)
          ' Đọc giá trị kế tiếp trong Stream
        myNextInt = myStreamReader.Read()
          ' Cập nhật TextBox để thấy ký tự được thêm vào
        txtFileText.Refresh()
          ' Sleep 100 mili giây để người sử dụng thấy được mỗi lần đọc một ký tự
        System.Threading.Thread.CurrentThread.Sleep(100)
      End While
    Catch exc As Exception
      MsgBox("File could not be opened or read." + vbCrLf + _
        "Please verify that the filename is correct, " + _
        "and that you have read permissions for the desired " + _
        "directory." + vbCrLf + vbCrLf + "Exception: " + exc.Message)
    Finally
      If Not myStreamReader Is Nothing Then
        myStreamReader.Close()
      End If
    End Try
  End Sub

End Class
```

81. Tương tác với các Performance Counter

Chương trình này chỉ cách đọc thông tin và ghi vào Performance Counters sử dụng Visual Basic .NET. Các counter này có thể được sử dụng để theo dõi hiệu xuất nhiều phần của hệ điều hành cũng như các ứng dụng .NET.



Các chức năng chính:

- Tham chiếu đến một perfomance counter đã tồn tại và đọc các giá trị của nó.
- Tạo một perfomance counter để ghi vào và đọc thông tin từ nó.
- Xác định một performance counter là một cái được xây dựng sẵn hay do người lập trình tạo.

Chú ý khi chạy chương trình này:

- Mở solution ra.
- Tạo một performance counter để có thể ghi thông tin vào:

  1. Mở Server Explorer
  2. Chọn Servers -> (Tên máy tính) -> Performance Counters
  3. Nhấn chuột phải vào Performance Counters và chọn "Create New Category..."
  4. Nhập vào tên thể loại ("Category Name"): "Custom Counters".
  5. Nhấn vào "New" để tạo mới một Performance Counter

6. Nhập vào tên (Name): "My First Custom Counter", kiểu (Type): "NumberOfItem32", và nhập mô tả vào phần Counter Description.

Project: **Performance Counters**

frmMain.vb

```
Option Strict On

Imports System.Diagnostics

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Private m_Counter As PerformanceCounter

' Code phát sinh form được bỏ qua

    ' Thủ tục này giảm giá trị của custom counter. Chỉ có các custom counter mới
    ' cho phép gán thuộc tính ReadOnly bằng False
    Private Sub btnDecrementCounter_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnDecrementCounter.Click
        Try
            ' Gán thuộc tính ReadOnly bằng False để có thể thao tác với counter
            m_Counter.ReadOnly = False

            ' Chỉ giảm giá trị khi nào nó có giá trị > 0
            If m_Counter.RawValue > 0 Then
                m_Counter.Decrement()
                sbrStatus.Text = ""
            Else
                ' Hiển thị trạng thái cho người sử dụng
                sbrStatus.Text = "Counter is already zero."
            End If
        Catch exc As Exception
            sbrStatus.Text = "Could not decrement counter."
        End Try

    End Sub
```

```
' Tăng giá trị của custom counter
Private Sub btnIncrementCounter_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnIncrementCounter.Click
    Try
        m_Counter.ReadOnly = False
        m_Counter.Increment()
        sbrStatus.Text = ""
    Catch
        sbrStatus.Text = "Could not increment counter."
    End Try
End Sub

' Cho phép người sử dụng cập nhật lại form sau khi thêm một Performance Counter
Private Sub btnRefreshCategories_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnRefreshCategories.Click
    Me.cboCategories.Items.Clear()
    Me.cboCounters.Items.Clear()
    Me.m_Counter = Nothing
    Me.txtBuiltInOrCustom.Text = ""
    Me.txtCounterHelp.Text = ""
    Me.txtCounterType.Text = ""
    Me.txtCounterValue.Text = ""
    Me.btnDecrementCounter.Enabled = False
    Me.btnIncrementCounter.Enabled = False
    Me.cboCategories.Text = ""
    Me.cboCounters.Text = ""

    ' Gọi Form_Load
    Me.frmMain_Load(Me, New System.EventArgs())
End Sub

' Sự kiện này phát sinh khi người sử dụng thay đổi Counter Category. Thay
' đổi cboCounter tương ứng với các counter thuộc Category đó.
Private Sub cboCategories_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboCategories.SelectedIndexChanged

    Dim myCategory As PerformanceCounterCategory
    Dim myCounters As New ArrayList()
    Dim myCounter As PerformanceCounter
```

```
    Dim myCounterNames() As String

    If cboCategories.SelectedIndex <> -1 Then
        ' Cho tên các Counter vào cboCounter
      Try
          ' Lấy các category có thể có
        myCategory = New PerformanceCounterCategory(
          Me cboCategories.SelectedItem.ToString())

          ' Lấy tất cả instance của category được chọn
        myCounterNames = myCategory.GetInstanceNames(
          ' Nếu không có instance nào, lấy các counter khong có giá trị instance
        If myCounterNames.Length = 0 Then
          myCounters.AddRange(myCategory.GetCounters())
        Else
          Dim i As Integer
          For i = 0 To myCounterNames.Length - 1
            myCounters.AddRange( _
              myCategory.GetCounters(myCounterNames(i)))
          Next
        End If

          ' Xóa nội dung trong cboCounter
        Me cboCounters Items.Clear()
        Me.cboCounters.Text = ""
          ' Cho các counter vào cboCounters. Sử dụng class CounterDisplayItem
          ' để bảo đảm là chúng được hiển thị đúng, và bảo đảm có một tham
          ' chiếu đến một counter được lưu trong item của ComboBox
        For Each myCounter In myCounters
          Me.cboCounters.Items.Add(New CounterDisplayItem(myCounter))
        Next

      Catch exc As Exception
        MsgBox("Unable to list the Counters in this Category." + _
          "Please select another Category.")
      End Try
    End If
  End Sub
```

```
' Sự kiện này phát sinh khi người sử dụng thay đổi counter được chọn.
' Gán counter được chọn cho biến m_Counter.
Private Sub cboCounters_SelectedIndexChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cboCounters.SelectedIndexChanged

    Dim myCounterDisplay As CounterDisplayItem
    Try
        ' Lấy CounterDisplayItem kết với counter được chọn
        myCounterDisplay = CType(cboCounters.SelectedItem, CounterDisplayItem)
        ' Lấy đối tượng PerformanceCounter lưu trong CounterDisplayItem
        m_Counter = myCounterDisplay.Counter
        ' Hiển thị thông tin về Counter
        Me.txtCounterType.Text = m_Counter.CounterType.ToString()
        Me.txtCounterHelp.Text = m_Counter.CounterHelp.ToString()
        Me.sbrStatus.Text = ""

        ' Nếu là custom counter thì cho phép sử dụng các nút thích hợp
        If myCounterDisplay.IsCustom Then
            ' Enable các nút Increment và Decrement
            Me.txtBuiltInOrCustom.Text = "Custom"
            Me.btnDecrementCounter.Enabled = True
            Me.btnIncrementCounter.Enabled = True
        Else
            ' Disable các nút Increment và Decrement
            Me.txtBuiltInOrCustom.Text = "Built-In"
            Me.btnDecrementCounter.Enabled = False
            Me.btnIncrementCounter.Enabled = False
        End If

    Catch exc As Exception
        m_Counter = Nothing
    End Try
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

    Dim myCounters As PerformanceCounter
    Dim myCategory As PerformanceCounterCategory
    Dim myCategories() As PerformanceCounterCategory
```

```
        ' Gán myCategories chứa tất cả các Category có thể có
        myCategories = myCategory.GetCategories()

        Dim myCategoryNames(myCategories.Length - 1) As String
        Dim i As Integer = 0 ' Used as a counter
        ' Duyệt qua các category và cho chúng vào mảng các chuỗi
        For Each myCategory In myCategories
            myCategoryNames(i) = myCategory.CategoryName
            i += 1
        Next

        ' Sắp xếp mảng
        Array.Sort(myCategoryNames)
        ' Cho các giá trị trong mảng vào cboCategories
        Dim nameString As String
        For Each nameString In myCategoryNames
            Me.cboCategories.Items.Add(nameString)
        Next

        ' Cho phép Timer
        Me.tmrUpdateUI.Interval = 500
        Me.tmrUpdateUI.Enabled = True
    End Sub

    ' Cập nhật giá trị của counter sau một chu kỳ thời gian
    Private Sub tmrUpdateUI_Tick(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles tmrUpdateUI.Tick
        ' Kiểm tra xem Counter có tồn tại không, nếu có thì lấy kiểu và giá trị
        Try
            If Not m_Counter Is Nothing Then
                Me.txtCounterValue.Text = m_Counter.NextValue().ToString()
            End If
        Catch exc As Exception
            Me.txtCounterValue.Text = ""
        End Try
    End Sub

End Class
```

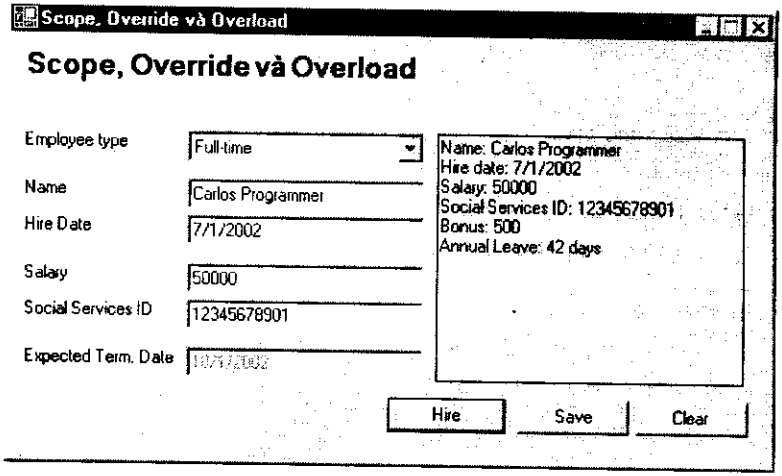
CounterDisplayItem.vb

```
Option Strict On
Imports System.Diagnostics

' Class này được sử dụng để lưu một counter, cũng như định nghĩa phương
' thức ToString() để ComboBox hiển thị instance và tên của counter
Public Class CounterDisplayItem
    ' Lưu một instance của counter bên trong class
    Private m_Counter As PerformanceCounter

    ' Định nghĩa constructor, truyền vào một PerformanceCounter
    Public Sub New(ByVal inCounter As PerformanceCounter)
        If TypeName(inCounter) = "PerformanceCounter" Then
            m_Counter = inCounter
        Else
            m_Counter = Nothing
        End If
    End Sub

    ' Thuộc tính này cho phép lấy ra hoặc lưu vào PerformanceCounter được lưu bởi
    ' một đối tượng CounterDisplayItem
    Public Property Counter() As PerformanceCounter
        Get
            Return m_Counter
        End Get
        Set(ByVal Value As PerformanceCounter)
            m_Counter = Value
        End Set
    End Property

    ' Hàm này override phương thức ToString() để hiển thị thông tin về Counter để
    ' người sử dụng chọn
    Public Overrides Function ToString() As String
        If Not m_Counter Is Nothing Then
            Return m_Counter.InstanceName + " - " + m_Counter.CounterName
        Else
            Return ""
        End If
    End Function
```

```
    ' Thủ tục này trả về True nếu counter là custom counter, nếu khác thì trả về False.
    Public ReadOnly Property IsCustom() As Boolean
        Get
            ' Lưu giá trị hiện hành của thuộc tính ReadOnly
            Dim isReadOnly As Boolean = m_Counter.ReadOnly
            Try
                ' Vì không có phương thức IsCustom, do đó ta dùng một mẹo nhỏ
                m_Counter.ReadOnly = False
                m_Counter.NextValue()
                ' Nếu chạy được đến đây, nó là custom counter
                Return True
            Catch exc As Exception
                ' Đây không phải là một custom couter
                Return False
            Finally
                ' Trả lại giá trị cũ
                m_Counter.ReadOnly = isReadOnly
            End Try
        End Get
    End Property
End Class
```

82. Scope, Override và Overload

Chương trình này cho thấy cách truy xuất nhiều cấp độ đến các thành phần của một class, bao gồm Public, Private, Protected.... Ngoài ra, chương trình này còn chỉ cách mở rộng các class phát sinh (thừa kế) với các tính năng như Overloading và Overriding.



Các chức năng chính:

Chương trình này giả lập một hệ thống trả lương đơn giản, cho phép trả lương cho các nhân viên chính thức (full-time), nhân viên làm việc bán thời gian (part-time) và nhân viên thời vụ (temporary). Mỗi loại nhân viên có các đặc điểm mà loại khác không có. Ví dụ: chỉ nhân viên chính thức mới được nghĩ phép hàng năm, và nhân viên thời vụ có một ngày kết thúc khi được trả lương. Tuy nhiên tất cả các nhân viên đều có một số điểm chung: được trả lương, có tên riêng...

Một số từ khóa sử dụng trong chương trình:

Inherits: Chỉ định class (còn gọi là class cơ sở) để class hiện hành thừa kế.

NotInheritable: Không cho phép sử dụng class này làm class cơ sở.

MustInherit: Không thể tạo instance của class này. Chỉ có một cách sử dụng class này là thừa kế nó.

Overridable: Cho phép một thuộc tính hay một phương thức có thể được viết lại trong class thừa kế. Các phương thức Public mặc nhiên là NotOverridable.

Overrides: Thay thế (override) một thuộc tính hay phương thức được định nghĩa trong class cơ sở.

NotOverridable (kiểu mặc nhiên): Không cho phép override một phương thức hay thuộc tính trong class thừa kế.

MustOverride: Bắt buộc class thừa kế phải override phương thức hay thuộc tính.

Shadows: Cho phép sử dụng lại tên của các thành phần của class cơ sở, kiểu của các thành phần của lớp cơ sở sẽ không có tác dụng trong class thừa kế.

Tầm ảnh hưởng của các từ khóa:

Public: Các class, thủ tục, và biến khai báo với từ khóa Public sẽ có quyền sử dụng toàn cục. Không có giới hạn truy xuất nào đến các thành phần này.

Protected: Các thành phần của class được khai báo với từ khóa Protected chỉ có thể truy xuất được trong cùng class hoặc từ các class thừa kế.

Friend: Các class và thủ tục được khai báo với từ khóa Friend được truy xuất từ bên trong cùng chương trình có khai báo nó và trong cùng một assembly.

Private: Các thành phần của class được khai báo với từ khóa Private chỉ có thể truy xuất từ bên trong class có khai báo chúng.

Shared: Các thủ tục được khai báo với từ khóa Shared có thể sử dụng nó mà không cần tạo một instance của class chứa nó. Ta có thể gọi một thủ tục shared thông qua tên đầy đủ của nó (gồm tên class và tên thủ tục: EmployeeDataManager.WriteEmployeeData), hoặc gọi từ một biến là instance của class đó (edmManager.WriteEmployeeData).

Project: **Scope-Override-Overload**

Employee.vb

```
Option Strict On

' Class này khai báo với từ khóa MustInherit. Nó làm cơ sở cho các class khác.
Public MustInherit Class Employee
    ' Các biến khai báo với từ khóa Protected nên chúng có thể sử dụng được trong các
    ' class thừa kế từ Employee. Từ khóa Private sẽ làm cho các biến này chỉ có thể
    ' truy xuất trong class Employee.
    '
    ' Chỉ có thể sử dụng Protected cho các biến cấp class, bên ngoài các thủ tục.
    ' Ta không thể sử dụng nó cho các biến cấp module, namespace hoặc file.
    Protected c_HireDate As DateTime
    Protected c_Name As String
    Protected c_Salary As Decimal
    Protected c_SocialServicesID As String

    ' Constructor mặc nhiên. Thủ tục này chạy khi một instance của đối tượng Employee
    ' được tạo mà không có tham số. Ta có thể sử dụng nó để gán các giá trị mặc
    ' nhiên cho các thuộc tính...

    Public Sub New()
        c_HireDate = DateTime.Now.Date
    End Sub

    ' Đây là một overload constructor có tham số. Thủ tục này chạy khi instance được
    ' tạo với tham số là một chuỗi. Construcotor có tham số cho phép dữ liệu được
    ' truyền vào cho đối tượng vào lúc tạo một instance của nó.

    Public Sub New(ByVal Name As String)
        c_Name = Name
        c_HireDate = DateTime.Now.Date
    End Sub
```

```
Public ReadOnly Property Bonus() As Decimal
    Get
      Return ComputeBonus()
    End Get
  End Property

  Public Property HireDate() As Date
    Get
      Return c_HireDate
    End Get
    Set(ByVal Value As Date)
      If Value <= DateTime.Now.Date Then
        c_HireDate = Value
      Else
        Throw New ArgumentException( _
          "Hire Date cannot be later than today", "HireDate")
      End If
    End Set
  End Property

  Public Property Name() As String
    Get
      Return c_Name
    End Get
    Set(ByVal Value As String)
      c_Name = Value
    End Set
  End Property

  ' Nếu muốn các class thừa kế có các phiên bản tính lương riêng ta khai báo
  ' từ khóa MustOverride để bắt buộc các class thừa kế viết code cho thủ tục này.
  '
  ' Chú ý là không có dòng code nào cho thủ tục thuộc tính này và không
  ' có lệnh End Property
  Public MustOverride Property Salary() As Decimal
  Public Overridable Property SocialServicesID() As String
    Get
      Return c_SocialServicesID
    End Get
```

```
        Set(ByVal Value As String)
            If IsNumeric(Value) And Len(Value) = 11 Then
                c_SocialServicesID = Value
            Else
                Throw New ArgumentException( _
                    "Social Security Number must be 11 numeric " & _
                    "characters", "SocialServicesID")
            End If
        End Set
    End Property

    Public MustOverride Function ComputeBonus() As Decimal

    ' Hire là một phương thức overload. Có thể gọi theo ba cách khác nhau.
    Public Sub Hire(ByVal Name As String)
        c_Name = Name
    End Sub

    Public Sub Hire(ByVal Name As String, ByVal HireDate As DateTime)
        c_Name = Name
        c_HireDate = HireDate
    End Sub

    Public Sub Hire(ByVal Name As String, ByVal HireDate As DateTime, ByVal
    StartingSalary As Decimal)
        c_Name = Name
        c_HireDate = HireDate
        c_Salary = StartingSalary
    End Sub
End Class
```

FullTimeEmployee.vb

```
Option Strict On

' Class FullTimeEmployee thừa kế từ class Employee và do đó nó có tất cả các
' thuộc tính, phương thức, và các sự kiện mà Employee có. Ngoài ra nó còn mở
' rộng thêm bằng cách thêm phương thức ComputeAnnualLeave và override
' phương thức ComputeBonus.
```

```
Public Class FullTimeEmployee
    Inherits Employee

    ' Constructor này gọi constructor mặc nhiên trong class cơ sở
    Public Sub New()
        MyBase.New()
    End Sub

    ' Constructor này gọi constructor có tham số của class cơ sở.
    ' Truyền vào tên Employee
    Public Sub New(ByVal Name As String)
        MyBase.New(Name)
    End Sub

    ' AnnualLeave là thuộc tính tính theo ngày, và chỉ có trong class FullTimeEmployee
    Public ReadOnly Property AnnualLeave() As Integer
        Get
            Return ComputeAnnualLeave()
        End Get
    End Property

    ' Thủ tục tính lương chưa được viết code trong class cơ sở
    Public Overrides Property Salary() As Decimal
        Get
            Return c_Salary
        End Get
        Set(ByVal Value As Decimal)
            If Value < 30000.0 Or Value > 500000.0 Then
                Throw New ArgumentOutOfRangeException( _
                    "Salary", _
                    "Full-time employee salary must be between " & _
                    "$30,000 and $500,000")
            Else
                c_Salary = Value
            End If
        End Set
    End Property
```

```
    ' Thủ tục mở rộng class cơ sở. Phương thức này không có trong class cơ sở cũng
    ' như class PartTimeEmployee.
    ' Phương thức này tính số ngày employee đã làm việc với công ty và cho
    ' emplyee 4 ngày phép cho mỗi quý.
    Public Function ComputeAnnualLeave() As Integer
        ' Giả sử trả về một con số
        Return 42
    End Function

    ' Viết code thủ tục ComputeBonus cho nhân viên full-time
    Public Overrides Function ComputeBonus() As Decimal
        ' Nhân viên full-time được hưởng thưởng hàng năm là 1% lương
        Return c_Salary * CType(0.01, Decimal)
    End Function
End Class
```

PartTimeEmployee.vb

```
Option Strict On
' Class này thừa kế từ class Employee.
Public Class PartTimeEmployee
    Inherits Employee
    ' c_MinHoursPerWeek là private, chỉ thấy (sử dụng) được trong class này.
    Private c_MinHoursPerWeek As Double

    Public Sub New()
        MyBase.New()
    End Sub

    Public Sub New(ByVal Name As String)

        MyBase.New(Name)
    End Sub

    ' Tính số giờ tối thiểu mà nhân viên part-time phải làm trong một tuần
    Public Property MinHoursPerWeek() As Double
        Get
            Return c_MinHoursPerWeek
        End Get
```

```
    Set(ByVal Value As Double)
      c_MinHoursPerWeek = Value
    End Set
  End Property

  Public Overrides Property Salary() As Decimal
    Get
      Return c_Salary
    End Get
    Set(ByVal Value As Decimal)
      If Value < 10000.0 Or Value > 30000.0 Then
        Throw New ArgumentOutOfRangeException( _
          "Salary", "Part-time employee salary must " & _
          "be between $10,000 and $30,000")
      Else
        c_Salary = Value
      End If
    End Set
  End Property

  Public Overrides Function ComputeBonus() As Decimal
    ' Nhân viên Part-time nhận thưởng 0.5% lương hàng năm
    Return c_Salary * CType(0.005, Decimal)
  End Function

  Public Overloads Sub Hire(ByVal Name As String, ByVal HireDate As DateTime,
  ByVal StartingSalary As Decimal, ByVal MinHoursPerWeek As Double)
    c_Name = Name
    c_HireDate = HireDate
    c_Salary = StartingSalary
    c_MinHoursPerWeek = MinHoursPerWeek
  End Sub
End Class
```

TempEmployee.vb

```
Option Strict On

' TempEmployee thừa kế từ class Employee.
'
```

```
' Class này Shadow phương thức Hire, nghĩa là phương thức Hire của class Employee
' không thể sử dụng được từ đối tượng TempEmployee
Public Class TempEmployee
    Inherits Employee

    Public ExpectedTermDate As DateTime

    Public Overrides Property Salary() As Decimal
        Get
            Return c_Salary
        End Get
        Set(ByVal Value As Decimal)
            If Value < 10000.0 Or Value > 20000.0 Then
                Throw New ArgumentOutOfRangeException( _
                    "Salary", "Temporary employee salary must " & _
                    "be between $10,000 and $20,000")
            Else
                c_Salary = Value
            End If
        End Set
    End Property

    Public Overrides Function ComputeBonus() As Decimal
        ' Nhân viên tạm thời không có thưởng
        Return 0
    End Function

    Public Shadows Sub Hire(ByVal Name As String, ByVal HireDate As DateTime,
    ByVal StartingSalary As Decimal, ByVal EmploymentEndDate As DateTime)
        c_Name = Name
        c_HireDate = HireDate
        c_Salary = StartingSalary
        ExpectedTermDate = EmploymentEndDate
    End Sub
End Class
```

EmployeeDataManager.vb

```
Option Strict On
```

```
' Class EmployeeDataManager có các phương thức để chuyển đến và lấy dữ liệu của
' employee từ database. Bằng cách khai báo với từ khóa Friend, tất cả các thành
' phần của nó có thể sử dụng được trong assembly này.

' Từ khóa NotInheritable nghĩa là không cho phép class khác thừa kế nó
Friend NotInheritable Class EmployeeDataManager

    ' WriteEmployeeData là một hàm overload, với ba cách gọi khác nhau.
    ' Nó cũng là một hàm Shared.
    Public Shared Function WriteEmployeeData(ByVal Employee As
    FullTimeEmployee) As String
        ' Code để ghi dữ liệu vào database
        Return "Full-Time Employee data written to database."
    End Function

    Public Shared Function WriteEmployeeData(ByVal Employee As
    PartTimeEmployee) As String
        ' Code để ghi dữ liệu vào database
        Return "Part-Time Employee data written to database."
    End Function

    Public Shared Function WriteEmployeeData(ByVal Employee As TempEmployee)
    As String
        ' Code để ghi dữ liệu vào database
        Return "Temporary Employee data written to database."
    End Function

    Public Shared Function ReadEmployeeData(ByVal Employee As
    FullTimeEmployee) As String

        ' Code để đọc dữ liệu từ database
        Return "Full-Time Employee data read from database."
    End Function

    Public Shared Function ReadEmployeeData(ByVal Employee As
    PartTimeEmployee) As String

        ' Code để đọc dữ liệu từ database
        Return "Part-Time Employee data read from database."
    End Function
```

```
    Public Shared Function ReadEmployeeData(ByVal Employee As TempEmployee)
    As String
        ' Code để đọc dữ liệu từ database
        Return "Temporary Employee data read from database."
    End Function
End Class
```

frmMain.vb

```
Option Strict On

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Private c_empFull As FullTimeEmployee
    Private c_empPart As PartTimeEmployee
    Private c_empTemp As TempEmployee

' Code phát sinh form được bỏ qua

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnClear.Click
        Dim ctl As Control
        cboEmployeeType.SelectedIndex = -1
        For Each ctl In Me.Controls
            If TypeOf ctl Is TextBox Then
                ctl.Text = vbNullString
            End If
        Next
    End Sub

    Private Sub btnHire_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnHire.Click
        If Not ValidateData() Then
            Return
        End If
        Select Case cboEmployeeType.SelectedIndex
            Case -1
                Exit Sub
```

```
        Case 0
            HireFullTimeEmployee()
        Case 1
            HirePartTimeEmployee()
        Case 2
            HireTempEmployee()
    End Select
End Sub

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    Dim strResult As String

    ' Giả lập hành động lưu vào database sử dụng phương thức Shared
    ' của class EmployeeDataManager
    Select Case cboEmployeeType.SelectedIndex
        Case -1
            Exit Sub
        Case 0
            strResult = EmployeeDataManager.WriteEmployeeData(c_empFull)
        Case 1
            strResult = EmployeeDataManager.WriteEmployeeData(c_empPart)
        Case 2
            strResult = EmployeeDataManager.WriteEmployeeData(c_empTemp)
    End Select
    MsgBox(strResult, MsgBoxStyle.Information, Me.Text)
End Sub

' Theo dõi các thay đổi trên các TextBox, enable nút Hire khi đủ dữ liệu
Private Sub TextBoxes_TextChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles txtName.TextChanged, txtHireDate.TextChanged,
txtSalary.TextChanged, txtSocialServicesID.TextChanged,
txtExpectedTermDate.TextChanged, cboEmployeeType.SelectedIndexChanged
    Select Case cboEmployeeType.SelectedIndex

        Case -1
            'btnHire.Enabled = False
            txtName.Enabled = False
            txtHireDate.Enabled = False
            txtSalary.Enabled = False
```

```
            txtSocialServicesID.Enabled = False
            txtExpectedTermDate.Enabled = False
        Case 0, 1
            'btnHire.Enabled = True
            txtName.Enabled = True
            txtHireDate.Enabled = True
            txtSalary.Enabled = True
            txtSocialServicesID.Enabled = True
            txtExpectedTermDate.Enabled = False
        Case 2
            'btnHire.Enabled = True
            txtName.Enabled = True
            txtHireDate.Enabled = True
            txtSalary.Enabled = True
            txtSocialServicesID.Enabled = True
            txtExpectedTermDate.Enabled = True
    End Select

    btnHire.Enabled = _
      txtName.Text.Trim.Length <> 0 And _
      txtHireDate.Text.Trim.Length <> 0 And _
      txtSalary.Text.Trim.Length <> 0 And _
      txtSocialServicesID.Text.Trim.Length <> 0 And _
      cboEmployeeType.SelectedIndex >= 0

    ' Nếu là Temp Employee, kiểm tra TextBox ExpectedTermDate
    If cboEmployeeType.SelectedIndex = 2 Then
      btnHire.Enabled = btnHire.Enabled And _
                        txtExpectedTermDate.Text.Trim.Length <> 0
    End If
    btnSave.Enabled = False
End Sub

Private Sub HireFullTimeEmployee()

    c_empFull = New FullTimeEmployee()
    Try
      With c_empFull
        .Hire(txtName.Text.Trim, CType(txtHireDate.Text.Trim, DateTime), _
          CType(txtSalary.Text.Trim, Decimal))
```

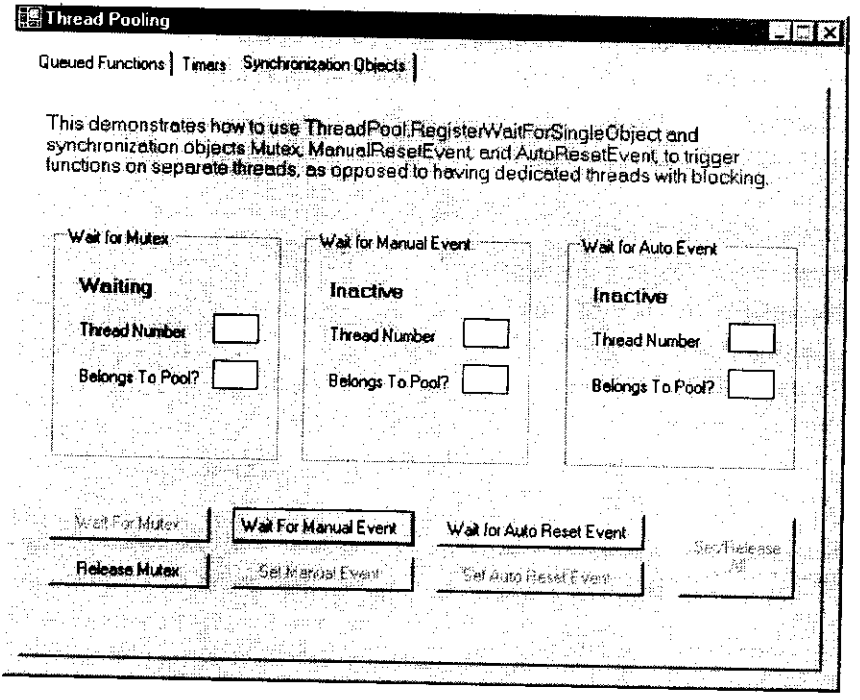
```
            .SocialServicesID = txtSocialServicesID.Text.Trim
            txtResults.Clear()
            txtResults.Text &= "Name: " & .Name & vbCrLf & _
               "Hire date: " & .HireDate & vbCrLf & _
               "Salary: " & .Salary & vbCrLf & _
               "Social Services ID: " & .SocialServicesID & vbCrLf & _
               "Bonus: " & .Bonus & vbCrLf & _
               "Annual Leave: " & .AnnualLeave & " days"
         End With
         btnSave.Enabled = True
      Catch exp As Exception
         MsgBox(exp.ToString(), MsgBoxStyle.Critical, Me.Text)
      End Try
   End Sub

   Private Sub HirePartTimeEmployee()
      c_empPart = New PartTimeEmployee()
      Try
         With c_empPart
            .Hire(txtName.Text.Trim, CType(txtHireDate.Text.Trim, DateTime), _
               CType(txtSalary.Text.Trim, Decimal), 20)
            .SocialServicesID = txtSocialServicesID.Text.Trim
            txtResults.Clear()
            txtResults.Text &= "Name: " & .Name & vbCrLf & _
               "Hire date: " & .HireDate & vbCrLf & _
               "Salary: " & .Salary & vbCrLf & _
               "Social Services ID: " & .SocialServicesID & vbCrLf & _
               "Bonus: " & .Bonus & vbCrLf & _
               "Min hrs. per week: " & .MinHoursPerWeek
         End With
         btnSave.Enabled = True
      Catch exp As Exception
         MsgBox(exp.ToString(), MsgBoxStyle.Critical, Me.Text)
      End Try

   End Sub
```

```
Private Sub HireTempEmployee()
    Dim c_empTemp As New TempEmployee()
    Try
        With c_empTemp
            .Hire(txtName.Text.Trim, CType(txtHireDate.Text.Trim, DateTime), _
                CType(txtSalary.Text.Trim, Decimal), _
                CType(txtExpectedTermDate.Text.Trim, DateTime))
            .SocialServicesID = txtSocialServicesID.Text.Trim
            txtResults.Clear()
            txtResults.Text &= "Name: " & .Name & vbCrLf & _
                "Hire date: " & .HireDate & vbCrLf & _
                "Salary: " & .Salary & vbCrLf & _
                "Social Services ID: " & .SocialServicesID & vbCrLf & _
                "Bonus: " & .Bonus & vbCrLf & _
                "Expected termination date: " & .ExpectedTermDate.ToShortDateString
        End With
        btnSave.Enabled = True
    Catch exp As Exception
        MsgBox(exp.ToString(), MsgBoxStyle.Critical, Me.Text)
    End Try
End Sub

Private Function ValidateData() As Boolean
    Dim dataInValid As Boolean = True

    If Not IsDate(txtHireDate.Text) Then
        MsgBox("Hire Date must be a date in the format MM/DD/YY.", _
            MsgBoxStyle.Exclamation, Me.Text)
        dataInValid = False
    End If

    If txtSocialServicesID.Text.Length <> 11 Then
        MsgBox("Social Services ID must be 11 characters in length", _
         MsgBoxStyle.Exclamation, Me.Text)
        dataInValid = False
    End If

    Return (dataInValid)
End Function
```

```
    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        txtName.Text = "Carlos Programmer"
        txtHireDate.Text = "7/1/2002"
        txtSalary.Text = "50000"
        txtSocialServicesID.Text = "12345678901"
        txtExpectedTermDate.Text = "10/1/2002"

    End Sub
End Class
```

83. Sử dụng Thread Pooling

Chương trình này chỉ cách sử dụng Thread Pooling trong Visual Basic.NET.



Các chức năng chính:

- Function Queuing: So sánh việc chạy ba process có thứ tự, sử dụng các thread riêng biệt, và sử dụng ThreadPool.QueueUserWorkItem.
- Timers: Sử dụng class Threading.Timer.
- Synchronization Object: Sử dụng ThreadPool để chạy các process có đợi các đối tượng đồng bộ hóa.

Project: **Thread Pooling**

ProcessGroup.vb

```
Option Strict On

' Class này gom nhóm hai process thứ hai trong các tab non-threaded, threaded,
' và thread pool
Public Class ProcessGroup
    ' Constructor
    Public Sub New(ByVal ActiveLabel As Label, _
                   ByVal ThreadNumLabel As Label, _
                   ByVal IsPoolThreadLabel As Label)
        lblActive = ActiveLabel
        lblThreadNum = ThreadNumLabel
        lblIsPoolThread = IsPoolThreadLabel
        ' Cộng thêm 1 vào thuộc tính Shared numberOfProcesses
        numberOfProcesses += 1
    End Sub

    Private lblActive As Label
    Private lblIsPoolThread As Label
    Private lblThreadNum As Label
    ' Các thành phần Shared
    Private Shared highIntensity As Boolean
    Private Shared numberOfProcesses As Integer = 0
    Private Shared processesCompleted As Integer
    Private Shared ticksElapsed As Integer

    Shared ReadOnly Property GetTicksElapsed() As Integer
        Get
            Return ticksElapsed
        End Get
    End Property

    Shared WriteOnly Property SetHighIntensity() As Boolean
        Set(ByVal Value As Boolean)
            highIntensity = Value
        End Set
    End Property
```

```
Public Event Completed()

' Khởi tạo các thành phần Shared
Shared Sub PrepareToRun()
    processesCompleted = 0
    ticksElapsed = Environment.TickCount
End Sub

' Thủ tục này chạy process thứ hai, cập nhật các label
Rublic Sub Run()
    ' Cho biết đang Active
    lblActive.Text = "Active"
    lblActive.ForeColor = Color.Green
    lblActive.Refresh()

' Cập nhật label ThreadNum. GetHashCode trả về giá trị số duy nhất cho mỗi thread.
    lblThreadNum.Text = Threading.Thread.CurrentThread.GetHashCode().ToString
    lblThreadNum.Refresh()

    ' Cập nhật label IsThreadPooled
    If Threading.Thread.CurrentThread.IsThreadPoolThread Then
        lblIsPoolThread.Text = "Yes"
    Else
        lblIsPoolThread.Text = "No"
    End If
    lblIsPoolThread.Refresh()

    ' Nếu highIntensity được chọn thì loop qua TickCount trong 2 giây để sử dụng
    ' tối đa CPU, nếu không chọn thì đợi trong 2 giây.
    If highIntensity Then
        Dim ticks As Integer = Environment.TickCount
        While Environment.TickCount - ticks < 2000
        End While
    Else

        Threading.Thread.CurrentThread.Sleep(2000)
    End If
```

```
        ' Process đã kết thúc
        lblActive.Text = "Inactive"
        lblActive.ForeColor = Color.Red
        lblActive.Refresh()
        ' Cộng 1 vào thuộc tính shared ProcessCompleted. Nếu tất cả các process đã
        ' kết thúc thì phát sinh sự kiện đã hoàn tất. Điều này cần thiết khi muốn cập
        ' nhật giao diện kịp thời.
        processesCompleted += 1
        If processesCompleted = numberOfProcesses Then
            ticksElapsed = Environment.TickCount - ticksElapsed
            RaiseEvent Completed()
        End If
    End Sub

    ' Thủ tục này là một callback cho Threadpool.QueueWorkItem để tránh nhầm
    ' lẫn giữa hai cách gọi process bên dưới.
    Private Sub RunPooledThread(ByVal state As Object)
        Run()
    End Sub

    ' Thêm một yêu cầu vào Threadpool với một callback gọi RunPooledThread
    Sub StartPooledThread()
        ' Tạo một callback đến thủ tục RunPooledThread
        Dim callback As New Threading.WaitCallback(AddressOf RunPooledThread)
        ' Đưa nó vào hàng đợi để chạy process
        Threading.ThreadPool.QueueUserWorkItem(callback, Nothing)
    End Sub

    ' Start một thread mới, chạy thủ tục Run
    Sub StartThread()
        Dim newThread As New Threading.Thread(AddressOf Run)
        newThread.Start()
    End Sub

End Class
```

TimerGroup.vb

```
Option Strict On
' Class này gom nhóm chức năng class Threading.Timer. Mục đích chính là cho
' phép thủ tục OnTimer có thể biết cập nhật label nào bởi vì ta không thể truyền
' tham số vào nó
Public Class TimerGroup
    ' Constructor
    Public Sub New(ByVal OutputLabel As Label, _
                   ByVal ThreadNumLabel As Label, _
                   ByVal IsPoolThreadLabel As Label)
        lblOutput = OutputLabel
        lblThreadNum = ThreadNumLabel
        lblIsPoolThread = IsPoolThreadLabel
    End Sub

    Private lblIsPoolThread As Label
    Private lblOutput As Label
    Private lblThreadNum As Label
    Private myTimer As Threading.Timer

    ' Hàm callback cho myTimer, được gọi bất cứ khi nào timer phát sinh sự kiện
    Private Sub OnTimer(ByVal State As Object)
        ' Cập nhật các label
        lblOutput.Text = Now.Hour & ":" & Now.Minute & "." & Now.Second
        lblThreadNum.Text = Threading.Thread.CurrentThread.GetHashCode().ToString
        If Threading.Thread.CurrentThread.IsThreadPoolThread Then
            lblIsPoolThread.Text = "Yes"
        Else
            lblIsPoolThread.Text = "No"
        End If
    End Sub

    ' Thủ tục này start cho timer, bằng cách tạo một đối tượng Threading.Timer với
    ' một callback OnTimer
    Public Sub StartTimer(ByVal period As Integer)
        ' Không được start một timer NULL
        If IsNothing(myTimer) Then
            ' Tạo một callback
            Dim callback As New Threading.TimerCallback(AddressOf OnTimer)
```

```
        ' Start timer, truyền vào tần số thực hiện. Chú ý: không nhầm
        ' lẫn Threading.Timer với Timers.Timer
        myTimer = New Threading.Timer(callback, Nothing, 0, period)
      End If
    End Sub

    ' Kết thúc timer bằng cách huỷ đối tượng Threading.Timer
    Public Sub StopTimer()
      myTimer.Dispose()
      myTimer = Nothing
    End Sub
End Class
```

frmMain.vb

```
Option Strict On

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Private autoResetEvent1 As Threading.AutoResetEvent
    Private manualResetEvent1 As Threading.ManualResetEvent
    Private mutex1 As Threading.Mutex
    Private processGroup1 As ProcessGroup
    Private processGroup2 As ProcessGroup
    Private processGroup3 As ProcessGroup
    Private timerGroup1 As TimerGroup
    Private timerGroup2 As TimerGroup

' Code phát sinh form được bỏ qua

    ' Thủ tục này gọi ba process liên tiếp, không sử dụng thread
    Private Sub btnNonthreaded_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnNonthreaded.Click
        ' Không cho phép các process khác khi chưa kết thúc
        btnNonthreaded.Enabled = False
        btnThreaded.Enabled = False
        btnThreadPool.Enabled = False
        ' Chuẩn bị các biến shared
        ProcessGroup.PrepareToRun()
```

```
        ' Chạy từng process
        processGroup1.Run()
        processGroup2.Run()
        processGroup3.Run()
    End Sub

    ' Thủ tục này giải phóng Mutex, cho phép phát sinh sự kiện OnMutexReleased
    Private Sub btnReleaseMutex_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReleaseMutex.Click
        btnReleaseMutex.Enabled = False
        btnSetReleaseAll.Enabled = False

        mutex1.ReleaseMutex()
    End Sub

    ' Gửi tín hiệu sự kiện Auto Reset, cho phép phát sinh sự kiện OnAutoEventSet
    Private Sub btnSetAutoEvent_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSetAutoEvent.Click
        btnSetAutoEvent.Enabled = False
        btnSetReleaseAll.Enabled = False

        autoResetEvent1.Set()
    End Sub

    ' Gửi tín hiệu sự kiện Manual Reset, phát sinh sự kiện OnManualEventSet
    Private Sub btnSetManualEvent_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSetManualEvent.Click
        btnSetManualEvent.Enabled = False
        btnSetReleaseAll.Enabled = False

        manualResetEvent1.Set()
    End Sub

    ' Thủ tục này giải phóng Mutex và set các sự kiện auto và manual
    Private Sub btnSetReleaseAll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSetReleaseAll.Click
        btnReleaseMutex.Enabled = False
        btnSetManualEvent.Enabled = False
        btnSetAutoEvent.Enabled = False
        btnSetReleaseAll.Enabled = False
```

```
    mutex1.ReleaseMutex()
    manualResetEvent1.Set()
    autoResetEvent1.Set()
  End Sub

  ' Start và Stop các Timer
  Private Sub btnStartStop_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnStartStop.Click
    If btnStartStop.Text = "&Start" Then
      btnStartStop.Text = "&Stop"
      timerGroup1.StartTimer(1000)
      timerGroup2.StartTimer(2000)
    Else
      btnStartStop.Text = "&Start"
      timerGroup1.StopTimer()
      timerGroup2.StopTimer()
    End If
  End Sub

  ' Thủ tục này gọi ba process trong ba thread riêng biệt
  Private Sub btnThreaded_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnThreaded.Click
    btnNonthreaded.Enabled = False
    btnThreaded.Enabled = False
    btnThreadPool.Enabled = False

    ProcessGroup.PrepareToRun()

    ' Chạy mỗi process trong thread riêng của nó
    processGroup1.StartThread()
    processGroup2.StartThread()
    processGroup3.StartThread()
  End Sub

  ' Thủ tục này yêu cầu đến ThreadPool để chạy ba process
  Private Sub btnThreadPool_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnThreadPool.Click
    btnNonthreaded.Enabled = False
    btnThreaded.Enabled = False
```

```
    btnThreadPool.Enabled = False

    ProcessGroup.PrepareToRun()

    ' Gửi yêu cầu đến ThreadPool để chạy ba process
    processGroup1.StartPooledThread()
    processGroup2.StartPooledThread()
    processGroup3.StartPooledThread()
End Sub


Private Sub btnWaitForAutoEvent_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnWaitForAutoEvent.Click
    autoResetEvent1 = New Threading.AutoResetEvent(False)

    Dim callback As New Threading.WaitOrTimerCallback( _
                                        AddressOf OnAutoEventSet)

    Threading.ThreadPool.RegisterWaitForSingleObject(autoResetEvent1, _
            callback, Nothing, Threading.Timeout.Infinite, False)

    lblAutoEventStatus.Text = "Waiting"
    lblAutoEventStatus.ForeColor = Color.Blue

    btnWaitForAutoEvent.Enabled = False
    btnSetAutoEvent.Enabled = True

    If Not btnWaitForManualEvent.Enabled And Not btnWaitForMutex.Enabled Then
        btnSetReleaseAll.Enabled = True
    End If
End Sub

Private Sub btnWaitForManualEvent_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnWaitForManualEvent.Click
    manualResetEvent1 = New Threading.ManualResetEvent(False)

    Dim callback As New Threading.WaitOrTimerCallback( _
                                    AddressOf OnManualEventSet)
```

```
    Threading.ThreadPool.RegisterWaitForSingleObject(manualResetEvent1, _
                callback, Nothing, Threading.Timeout.Infinite, True)

    lblManualEventThreadStatus.Text = "Waiting"
    lblManualEventThreadStatus.ForeColor = Color.Blue

    btnWaitForManualEvent.Enabled = False
    btnSetManualEvent.Enabled = True

    If Not btnWaitForMutex.Enabled And Not btnWaitForAutoEvent.Enabled Then
      btnSetReleaseAll.Enabled = True
    End If
  End Sub

  ' Thủ tục này tạo một mutex, và đưa mutex process vào trạng thái đợi cho đến
  ' khi mutex được giải phóng
  Private Sub btnWaitForMutex_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnWaitForMutex.Click
    mutex1 = New Threading.Mutex(True)

    ' Tạo một callback đến OnMutexReleased
    Dim callback As New Threading.WaitOrTimerCallback( _
                                AddressOf OnMutexReleased)

    ' Đăng ký mutex để phát sinh OnManualEventSet khi giải phóng.
    Threading.ThreadPool.RegisterWaitForSingleObject(mutex1, callback, _
                Nothing, Threading.Timeout.Infinite, True)

    lblMutexThreadStatus.Text = "Waiting"
    lblMutexThreadStatus.ForeColor = Color.Blue

    btnWaitForMutex.Enabled = False
    btnReleaseMutex.Enabled = True
    If Not btnWaitForManualEvent.Enabled And _
                            Not btnWaitForAutoEvent.Enabled Then
      btnSetReleaseAll.Enabled = True
    End If
  End Sub
```

```
Private Sub chkHighIntensity_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles chkHighIntensity.CheckedChanged
    ProcessGroup.SetHighIntensity = chkHighIntensity.Checked
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    processGroup1 = New ProcessGroup(lblProcess1Active, _
                lblProcess1ThreadNum, lblProcess1IsPoolThread)
    processGroup2 = New ProcessGroup(lblProcess2Active, _
                lblProcess2ThreadNum, lblProcess2IsPoolThread)
    processGroup3 = New ProcessGroup(lblProcess3Active, _
                lblProcess3ThreadNum, lblProcess3IsPoolThread)

    AddHandler processGroup1.Completed, AddressOf OnProcessesCompleted
    AddHandler processGroup2.Completed, AddressOf OnProcessesCompleted
    AddHandler processGroup3.Completed, AddressOf OnProcessesCompleted

    ProcessGroup.PrepareToRun()

    timerGroup1 = New TimerGroup(lblTimer1Output, lblTimer1ThreadNum, _
                        lblTimer1IsThreadPool)
    timerGroup2 = New TimerGroup(lblTimer2Output, lblTimer2ThreadNum, _
                        lblTimer2IsThreadPool)
End Sub

' Thủ tục callback khi reset event được set
Private Sub OnAutoEventSet(ByVal obj As Object, ByVal TimedOut As Boolean)
    lblAutoEventStatus.Text = "Running"
    lblAutoEventStatus.ForeColor = Color.Green

    lblAutoEventThreadNum.Text = _
        Threading.Thread.CurrentThread.GetHashCode().ToString

    If Threading.Thread.CurrentThread.IsThreadPoolThread Then
        lblAutoEventIsPoolThread.Text = "Yes"
    Else
        lblAutoEventIsPoolThread.Text = "No"
    End If
```

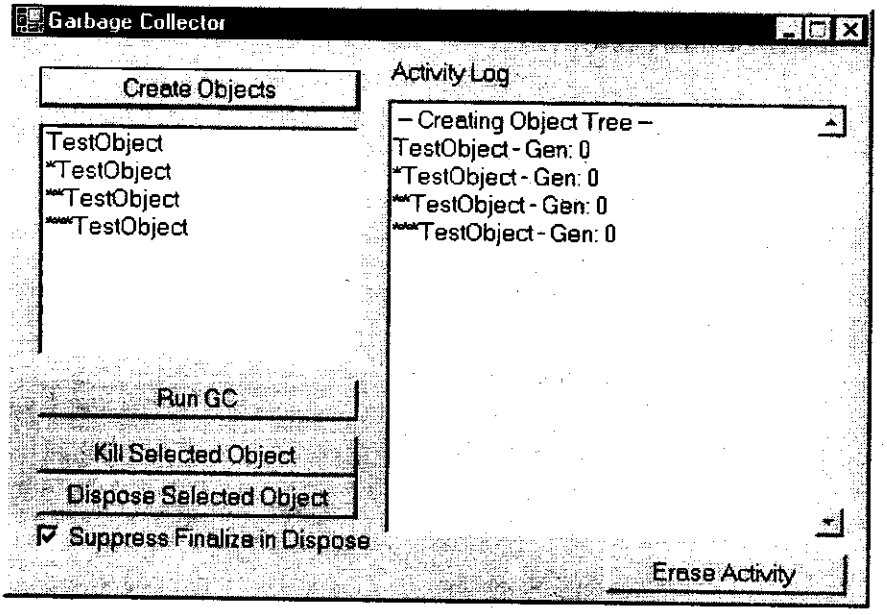
```
        ' Đưa thread hiện hành vào trạng thái đợi 2 giây
        Threading.Thread.CurrentThread.Sleep(2000)

        lblAutoEventStatus.Text = "Waiting"
        lblAutoEventStatus.ForeColor = Color.Blue

        btnSetAutoEvent.Enabled = True

        If Not btnWaitForManualEvent.Enabled And Not btnWaitForMutex.Enabled Then
            btnSetReleaseAll.Enabled = True
        End If
    End Sub

    ' Callback khi manual reset event được set
    Private Sub OnManualEventSet(ByVal obj As Object, ByVal TimedOut As Boolean)
        lblManualEventThreadStatus.Text = "Running"
        lblManualEventThreadStatus.ForeColor = Color.Green

        lblManualEventThreadNum.Text = _
            Threading.Thread.CurrentThread.GetHashCode().ToString

        If Threading.Thread.CurrentThread.IsThreadPoolThread Then
            lblManualEventIsPoolThread.Text = "Yes"
        Else
            lblManualEventIsPoolThread.Text = "No"
        End If

        Threading.Thread.CurrentThread.Sleep(2000)

        lblManualEventThreadStatus.Text = "Inactive"
        lblManualEventThreadStatus.ForeColor = Color.Red

        manualResetEvent1.Close()

        btnWaitForManualEvent.Enabled = True

    End Sub
```

```
    ' Callback khi mutex được giải phóng
    Private Sub OnMutexReleased(ByVal obj As Object, ByVal TimedOut As Boolean)
        lblMutexThreadStatus.Text = "Running"
        lblMutexThreadStatus.ForeColor = Color.Green

        lblMutexThreadNum.Text = _
            Threading.Thread.CurrentThread.GetHashCode().ToString

        If Threading.Thread.CurrentThread.IsThreadPoolThread Then
            lblMutexIsPoolThread.Text = "Yes"
        Else
            lblMutexIsPoolThread.Text = "No"
        End If

        Threading.Thread.CurrentThread.Sleep(2000)

        lblMutexThreadStatus.Text = "Inactive"
        lblMutexThreadStatus.ForeColor = Color.Red

        mutex1.Close()

        btnWaitForMutex.Enabled = True
    End Sub

    ' Thủ tục này được gọi khi cả ba process đã kết thúc
    Private Sub OnProcessesCompleted()
        ' Lấy thời gian các process đã chạy và hiển thị trên label
        Dim secondsElapsed As Double = ProcessGroup.GetTicksElapsed / 1000
        lblSecondsElapsed.Text = secondsElapsed.ToString

        btnNonthreaded.Enabled = True
        btnThreaded.Enabled = True
        btnThreadPool.Enabled = True

        ' Reset các process để chạy lần khác
        ProcessGroup.PrepareToRun()
    End Sub
End Class
```

```
    ' Callback khi mutex được giải phóng
    Private Sub OnMutexReleased(ByVal obj As Object, ByVal TimedOut As Boolean)
        lblMutexThreadStatus.Text = "Running"
        lblMutexThreadStatus.ForeColor = Color.Green

        lblMutexThreadNum.Text = _
            Threading.Thread.CurrentThread.GetHashCode().ToString

        If Threading.Thread.CurrentThread.IsThreadPoolThread Then
            lblMutexIsPoolThread.Text = "Yes"
        Else
            lblMutexIsPoolThread.Text = "No"
        End If

        Threading.Thread.CurrentThread.Sleep(2000)

        lblMutexThreadStatus.Text = "Inactive"
        lblMutexThreadStatus.ForeColor = Color.Red

        mutex1.Close()

        btnWaitForMutex.Enabled = True
    End Sub

    ' Thủ tục này được gọi khi cả ba process đã kết thúc
    Private Sub OnProcessesCompleted()
        ' Lấy thời gian các process đã chạy và hiển thị trên label
        Dim secondsElapsed As Double = ProcessGroup.GetTicksElapsed / 1000
        lblSecondsElapsed.Text = secondsElapsed.ToString

        btnNonthreaded.Enabled = True
        btnThreaded.Enabled = True
        btnThreadPool.Enabled = True

        ' Reset các process để chạy lần khác
        ProcessGroup.PrepareToRun()
    End Sub
End Class
```

84. Tìm hiểu Garbage Collector

Chương trình này chỉ cách điều khiển cách hoạt động của Garbage Collector. Thông báo khi garbage collector gọi bất kỳ phương thức Finalize trên các đối tượng đã được tạo.



Các chức năng chính:

- Sử dụng phương thức GC.Collect() để ép garbage collector chạy.
- Tăng hiệu xuất của garbage collector trên các đối tượng cụ thể bằng cách sử dụng phương thức GC.SuppressFinalize().
- Cách đánh dấu một đối tượng để garbage collection theo dõi.

Project: **Garbage Collector**

GcTest.vb

```
Option Explicit On
' Xây dựng class để sử dụng thử Gargage Collector
Public Class GcTest

    ' Khai báo hai biến private, một để lưu tên của đối tượng hiện hành và một
    ' để chứa đối tượng GcTest con.
    Private m_Name As String
    Private m_Child As GcTest
```

```
' Các sự kiện phải được khai báo tương ứng với một event handler
Public Delegate Sub InfoEventHandler(ByVal message As String)
Public Event ObjectGcInfo As InfoEventHandler
Public Sub New(ByVal name As String, ByVal numLevels As Integer)

    m_Name = name

    ' Chỉ có các đối tượng được khai báo với WithEvents sẽ nhận được sự kiện này
    RaiseEvent ObjectGcInfo(m_Name + " Created")
    ' Bảo đảm độ sâu của cây không quá sâu, giới hạn tới 5
    If numLevels > 5 Then
        numLevels = 5
    End If

    If numLevels > 0 Then
        m_Child = New GcTest("*" + m_Name, numLevels - 1)
        AddHandler m_Child.ObjectGcInfo, AddressOf myChildHandler
    End If
End Sub

' Trả về vị trí (cấp) hiện hành của GC
Public ReadOnly Property Generation() As Integer
    Get
        ' GetGeneration trả về cấp của đối tượng hiện hành
        Return GC.GetGeneration(Me)
    End Get
End Property

Public ReadOnly Property Name() As String
    Get
        ' Trả về tên của đối tượng hiện hành
        Return m_Name
    End Get

End Property
```

```
' Khai báo phương thức Dispose
Sub Dispose(Optional ByVal isFinalizeSupressed As Boolean = True)
    ' Nếu có yêu cầu SupressFinalize, báo cho Garbage Collector bỏ qua
    ' Finalize của đối tượng này. Tăng hiệu xuất của garbage collector.
  If isFinalizeSupressed Then
    GC.SuppressFinalize(Me)
  End If

  If Not m_Child Is Nothing Then
    m_Child.Dispose(isFinalizeSupressed)
    m_Child = Nothing
  End If

  RaiseEvent ObjectGcInfo(m_Name + " Disposed (" + _
    isFinalizeSupressed.ToString() + ")")
End Sub

' Hàm này Dispose đối tượng có tên là tham số được truyền vào. Nó duyệt đệ quy
' qua toàn bộ cây. Trả về True nếu đối tượng đã được Dispose.
Public Function DisposeChildNamed(ByVal name As String, _
  ByVal isFinalizeSupressed As Boolean) As Boolean

  ' Kiểm tra m_child
  If Not m_Child Is Nothing Then
    If m_Child.Name = name Then
      m_Child.Dispose(isFinalizeSupressed)
      Return True
    Else
      ' Tìm đối tượng để Dispose bằng cách duyệt qua cây
      m_Child.DisposeChildNamed(name, isFinalizeSupressed)
    End If
  Else
    ' Không tìm thấy đối tượng
    Return False
  End If
End Function
```

```
    ' Hàm này trả về một ArrayList chứa các cấp của tất cả đối tượng được tạo
    ' và con của chúng.
    Public Function GetSelfAndChildGenerations() As ArrayList
        ' Tạo một ArrayList để sử dụng
        Dim myArrayList As ArrayList
        ' Nếu tồn tại đối tượng con, gọi hàm đệ quy để lấy danh sách con
        If Not m_Child Is Nothing Then
            ' Lấy ArrayList từ đối tượng con
            myArrayList = m_Child.GetSelfAndChildGenerations()
        Else
            ' Không có đối tượng con, tạo một ArrayList mới
            myArrayList = New ArrayList()
        End If

        ' Cho tên của đối tượng hiện hành vào ArrayList ở vị trí đầu tiên
        myArrayList.Insert(0, m_Name + " - Gen: " + Me.Generation.ToString())

        Return myArrayList
    End Function

    ' Hàm này trả về một ArrayList chứa các tên của tất cả các đối tượng được tạo
    ' và con của chúng.
    Public Function GetSelfAndChildNames() As ArrayList

        Dim myArrayList As ArrayList
        If Not m_Child Is Nothing Then
            myArrayList = m_Child.GetSelfAndChildNames()
        Else
            myArrayList = New ArrayList()
        End If
        myArrayList.Insert(0, m_Name)
        Return myArrayList
    End Function

    ' Override phương thức Finalize để phát sinh thông tin
    Protected Overrides Sub Finalize()
        ' Để cho đối tượng cha của đối tượng này biết là Finalize được gọi
        RaiseEvent ObjectGcInfo(m_Name + " Finalized")
    End Sub
```

```
    ' Hàm này huỷ đối tượng có tên là tham số truyền vào
    Public Function KillChildNamed(ByVal name As String) As Boolean

        If Not m_Child Is Nothing Then
            If m_Child.Name = name Then
                ' Hủy m_Child
                m_Child = Nothing
                Return True
            Else
                m_Child.KillChildNamed(name)
            End If
        Else
            Return False
        End If
    End Function

    ' Cài event handler cho đối tượng con
    Public Sub myChildHandler(ByVal message As String)
        RaiseEvent ObjectGcInfo(message)
    End Sub

    ' Override phương thức ToString để trả về tên của đối tượng
    Public Overrides Function ToString() As String
        Return m_Name
    End Function
End Class
```

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form
    ' Khai báo một biến GcTest
    Private m_TestObject As GcTest
    Private Const OBJECT_DEPTH As Integer = 3

' Code phát sinh form được bỏ qua
```

```
Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClear.Click
    txtActivity.Clear()
End Sub

' Thủ tục này tạo một cây đối tượng chứa OBJECT_DEPTH + 1 đối tượng
Private Sub btnCreateObjects_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCreateObjects.Click


    ' Chỉ cho phép tồn tại một cây tại một thời điểm
    If m_TestObject Is Nothing Then
        Me.AddToActivityLog(" -- Creating Object Tree -- ")

        ' Tạo đối tượng Test sẽ tự động tạo cây đối tượng
        m_TestObject = New GcTest("TestObject", OBJECT_DEPTH)

        AddHandler m_TestObject.ObjectGcInfo, AddressOf myObjectEventHandler

        Dim myObject As Object
        For Each myObject In m_TestObject.GetSelfAndChildGenerations()
            AddToActivityLog(myObject.ToString())
        Next

        ' Hiển thị cây đối tượng
        ShowTestObjects()

    Else
        MsgBox("Please Kill existing objects before creating new objects.", _
            MsgBoxStyle.OKOnly, Me.Text)
    End If
End Sub

' Thủ tục này dispose đối tượng được chọn trong lstCreatedObjects
Private Sub btnDisposeObject_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDisposeObject.Click

    If Me.lstCreatedObjects.SelectedItem Is Nothing Then
        Return
    End If
```

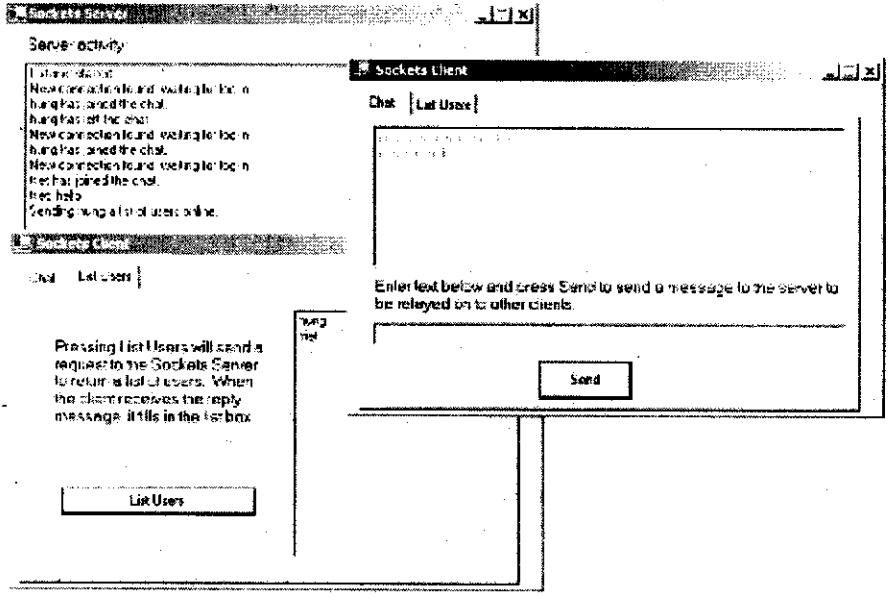
```
    ' Lấy tên của đối tượng được chọn
    Dim strName As String = Me.lstCreatedObjects.SelectedItem.ToString()

    AddToActivityLog(" -- Disposing of " + strName + " -- ")

    If strName = Me.m_TestObject.Name Then
      m_TestObject.Dispose(chkIsFinalizeSuppressed.Checked)
      m_TestObject = Nothing
    Else
      m_TestObject.DisposeChildNamed(strName, _
          chkIsFinalizeSuppressed.Checked)
    End If

    ShowTestObjects()
  End Sub

  ' Thủ tục này huỷ đối tượng được chọn trong lstCreatedObjects
  Private Sub btnKillObject_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnKillObject.Click

    If Me.lstCreatedObjects.SelectedItem Is Nothing Then
      Return
    End If

    Dim strName As String = Me.lstCreatedObjects.SelectedItem.ToString()
    Me.AddToActivityLog(" -- Killing " + strName + " -- ")

    If strName = Me.m_TestObject.Name Then
      m_TestObject = Nothing
    Else
      m_TestObject.KillChildNamed(strName)
    End If

    ShowTestObjects()

  End Sub
```

```
' Thủ tục này ép Garbage Collector (GC) chạy
Private Sub btnRunGC_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRunGC.Click
    Me.lstCreatedObjects.Items.Clear()
    Me.lstCreatedObjects.Refresh()
    Me.AddToActivityLog(" -- Running GC --")

    GC.Collect()

    GC.WaitForPendingFinalizers()

    If Not m_TestObject Is Nothing Then

        Dim myObject As Object
        For Each myObject In m_TestObject.GetSelfAndChildGenerations()
            Me.AddToActivityLog(myObject.ToString())
        Next
    End If
    ShowTestObjects()
End Sub

Private Sub AddToActivityLog(ByVal message As String)
    Try
        Me.txtActivity.AppendText(message + vbCrLf)
        Me.txtActivity.ScrollToCaret()
    Catch ode As ObjectDisposedException
        MsgBox(message, MsgBoxStyle.OKOnly, Me.Text)
    Catch aoor As System.ArgumentOutOfRangeException
        MsgBox(message, MsgBoxStyle.OKOnly, Me.Text)
    Catch ex As Exception
        MsgBox("Exception Occurred. " + ex.ToString(), _
            MsgBoxStyle.OKOnly, Me.Text)
    End Try
End Sub

' Thủ tục này xử lý tất cả các sự kiện do các đối tượng GcTest phát sinh
Private Sub myObjectEventHandler(ByVal message As String)
    AddToActivityLog(message)
End Sub
```

```
    Private Sub ShowTestObjects()
        Me.lstCreatedObjects.Items.Clear()
        If Not Me.m_TestObject Is Nothing Then
            Dim tmpObject As Object
            For Each tmpObject In m_TestObject.GetSelfAndChildNames()
                Me.lstCreatedObjects.Items.Add(tmpObject)
            Next
        End If
    End Sub
End Class
```

85. Lập trình Socket

Chương trình này chỉ cách sử dụng Sockets trong Visual Basic .NET.



Các chức năng chính:

Chương trình này sử dụng các class Socket .NET TcpListener và TcpClient và StreamWriter để xây dựng một Chat Server cho phép nhiều Client kết nối vào đồng thời. Ta sẽ tìm hiểu ba hướng quan trọng của lập trình socket:

- **Trao đổi giữa Server và Client.** Ứng dụng Server cho phép nhập vào một chuỗi và gửi cho tất cả các client có kết nối vào. Để thử chức năng này ta phải chạy Server và có ít nhất một client kết nối vào.

- **Trao đổi giữa Client và Client**. Gửi văn bản từ một client đến tất cả các client còn lại có kết nối đến server. Cần chạy server và ít nhất hai client để thử tính năng này.
- **Trao đổi giữa Client và Server** (Client yêu cầu và Server trả lời). Client gửi đến server một yêu cầu cần danh sách của tất cả người sử dụng đang kết nối vào Server. Khi Client nhận được trả lời, nó đưa danh sách này vào ListBox.

Solution: **Sockets**

Project: **Sockets Server**

UserConnection.vb

```
Option Strict On

Imports System.Net.Sockets
Imports System.Text

' Class UserConnection gom chức năng của một kết nối TcpClient cho một user
Public Class UserConnection
    Const READ_BUFFER_SIZE As Integer = 255

    ' Overload New để thiết lập một tiểu trình đọc
    Public Sub New(ByVal client As TcpClient)
        Me.client = client
        ' Lệnh này Start tiểu trình đọc. Dữ liệu sẽ được lưu vào readBuffer
        Me.client.GetStream.BeginRead(readBuffer, 0, READ_BUFFER_SIZE, _
                                      AddressOf StreamReceiver, Nothing)
    End Sub

    Private client As TcpClient
    Private readBuffer(READ_BUFFER_SIZE) As Byte
    Private strName As String

    ' Thuộc tính Name định danh kết nối cho mỗi user
    Public Property Name() As String
        Get
            Return strName
        End Get
```

```
    Set(ByVal Value As String)
      strName = Value
    End Set
  End Property

  Public Event LineReceived(ByVal sender As UserConnection, _
                                                ByVal Data As String)
  ' Thủ tục này sử dụng một StreamWriter để gửi một message cho user
  Public Sub SendData(ByVal Data As String)
    ' SyncLock bảo đảm là không có tiểu trình khác cố sử dụng stream này
    ' vào cùng thời điểm.
    SyncLock client.GetStream
      Dim writer As New IO.StreamWriter(client.GetStream)
      writer.Write(Data & Chr(13) & Chr(10))
        ' Bảo đảm là tất cả dữ liệu đã được gửi
      writer.Flush()
    End SyncLock
  End Sub

  ' Đây là hàm callback cho TcpClient.GetStream.Begin.
  Private Sub StreamReceiver(ByVal ar As IAsyncResult)
    Dim BytesRead As Integer
    Dim strMessage As String

    Try
      SyncLock client.GetStream
          ' Kết thúc việc đọc bất đồng bộ vào readBuffer và lấy số byte đọc được
        BytesRead = client.GetStream.EndRead(ar)
      End SyncLock

        ' Chuyển mảng byte nhận được thành message, trừ đi một ký tự (Chr(13))
      strMessage = Encoding.ASCII.GetString(readBuffer, 0, BytesRead - 1)
      RaiseEvent LineReceived(Me, strMessage)

      SyncLock client.GetStream
          ' Bắt đầu một hành động đọc khác
        client.GetStream.BeginRead(readBuffer, 0, READ_BUFFER_SIZE, _
                          AddressOf StreamReceiver, Nothing)
      End SyncLock
```

```
    Catch e As Exception
    End Try
  End Sub
End Class
```

frmMain.vb

```
Option Strict On
Imports System.Net.Sockets
Public Class frmMain
  Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

  Const PORT_NUM As Integer = 10000
  Private clients As New Hashtable()
  Private listener As TcpListener
  Private listenerThread As Threading.Thread

  ' Thủ tục này gửi một message đến tất cả các client có kết nối vào server
  Private Sub Broadcast(ByVal strMessage As String)
    Dim client As UserConnection
    Dim entry As DictionaryEntry

    ' Tất cả các phần tử trong HashTable là UserConnection
    For Each entry In clients
      client = CType(entry.Value, UserConnection)
      client.SendData(strMessage)
    Next
  End Sub

  ' Thủ tục này gửi nội dung của TextBox Broadcast đến tất cả client
  Private Sub btnBroadcast_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBroadcast.Click
    If txtBroadcast.Text <> "" Then
      UpdateStatus("Broadcasting: " & txtBroadcast.Text)
      Broadcast("BROADI" & txtBroadcast.Text)
      txtBroadcast.Text = ""
    End If
  End Sub
```

```
' Thủ tục này kiểm tra một user đã có trong HashTable hay chưa. Nếu có thì gửi
' một message REFUSE, nếu chưa có thì xác nhận là JOIN
Private Sub ConnectUser(ByVal userName As String, ByVal sender As
UserConnection)
    If clients.Contains(userName) Then
        ReplyToSender("REFUSE", sender)
    Else
        sender.Name = userName
        UpdateStatus(userName & " has joined the chat.")
        clients.Add(userName, sender)

        ' Gửi thông báo JOIN đến sender, và thông báo cho tất cả các client khác
        SendToClients("CHAT!" & sender.Name & " has joined the chat.", sender)
    End If
End Sub

' Thủ tục này thông báo cho các client khác biết sender đã rời phòng chat
Private Sub DisconnectUser(ByVal sender As UserConnection)
    UpdateStatus(sender.Name & " has left the chat.")
    SendToClients("CHAT!" & sender.Name & " has left the chat.", sender)
    clients.Remove(sender.Name)
End Sub

' Thủ tục này được sử dụng làm tiểu trình lắng nghe background để cho phép
' đọc các message đến mà không ảnh hưởng đến giao diện người sử dụng
Private Sub DoListen()
    Try
        ' Lắng nghe các kết nối mới
        listener = New TcpListener(PORT_NUM)
        listener.Start()
        Do
            ' Tạo một kết nối user mới sử dụng TcpClient được trả về từ
            ' TcpListener.AcceptTcpClient()
            Dim client As New UserConnection(listener.AcceptTcpClient)
            ' Tạo một event handler để cho phép UserConnection trao đổi với cửa sổ
            AddHandler client.LineReceived, AddressOf OnLineReceived
            UpdateStatus("New connection found: waiting for log-in")
        Loop Until False
```

```
    Catch
    End Try
End Sub

' Khi cửa sổ đóng, ngừng việc lắng nghe
Private Sub frmMain_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    listener.Stop()
End Sub

' Start tiểu trình lắng nghe
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    listenerThread = New Threading.Thread(AddressOf DoListen)
    listenerThread.Start()
    UpdateStatus("Listener started")
End Sub

' Nối tên tất cả user và gửi cho user có yêu cầu danh sách
Private Sub ListUsers(ByVal sender As UserConnection)
    Dim client As UserConnection
    Dim entry As DictionaryEntry
    Dim strUserList As String

    UpdateStatus("Sending " & sender.Name & " a list of users online.")

    strUserList = "LISTUSERS"

    For Each entry In clients
        client = CType(entry.Value, UserConnection)
        strUserList = strUserList & "|" & client.Name
    Next

    ' Gửi danh sách cho sender
    ReplyToSender(strUserList, sender)
End Sub
```

```
' Xử lý khi UserConnection nhận một dòng đầy đủ. Phân tích dữ liệu nhận được
Private Sub OnLineReceived(ByVal sender As UserConnection, ByVal data As
String)
    Dim dataArray() As String

    ' Các phần của message được phân tách bằng dấu "|"
    dataArray = data.Split(Chr(124))

    ' dataArray(0) là lệnh
    Select Case dataArray(0)
        Case "CONNECT"
            ConnectUser(dataArray(1), sender)
        Case "CHAT"
            SendChat(dataArray(1), sender)
        Case "DISCONNECT"
            DisconnectUser(sender)
        Case "REQUESTUSERS"
            ListUsers(sender)
        Case Else
            UpdateStatus("Unknown message:" & data)
    End Select
End Sub

' Thủ tục này gửi một trả lời đến sender
Private Sub ReplyToSender(ByVal strMessage As String, ByVal sender As
UserConnection)
    sender.SendData(strMessage)
End Sub

' Gửi một chat message cho tất cả client ngoại trừ sender
Private Sub SendChat(ByVal message As String, ByVal sender As
UserConnection)

    UpdateStatus(sender.Name & ": " & message)
    SendToClients("CHAT|" & sender.Name & ": " & message, sender)
End Sub
```

```
    ' Thủ tục này gửi một message đến tất cả client có kết nối ngoại trừ sender
    Private Sub SendToClients(ByVal strMessage As String, ByVal sender As
    UserConnection)
        Dim client As UserConnection
        Dim entry As DictionaryEntry

        For Each entry In clients
            client = CType(entry.Value, UserConnection)

            ' Không gửi cho người gửi message.
            If client.Name <> sender.Name Then
                client.SendData(strMessage)
            End If
        Next
    End Sub

    ' Thêm một dòng vào listbox Status
    Private Sub UpdateStatus(ByVal statusMessage As String)
        lstStatus.Items.Add(statusMessage)
    End Sub
End Class
```

Project: **Sockets Client**

frmMain.vb

```
Option Strict On

Imports System.Net.Sockets
Imports System.Text
Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Const READ_BUFFER_SIZE As Integer = 255
    Const PORT_NUM As Integer = 10000

    Private client As TcpClient
    Private readBuffer(READ_BUFFER_SIZE) As Byte
```

```
' Mở hộp thoại Connect User và gửi một message yêu cầu tham gia chat
Sub AttemptLogin()
    Dim frmConnectUser As New frmConnectUser()
    frmConnectUser.StartPosition = FormStartPosition.CenterParent
    frmConnectUser.ShowDialog(Me)
    SendData("CONNECT|" & frmConnectUser.txtUserLogin.Text)
    frmConnectUser.Dispose()
End Sub

' Xóa ListBox User và yêu cầu server gửi danh sách các user
Private Sub btnListUsers_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnListUsers.Click
    lstUsers.Items.Clear()
    SendData("REQUESTUSERS")
End Sub

' Gửi nội dung trong textbox Send
Private Sub btnSend_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSend.Click
    If txtSend.Text <> "" Then
        DisplayText("You say: " & txtSend.Text & Chr(13) & Chr(10))
        SendData("CHAT|" & txtSend.Text)
        txtSend.Text = ""
    End If
End Sub

' Ghi thêm văn bản vào TextBox output
Private Sub DisplayText(ByVal text As String)
    txtDisplay.AppendText(text)
End Sub

' Đây là hàm callback cho TcpClient.GetStream.Begin để đọc dữ liệu
Private Sub DoRead(ByVal ar As IAsyncResult)
    Dim BytesRead As Integer
    Dim strMessage As String

    Try
        ' Kết thúc việc đọc và trả về số byte đọc được
        BytesRead = client.GetStream.EndRead(ar)
```

```
        If BytesRead < 1 Then
            ' Nếu không đọc được byte nào do server đã đóng. Disable cửa sổ input
            MarkAsDisconnected()
            Exit Sub
        End If
        ' Chuyển mảng byte sang message, trừ hai ký tự Chr(13) và Chr(10)
        strMessage = Encoding.ASCII.GetString(readBuffer, 0, BytesRead - 2)
        ProcessCommands(strMessage)
        ' Start một tiểu trình đọc khác
        client.GetStream.BeginRead(readBuffer, 0, READ_BUFFER_SIZE, _
                                   AddressOf DoRead, Nothing)
    Catch e As Exception
        MarkAsDisconnected()
    End Try
End Sub

' Gửi đến server một messge thông báo cắt kết nối
Private Sub frmMain_Closing(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    ' Chỉ gửi khi server còn hoạt động
    If btnSend.Enabled = True Then
        SendData("DISCONNECT")
    End If
End Sub

' Khi chạy chương trình, thủ tục này kết nối vào server và tiến hành log in
Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim frmConnectUser As New frmConnectUser()

    Try
        client = New TcpClient("localhost", PORT_NUM)

        client.GetStream.BeginRead(readBuffer, 0, READ_BUFFER_SIZE, _
                                   AddressOf DoRead, Nothing)
        ' Bảo đảm cửa sổ chính hiển thị trước khi hiển thị hộp thoại kết nối

        Me.Show()
        AttemptLogin()
```

```
    Catch Ex As Exception
      MsgBox("Server is not active.  Please start server and try again.", _
          MsgBoxStyle.Exclamation, Me.Text)
      Me.Dispose()
    End Try
  End Sub

  ' Cho danh sách các user vào listbox
  Private Sub ListUsers(ByVal users() As String)
    Dim I As Integer
    For I = 1 To users.Length - 1
      lstUsers.Items.Add(users(I))
    Next
  End Sub

  ' Khi Server cắt kết nối, không cho gửi các message khác
  Private Sub MarkAsDisconnected()
    txtSend.ReadOnly = True
    btnSend.Enabled = False
  End Sub

  ' Xử lý lệnh nhận từ server và có hành động thích hợp
  Private Sub ProcessCommands(ByVal strMessage As String)
    Dim dataArray() As String
    ' Các phần của message cách nhau bởi dấu "|"
    dataArray = strMessage.Split(Chr(124))

    ' dataArray(0) là lệnh
    Select Case dataArray(0)
      Case "JOIN"
          ' Server báo cho biết là đã login
        DisplayText("You have joined the chat" & Chr(13) & Chr(10))
      Case "CHAT"
          ' Nhận message chat, hiển thị nó
        DisplayText(dataArray(1) & Chr(13) & Chr(10))
      Case "REFUSE"
          ' Server từ chối đăng nhập do tên user, login với tên khác
        AttemptLogin()
```

```
        Case "LISTUSERS"
            ' Server gửi danh sách các user
            ListUsers(dataArray)
        Case "BROAD"
            ' Server gửi một message broadcast (gửi cho tất cả client)
            DisplayText("ServerMessage: " & dataArray(1) & Chr(13) & Chr(10))
    End Select
  End Sub

  ' Sử dụng StreamWriter để gửi một message đến server
  Private Sub SendData(ByVal data As String)
    Dim writer As New IO.StreamWriter(client.GetStream)
    writer.Write(data & vbCr)
    writer.Flush()
  End Sub
End Class
```

frmConnectUser.vb

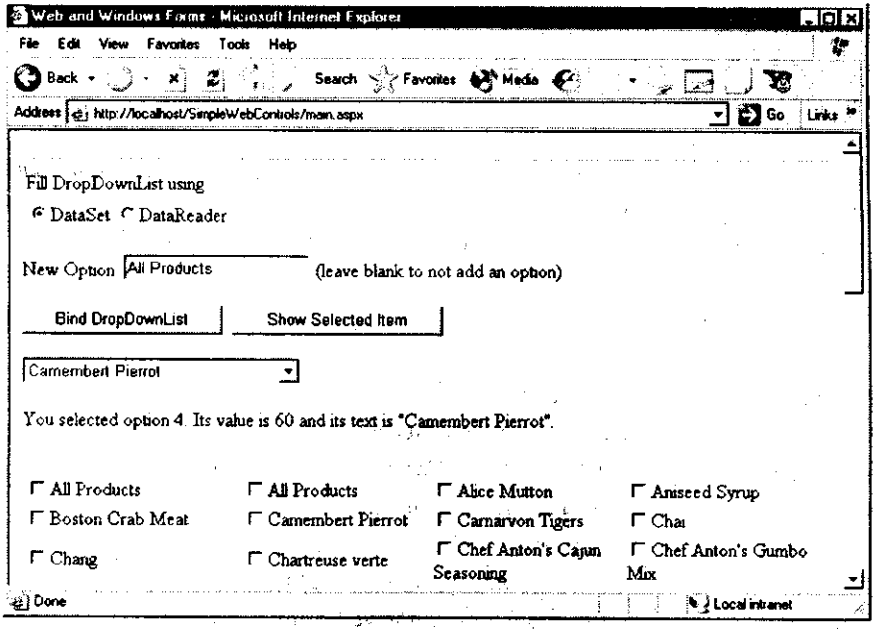
```
Option Strict On
Public Class frmConnectUser
  Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "
  Public Sub New()
    MyBase.New()
    'This call is required by the Windows Form Designer.
    InitializeComponent()
    'Add any initialization after the InitializeComponent() call
  End Sub

  'Form overrides dispose to clean up the component list.
  Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
    If disposing Then
      If Not (components Is Nothing) Then
        components.Dispose()
      End If
    End If
    MyBase.Dispose(disposing)
  End Sub
```

```
    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents btnLogin As System.Windows.Forms.Button
    Friend WithEvents txtUserLogin As System.Windows.Forms.TextBox
    Friend WithEvents lblInstructions As System.Windows.Forms.Label
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.btnLogin = New System.Windows.Forms.Button()
        Me.txtUserLogin = New System.Windows.Forms.TextBox()
        Me.lblInstructions = New System.Windows.Forms.Label()
        Me.SuspendLayout()
        '
        'btnLogin
        '
        Me.btnLogin.DialogResult = System.Windows.Forms.DialogResult.Cancel
        Me.btnLogin.Location = New System.Drawing.Point(75, 150)
        Me.btnLogin.Name = "btnLogin"
        Me.btnLogin.Size = New System.Drawing.Size(141, 46)
        Me.btnLogin.TabIndex = 0
        Me.btnLogin.Text = "&Login"
        '
        'txtUserLogin
        '
        Me.txtUserLogin.Location = New System.Drawing.Point(18, 103)
        Me.txtUserLogin.Name = "txtUserLogin"
        Me.txtUserLogin.Size = New System.Drawing.Size(262, 20)
        Me.txtUserLogin.TabIndex = 1
        Me.txtUserLogin.Text = ""
        '
        'lblInstructions
        '
        Me.lblInstructions.Font = New System.Drawing.Font("Microsoft Sans Serif",
    9.75!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
    CType(0, Byte))
        Me.lblInstructions.ForeColor = System.Drawing.Color.FromArgb(CType(0, Byte),
    CType(0, Byte), CType(192, Byte))
```

```
        Me.lblInstructions.Location = New System.Drawing.Point(18, 19)
        Me.lblInstructions.Name = "lblInstructions"
        Me.lblInstructions.Size = New System.Drawing.Size(282, 75)
        Me.lblInstructions.TabIndex = 2
        Me.lblInstructions.Text = "Please enter a unique name to log into the chat utility.
    If login name already e" & _
        "xists you will be returned to this dialog to attempt another name."
        '
        'frmConnectUser
        '
        Me.AcceptButton = Me.btnLogin
        Me.AccessibleDescription = "Connect User Form"
        Me.AccessibleName = "Connect User Form"
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.CancelButton = Me.btnLogin
        Me.ClientSize = New System.Drawing.Size(317, 211)
        Me.Controls.AddRange(New System.Windows.Forms.Control()
    {Me.lblInstructions, Me.txtUserLogin, Me.btnLogin})
        Me.FormBorderStyle =
    System.Windows.Forms.FormBorderStyle.FixedToolWindow
        Me.MaximizeBox = False
        Me.MinimizeBox = False
        Me.Name = "frmConnectUser"
        Me.Text = "Connect User"
        Me.ResumeLayout(False)
    End Sub
#End Region
End Class
```

86. So sánh các control dùng trên Web Forms và Windows Forms

Chương trình này cho thấy cách kết dữ liệu và truy xuất đến thông tin được chọn trên các control cho phép chọn một giá trị (single-select) và chọn nhiều giá trị (mutil-select). Sử dụng các control có công dụng gần như nhau trên Web Forms và Windows Forms, trên Windows Forms sử dụng các control ComboBox và CheckedListBox, trên Web Forms sử dụng các control DropDownList và CheckBoxList.



Các chức năng chính:

- Kết dữ liệu với DataSet và ArrayList.
- Sử dụng một phương thức Shared của Class Library hỗ trợ để thêm một chọn lựa phụ cho một control có kết dữ liệu, chọn lựa này rất hay được sử dụng trong ứng dụng thực tế: "All____" hoặc "Any____".
- Truy xuất thông tin của item được chọn (trên control chỉ cho chọn một giá trị) hoặc của các item được chọn (trên control cho phép chọn nhiều giá trị).

Chú ý:

- Chương trình này sử dụng database Northwind của SQL Server cục bộ hoặc MSDE.
- Solution này có ba project: Helper, Simple Windows Forms Controls, SimpleWebControls. Nhấp đúp chuột vào file setup.vbs để cài đặt và chép project SimpleWebControls vào IIS.

Solution: **So sanh DataBinding giua Web Forms va Windows Forms**

Project: **Helper**

Helper.vb

```
Option Strict On
```

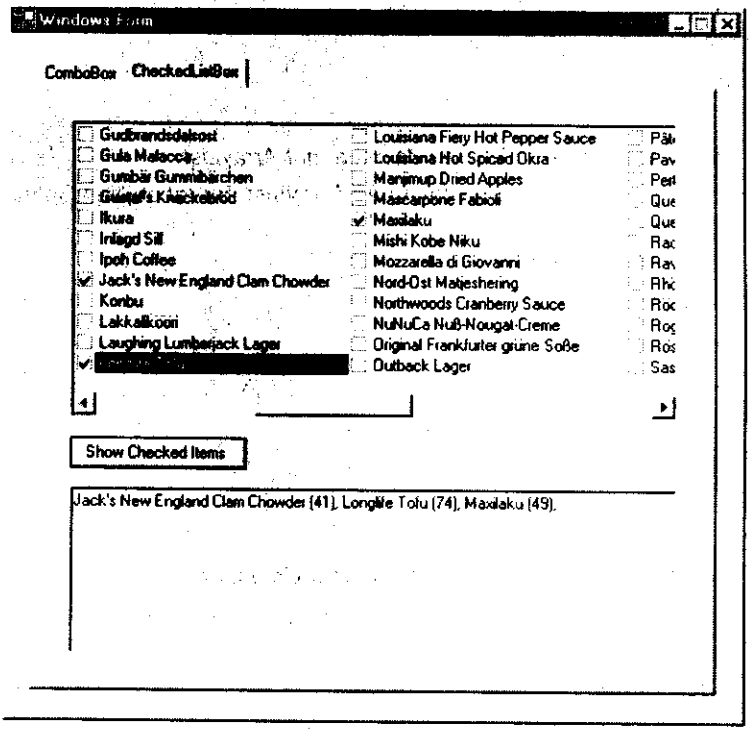
```
Public Class UI
    ' Hàm này thêm một chọn lựa khác vào dòng đầu tiên của DataTable.
    Shared Function AddOption(ByVal ds As DataSet, ByVal strDisplayField As String,
    ByVal strValueField As String, ByRef strNewOptionText As String, ByVal
    strNewOptionValue As String) As DataSet
        Dim dt As DataTable
        Dim drNew As DataRow
        dt = ds.Tables(0)

        drNew = dt.NewRow()
        drNew(strDisplayField) = strNewOptionText
        drNew(strValueField) = strNewOptionValue
        dt.Rows.InsertAt(drNew, 0)
        dt.AcceptChanges()

        Return ds
    End Function

    ' Hàm này thêm một chọn lựa vào đầu của một ArrayList
    Shared Function AddOption(ByVal arl As ArrayList, ByVal objNewOption As Object)
    As ArrayList
        arl.Insert(0, objNewOption)
        Return arl
    End Function
End Class
```

Product.vb

```
Option Strict On

' Class Product sử dụng cho kết dữ liệu vào một ComboBox.
' Các thuộc tính phải là ReadOnly để có thể kết vào ComboBox
Public Class Product
    Dim _ID As Integer
    Dim _Name As String

    Sub New(ByVal intID As Integer, ByVal strName As String)
        _ID = intID
        _Name = strName
    End Sub
```

```
    Public ReadOnly Property ID() As Integer
        Get
            Return _ID
        End Get
    End Property

    Public ReadOnly Property Name() As String
        Get
            Return _Name
        End Get
    End Property
End Class
```

Project: **Simple Windows Forms Controls**



frmMain.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.Text
```

```
Public Class frmMain
    Inherits System.Windows.Forms.Form

    Protected Const CONNECTION_ERROR_MSG As String = _
    "To run this sample, you must have SQL or MSDE."

    Protected Const MSDE_CONNECTION_STRING As String = _
        "Server=(local)\NetSDK;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"

    Protected Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"
    Private arlProducts As ArrayList
    Private dsProducts As DataSet
    Private FormHasLoaded As Boolean = False
    Private sdrProducts As SqlDataReader
    Private strConn As String

' Code phát sinh form được bỏ qua

    Private Sub btnBindComboBox_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnBindComboBox.Click
        lblSelected.Text = ""

        If optDataReader.Checked Then
            BindComboBoxUsingDataReader()
        Else
            BindComboBoxUsingDataSet()
        End If
    End Sub

    ' Hiển thị tất cả các item được chọn trên CheckedListBox
    Private Sub btnShowChecked_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnShowCheckedItems.Click

        Dim sb As New StringBuilder()
```

```
        ' DataRowView là kiểu của các đối tượng được lưu trong CheckedItemCollection
        ' trong thuộc tính CheckedItems
        Dim drwvItem As DataRowView

        For Each drwvItem In clstProducts.CheckedItems
            sb.Append(drwvItem("Name"))
            sb.Append(" (")
            sb.Append(drwvItem("ID"))
            sb.Append("), ")
        Next
        txtChecked.Text = sb.ToString

    End Sub


    ' Hiển thị index, văn bản, và giá trị của item được chọn trong ComboBox
    Private Sub btnShowSelectedItem_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnShowSelectedItem.Click
        With cboProducts
            lblSelected.Text = "You selected option " & .SelectedIndex.ToString & _
                ". Its value is " & .SelectedValue.ToString & " and its " & _
                "text is """ & .Text & """."
        End With
    End Sub

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        FormHasLoaded = True
        BindComboBoxUsingDataSet()
        BindCheckedListBoxUsingDataSet()
    End Sub

    Private Sub AddItemToDataSource(ByRef arl As ArrayList)
        With arl
            .Add(New Helper.Product(sdrProducts.GetInt32(0), _
                sdrProducts("ProductName").ToString))
        End With
    End Sub
```

```
Private Sub BindCheckedListBoxUsingDataSet()
    With clstProducts
        .DisplayMember = "Name"
        .ValueMember = "ID"

        If Trim(txtNewOption.Text) <> "" Then
            .DataSource = Helper.UI.AddOption(dsProducts, .DisplayMember, _
                .ValueMember, txtNewOption.Text, "0").Tables(0)
        Else
            .DataSource = dsProducts.Tables(0)
        End If
    End With
End Sub


Private Sub BindComboBoxUsingDataReader()
    strConn = SQL_CONNECTION_STRING

    Dim strSQL As String = _
        "SELECT ProductID, ProductName " & _
        "FROM Products " & _
        "ORDER BY ProductName"

    Dim scnnNorthwind As New SqlConnection(strConn)

    Dim scmd As New SqlCommand(strSQL, scnnNorthwind)
    arlProducts = New ArrayList()

    Dim frmStatusMessage As New frmStatus()

    Try
        frmStatusMessage.Show("Connecting to SQL Server to load Products " & _
            "using a DataReader")
        ' Đợi 1 giây để người sử dụng thấy các thông báo
        System.Threading.Thread.Sleep(1000)

        scnnNorthwind.Open()
        sdrProducts = scmd.ExecuteReader(CommandBehavior.CloseConnection)
```

```
        ' Duyệt qua DataReader. Phương thức Read trả về True nếu còn dòng
        ' khác để đọc
        While sdrProducts.Read
          AddItemToDataSource(arlProducts)
        End While

        With cboProducts
          .ValueMember = "ID"
          .DisplayMember = "Name"
          If Trim(txtNewOption.Text) <> "" Then
            .DataSource = Helper.UI.AddOption(arlProducts, _
              New Helper.Product(0, txtNewOption.Text))
          Else
            .DataSource = arlProducts
          End If
        End With

        sdrProducts.Close()
        frmStatusMessage.Close()

      Catch expSql As SqlException
        MsgBox(expSql.ToString, MsgBoxStyle.Critical, Me.Text)
      End Try
    End Sub

    Private Sub BindComboBoxUsingDataSet()
      strConn = SQL_CONNECTION_STRING

      Dim strSQL As String = _
        "SELECT ProductID As ID, ProductName As Name " & _
        "FROM Products " & _
        "ORDER BY ProductName"

      Dim scnnNorthwind As New SqlConnection(strConn)
      Dim scmd As New SqlCommand(strSQL, scnnNorthwind)
      Dim sda As New SqlDataAdapter(scmd)
      dsProducts = New DataSet()

      Dim frmStatusMessage As New frmStatus()
```

```
        frmStatusMessage.Show("Connecting to SQL Server to load Products " & _
          "using a DataSet")

        Dim IsConnecting As Boolean = True
        While IsConnecting
          Try
            sda.Fill(dsProducts)
            IsConnecting = False
            System.Threading.Thread.Sleep(1000)
            frmStatusMessage.Close()

          Catch expSql As SqlException
            MessageBox.Show(expSql.ToString, Me.Text, _
              MessageBoxButtons.OK, MessageBoxIcon.Error)
          End Try
        End While

        With cboProducts
          .DisplayMember = "Name"
          .ValueMember = "ID"

        ' Không giống với DropDownList WebControl, ta phải truy xuất đến collection
        ' Tables của DataSet để kết dữ liệu, cho dù chỉ có một DataTable trong DataSet.
          If Trim(txtNewOption.Text) <> "" Then
            .DataSource = Helper.UI.AddOption(dsProducts, .DisplayMember, _
              .ValueMember, txtNewOption.Text, "0").Tables(0)
          Else
            .DataSource = dsProducts.Tables(0)
          End If
        End With
      End Sub
    End Class
```

Project: **SimpleWebControls**

main.aspx

```
<%@Page Language="vb" AutoEventWireup="false" Codebehind="main.aspx.vb"
  Inherits="SimpleWebControls.Main"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<HTML>
  <HEAD>
    <title>Web and Windows Forms</title>
    <meta content="Microsoft Visual Studio.NET 7.0" name="GENERATOR">
    <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
  name="vs_targetSchema">
  </HEAD>
  <body>
    <form id="frmMain" method="post" runat="server">
      <HR width="100%" SIZE="1">
      <table>
        <tr>
          <td>Fill DropDownList using:</td>
        </tr>
      </table>
      <asp:radiobuttonlist id="optlDataSource" runat="server"
  RepeatDirection="Horizontal">
        <asp:ListItem Value="DataSet" Selected="True">DataSet</asp:ListItem>
        <asp:ListItem Value="DataReader">DataReader</asp:ListItem>
      </asp:radiobuttonlist>
      <P>New Option:
        <asp:textbox id="txtNewOption" runat="server">All
  Products</asp:textbox> (leave
        blank to not add an option)</P>
      <P><asp:button id="btnBindDropDownList" runat="server" Text="Bind
  DropDownList"></asp:button> 
        <asp:button id="btnShowSelectedItem" runat="server" Text="Show
  Selected Item"></asp:button></P>
      <P><asp:dropdownlist id="cboProducts"
  runat="server"></asp:dropdownlist></P>
      <P><asp:label id="lblSelected" runat="server"></asp:label></P>
      <P>
        <HR width="100%" SIZE="1">
        <asp:CheckBoxList id="clstProducts" runat="server"
  RepeatDirection="Horizontal" RepeatColumns="4"></asp:CheckBoxList>
      <P></P>
      <P>
        <asp:Button id="btnShowCheckedItems" runat="server" Text="Show
  Checked Items"></asp:Button></P>
```

```
        <P>
            <asp:Label id="lblChecked" runat="server"></asp:Label></P>
        </form>
    </body>
</HTML>
```

main.aspx.vb

```
Option Strict On
Imports System.Data.SqlClient
Imports System.Text

Public Class Main
    Inherits System.Web.UI.Page
    Protected WithEvents lblTitle, lblChecked, lblSelected As Label
    Protected WithEvents cboProducts As DropDownList
    Protected WithEvents txtNewOption As TextBox
    Protected WithEvents btnBindDropDownList, btnShowSelectedItem,
    btnShowCheckedItems As Button
    Protected WithEvents clstProducts As CheckBoxList
    Protected WithEvents optlDataSource As RadioButtonList

#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()

    End Sub

    Private Sub Page_Init(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()
    End Sub

#End Region

    Private arlProducts As ArrayList
    Private dsProducts As DataSet
    Private sdrProducts As SqlDataReader
```

```
Private strConn As String

Protected Const MSDE_CONNECTION_STRING As String = _
   "Server=(local)\NetSDK;" & _
   "DataBase=Northwind;" & _
   "Integrated Security=false;user id=sa;password="

Protected Const SQL_CONNECTION_STRING As String = _
   "Server=localhost;" & _
   "DataBase=Northwind;" & _
   "Integrated Security=false;user id=sa;password="

Private Sub btnBindDropDownList_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnBindDropDownList.Click
   lblSelected.Text = ""
   If optlDataSource.SelectedIndex = 0 Then
      BindDropDowListUsingDataSet()
   Else
      BindDropDowListUsingDataReader()
   End If
End Sub

' Hiển thị danh sách các item được chọn trong CheckBoxList
Private Sub btnShowChecked_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShowCheckedItems.Click
   Dim item As ListItem
   Dim sb As New StringBuilder()
   For Each item In clstProducts.Items
      If item.Selected Then
         sb.Append(item.Text)
         sb.Append(" (")
         sb.Append(item.Value)
         sb.Append("), ")
      End If
   Next
   lblChecked.Text = sb.ToString

End Sub
```

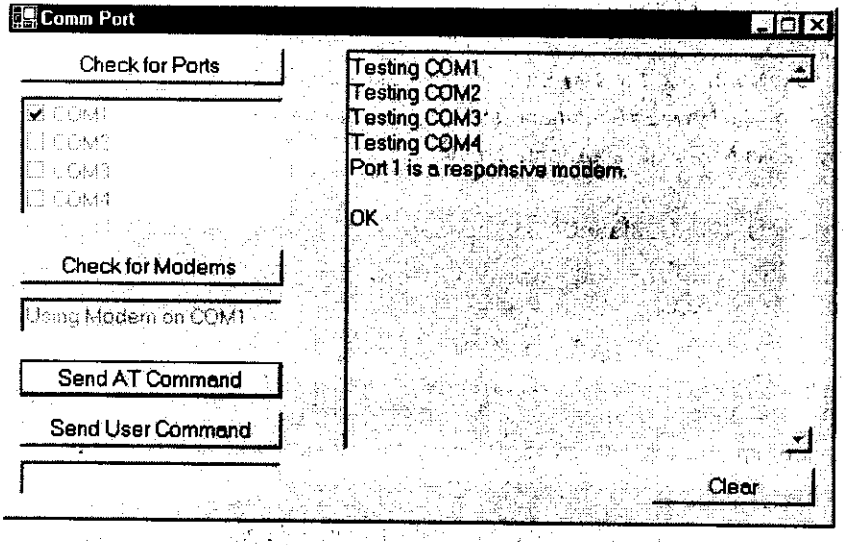
```
' Hiển thị index, văn bản, và giá trị của item được chọn trên DropDownList
Private Sub btnShowSelectedItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btnShowSelectedItem.Click
    With cboProducts
        lblSelected.Text = "You selected option " & .SelectedIndex.ToString & _
            ". Its value is " & .SelectedItem.Value & " and its " & _
            "text is """ & .SelectedItem.Text & """."
    End With
End Sub

Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    If Not Page.IsPostBack Then
        BindDropDowListUsingDataSet()
        BindCheckBoxListUsingDataSet()
    End If
End Sub

Private Sub AddItemToDataSource(ByRef arl As ArrayList)
    With arl
        .Add(New Helper.Product(sdrProducts.GetInt32(0), _
            sdrProducts("ProductName").ToString))
    End With
End Sub

Private Sub BindCheckBoxListUsingDataSet()
    With clstProducts
        .DataTextField = "Name"
        .DataValueField = "ID"

        If Trim(txtNewOption.Text) <> "" Then
            .DataSource = Helper.UI.AddOption(dsProducts, .DataTextField, _
                .DataValueField, txtNewOption.Text, "0")
        Else
            .DataSource = dsProducts
        End If

        .DataBind()
    End With
End Sub
```

```
Private Sub BindDropDowListUsingDataReader()

    strConn = SQL_CONNECTION_STRING

    Dim strSQL As String = _
      "SELECT ProductID, ProductName " & _
      "FROM Products"

    Dim scnnNorthwind As New SqlConnection(strConn)
    Dim scmd As New SqlCommand(strSQL, scnnNorthwind)
    arlProducts = New ArrayList()

    scnnNorthwind.Open()
    sdrProducts = scmd.ExecuteReader(CommandBehavior.CloseConnection)

    While sdrProducts.Read
      AddItemToDataSource(arlProducts)
    End While

    With cboProducts
      .DataValueField = "ID"
      .DataTextField = "Name"

      If Trim(txtNewOption.Text) <> "" Then
        .DataSource = Helper.UI.AddOption(arlProducts, _
            New Helper.Product(0, txtNewOption.Text))
      Else
        .DataSource = arlProducts
      End If

      ' Ta phải gọi DataBind() khi sử dụng các WebControl
      .DataBind()
    End With

    sdrProducts.Close()

End Sub
```

```
Private Sub BindDropDowListUsingDataSet()
    strConn = SQL_CONNECTION_STRING
    Dim strSQL As String = _
      "SELECT ProductID As ID, ProductName As Name " & _
      "FROM Products " & _
      "ORDER BY ProductName"

    Dim scnnNorthwind As New SqlConnection(strConn)
    Dim scmd As New SqlCommand(strSQL, scnnNorthwind)
    Dim sda As New SqlDataAdapter(scmd)

    dsProducts = New DataSet()

    sda.Fill(dsProducts)

    With cboProducts
      .DataTextField = "Name"
      .DataValueField = "ID"

       ' Không cần truy xuất đến bảng cụ thể khi DataSet chỉ chứa một bảng
      If Trim(txtNewOption.Text) <> "" Then
        .DataSource = Helper.UI.AddOption(dsProducts, .DataTextField, _
          .DataValueField, txtNewOption.Text, "0")
      Else
        .DataSource = dsProducts
      End If
       ' Phải gọi phương thức DataBind()
      .DataBind()
    End With
  End Sub
End Class
```

87. Sử dụng cổng COM

Chương trình này chỉ cách điều khiển cổng COM trong .NET. Bởi vì .NET Framework không hỗ trợ điều khiển trực tiếp Comm Port, do đó ta sử dụng một class khác có hỗ trợ gọi các hàm Win32 API trực tiếp để điều khiển nó.



Các chức năng chính:

- Kiểm tra một cổng comm có sẵn sàng để sử dụng.
- Sử dụng Comm Port để trao đổi với modem.
- Sử dụng các hàm Win32 API để điều khiển việc trao đổi với Comm Port (máy tính cần có một modem).

Project: **Comm Port**

Rs232.vb

```
Option Strict On
Imports System.Runtime.InteropServices
Imports System.Text
Imports System.Threading
```

```
' Class này cung cấp mọi thứ cần thiết để trao đổi với Comm Port
Public Class Rs232
    ' Khai báo các biến class và khởi tạo giá trị
    Private mhRS As Integer = -1   ' Handle của Com Port
    Private miPort As Integer = 1   ' Mặc nhiên là COM1
    Private miTimeout As Integer = 70   ' Timeout tính bằng mili giây
    Private miBaudRate As Integer = 9600
    Private meParity As DataParity = 0
    Private meStopBit As DataStopBit = 0
    Private miDataBit As Integer = 8
    Private miBufferSize As Integer = 512   ' Kích thước Buffer mặc nhiên là 512 byte
    Private mabtRxBuf As Byte()   ' Buffer nhận
    Private meMode As Mode   ' Chế độ làm việc của class
    Private mbWaitOnRead As Boolean
    Private mbWaitOnWrite As Boolean
    Private mbWriteErr As Boolean
    Private muOverlapped As OVERLAPPED
    Private muOverlappedW As OVERLAPPED
    Private muOverlappedE As OVERLAPPED
    Private mabtTmpTxBuf As Byte()   ' buffer tạm sử dụng bởi Async Tx
    Private moThreadTx As Thread
    Private moThreadRx As Thread
    Private miTmpBytes2Read As Integer
    Private meMask As EventMasks

#Region "Enums"

  Public Enum DataParity
      Parity_None = 0
      Pariti_Odd
      Parity_Even
      Parity_Mark
    End Enum

    Public Enum DataStopBit
      StopBit_1 = 1
      StopBit_2
    End Enum
```

```
Private Enum PurgeBuffers
    RXAbort = &H2
    RXClear = &H8
    TxAbort = &H1
    TxClear = &H4
End Enum

Private Enum Lines
    SetRts = 3
    ClearRts = 4
    SetDtr = 5
    ClearDtr = 6
    ResetDev = 7
    SetBreak = 8
    ClearBreak = 9
End Enum

' Enum này cung cấp các giá trị trạng thái của Modem, Sử dụng Flag() cho phép
' sử dụng phép toán bit của các giá trị
<Flags()> Public Enum ModemStatusBits
    ClearToSendOn = &H10
    DataSetReadyOn = &H20
    RingIndicatorOn = &H40
    CarrierDetect = &H80
End Enum

' Chế độ làm việc
Public Enum Mode
    NonOverlapped
    Overlapped
End Enum

<Flags()> Public Enum EventMasks
    RxChar = &H1
    RXFlag = &H2
    TxBufferEmpty = &H4
    ClearToSend = &H8
    DataSetReady = &H10
    ReceiveLine = &H20
```

```
        Break = &H40
        StatusError = &H80
        Ring = &H100
    End Enum
#End Region

#Region "Structures"
    ' Các cấu trúc này sử dụng bởi các hàm Windows API
    <StructLayout(LayoutKind.Sequential, Pack:=1)> Private Structure DCB
        Public DCBlength As Integer
        Public BaudRate As Integer
        Public Bits1 As Integer
        Public wReserved As Int16
        Public XonLim As Int16
        Public XoffLim As Int16
        Public ByteSize As Byte
        Public Parity As Byte
        Public StopBits As Byte
        Public XonChar As Byte
        Public XoffChar As Byte
        Public ErrorChar As Byte
        Public EofChar As Byte
        Public EvtChar As Byte
        Public wReserved2 As Int16
    End Structure

    <StructLayout(LayoutKind.Sequential, Pack:=1)> Private Structure
    COMMTIMEOUTS
        Public ReadIntervalTimeout As Integer
        Public ReadTotalTimeoutMultiplier As Integer
        Public ReadTotalTimeoutConstant As Integer
        Public WriteTotalTimeoutMultiplier As Integer
        Public WriteTotalTimeoutConstant As Integer
    End Structure

    <StructLayout(LayoutKind.Sequential, Pack:=1)> Private Structure COMMCONFIG
        Public dwSize As Integer
        Public wVersion As Int16
        Public wReserved As Int16
        Public dcbx As DCB
```

```
        Public dwProviderSubType As Integer
        Public dwProviderOffset As Integer
        Public dwProviderSize As Integer
        Public wcProviderData As Byte
    End Structure

    <StructLayout(LayoutKind.Sequential, Pack:=1)> Public Structure OVERLAPPED
        Public Internal As Integer
        Public InternalHigh As Integer
        Public Offset As Integer
        Public OffsetHigh As Integer
        Public hEvent As Integer
    End Structure

#End Region

#Region "Exceptions"

    ' Class này định nghĩa một Exception sẽ phát sinh khi NACK được phát sinh
    Public Class CIOChannelException : Inherits ApplicationException
        Sub New(ByVal Message As String)
            MyBase.New(Message)
        End Sub
        Sub New(ByVal Message As String, ByVal InnerException As Exception)
            MyBase.New(Message, InnerException)
        End Sub
    End Class

    ' Định nghĩa timeout exception
    Public Class IOTimeoutException : Inherits CIOChannelException
        Sub New(ByVal Message As String)
            MyBase.New(Message)
        End Sub
        Sub New(ByVal Message As String, ByVal InnerException As Exception)
            MyBase.New(Message, InnerException)
        End Sub
    End Class

#End Region
```

```
#Region "Events"
    ' Các sự kiện này cho phép chương trình sử dụng class này để tương tác với các
    ' sự kiện của Comm Port
    Public Event DataReceived(ByVal Source As Rs232, ByVal DataBuffer() As Byte)
    Public Event TxCompleted(ByVal Source As Rs232)
    Public Event CommEvent(ByVal Source As Rs232, ByVal Mask As EventMasks)
#End Region

#Region "Constants"
    Private Const PURGE_RXABORT As Integer = &H2
    Private Const PURGE_RXCLEAR As Integer = &H8
    Private Const PURGE_TXABORT As Integer = &H1
    Private Const PURGE_TXCLEAR As Integer = &H4
    Private Const GENERIC_READ As Integer = &H80000000
    Private Const GENERIC_WRITE As Integer = &H40000000
    Private Const OPEN_EXISTING As Integer = 3
    Private Const INVALID_HANDLE_VALUE As Integer = -1
    Private Const IO_BUFFER_SIZE As Integer = 1024
    Private Const FILE_FLAG_OVERLAPPED As Integer = &H40000000
    Private Const ERROR_IO_PENDING As Integer = 997
    Private Const WAIT_OBJECT_0 As Integer = 0
    Private Const ERROR_IO_INCOMPLETE As Integer = 996
    Private Const WAIT_TIMEOUT As Integer = &H102&
    Private Const INFINITE As Integer = &HFFFFFFFF
#End Region

#Region "Properties"
    ' Thuộc tính để nhận hoặc quy định BaudRate
    Public Property BaudRate() As Integer
        Get
            Return miBaudRate
        End Get
        Set(ByVal Value As Integer)
            miBaudRate = Value
        End Set
    End Property
```

```
' Thuộc tính để nhận hoặc quy định BufferSize
Public Property BufferSize() As Integer
  Get
    Return miBufferSize
  End Get
  Set(ByVal Value As Integer)
    miBufferSize = Value
  End Set
End Property

' Thuộc tính để nhận hoặc quy định DataBit
Public Property DataBit() As Integer
  Get
    Return miDataBit
  End Get
  Set(ByVal Value As Integer)
    miDataBit = Value
  End Set
End Property

' Thuộc tính chỉ ghi, dùng để set hoặc reset DTR
Public WriteOnly Property Dtr() As Boolean
  Set(ByVal Value As Boolean)
    If Not mhRS = -1 Then
      If Value Then
        EscapeCommFunction(mhRS, Lines.SetDtr)
      Else
        EscapeCommFunction(mhRS, Lines.ClearDtr)
      End If
    End If
  End Set
End Property

' Thuộc tính chỉ đọc, trả về một mảng các byte dữ liệu đến
Overridable ReadOnly Property InputStream() As Byte()
  Get
    Return mabtRxBuf
  End Get
End Property
```

```
' Trả về chuỗi các byte dữ liệu đến
Overridable ReadOnly Property InputStreamString() As String
  Get
    Dim oEncoder As New System.Text.ASCIIEncoding()
    Return oEncoder.GetString(Me.InputStream)
  End Get
End Property

' Trả về trạng thái (Open) của Comm Port
ReadOnly Property IsOpen() As Boolean
  Get
    Return CBool(mhRS <> -1)
  End Get
End Property

' Trả về trạng thái của modem
Public ReadOnly Property ModemStatus() As ModemStatusBits
  Get
    If mhRS = -1 Then
      Throw New ApplicationException("Please initialize and open " + _
        "port before using this method")
    Else
      ' Truy xuất trạng thái của modem
      Dim lpModemStatus As Integer
      If Not GetCommModemStatus(mhRS, lpModemStatus) Then
        Throw New ApplicationException("Unable to get modem status")
      Else
        Return CType(lpModemStatus, ModemStatusBits)
      End If
    End If
  End Get
End Property

' Thuộc tính này nhận hoặc quy định Parity
Public Property Parity() As DataParity
  Get
    Return meParity
  End Get
```

```
    Set(ByVal Value As DataParity)
      meParity = Value
    End Set
  End Property

  ' Thuộc tính này nhận hoặc quy định Port
  Public Property Port() As Integer
    Get
      Return miPort
    End Get
    Set(ByVal Value As Integer)
      miPort = Value
    End Set
  End Property

  ' Set hoặc reset RTS
  Public WriteOnly Property Rts() As Boolean
    Set(ByVal Value As Boolean)
      If Not mhRS = -1 Then
        If Value Then
          EscapeCommFunction(mhRS, Lines.SetRts)
        Else
          EscapeCommFunction(mhRS, Lines.ClearRts)
        End If
      End If
    End Set
  End Property

  ' Thuộc tính nhận hoặc quy định StopBit
  Public Property StopBit() As DataStopBit
    Get
      Return meStopBit
    End Get
    Set(ByVal Value As DataStopBit)
      meStopBit = Value
    End Set

  End Property
```

```
' Trả về chuỗi các byte dữ liệu đến
Overridable ReadOnly Property InputStreamString() As String
  Get
    Dim oEncoder As New System.Text.ASCIIEncoding()
    Return oEncoder.GetString(Me.InputStream)
  End Get
End Property

' Trả về trạng thái (Open) của Comm Port
ReadOnly Property IsOpen() As Boolean
  Get
    Return CBool(mhRS <> -1)
  End Get
End Property

' Trả về trạng thái của modem
Public ReadOnly Property ModemStatus() As ModemStatusBits
  Get
    If mhRS = -1 Then
      Throw New ApplicationException("Please initialize and open " + _
        "port before using this method")
    Else
      ' Truy xuất trạng thái của modem
      Dim lpModemStatus As Integer
      If Not GetCommModemStatus(mhRS, lpModemStatus) Then
        Throw New ApplicationException("Unable to get modem status")
      Else
        Return CType(lpModemStatus, ModemStatusBits)
      End If
    End If
  End Get
End Property

' Thuộc tính này nhận hoặc quy định Parity
Public Property Parity() As DataParity
  Get
    Return meParity
  End Get
```

```
    Set(ByVal Value As DataParity)
      meParity = Value
    End Set
  End Property

  ' Thuộc tính này nhận hoặc quy định Port
  Public Property Port() As Integer
    Get
      Return miPort
    End Get
    Set(ByVal Value As Integer)
      miPort = Value
    End Set
  End Property

  ' Set hoặc reset RTS
  Public WriteOnly Property Rts() As Boolean
    Set(ByVal Value As Boolean)
      If Not mhRS = -1 Then
        If Value Then
          EscapeCommFunction(mhRS, Lines.SetRts)
        Else
          EscapeCommFunction(mhRS, Lines.ClearRts)
        End If
      End If
    End Set
  End Property

  ' Thuộc tính nhận hoặc quy định StopBit
  Public Property StopBit() As DataStopBit
    Get
      Return meStopBit
    End Get
    Set(ByVal Value As DataStopBit)
      meStopBit = Value
    End Set

  End Property
```

```
    ' Timeout
    Public Overridable Property Timeout() As Integer
      Get
        Return miTimeout
      End Get
      Set(ByVal Value As Integer)
        miTimeout = CInt(IIf(Value = 0, 500, Value))
        pSetTimeout()
      End Set
    End Property

    Public Property WorkingMode() As Mode
      Get
        Return meMode
      End Get
      Set(ByVal Value As Mode)
        meMode = Value
      End Set
    End Property

#End Region

#Region "Win32API"

    ' Các hàm sau là các hàm Win32 API cần thiết để thực hiện trao đổi với cổng COM
    <DllImport("kernel32.dll")> Private Shared Function BuildCommDCB( _
      ByVal lpDef As String, ByRef lpDCB As DCB) As Integer
    End Function

    <DllImport("kernel32.dll")> Private Shared Function ClearCommError( _
      ByVal hFile As Integer, ByVal lpErrors As Integer, _
      ByVal l As Integer) As Integer
    End Function

    <DllImport("kernel32.dll")> Private Shared Function CloseHandle( _
      ByVal hObject As Integer) As Integer
    End Function
```

```
<DllImport("kernel32.dll")> Private Shared Function CreateEvent( _
  ByVal lpEventAttributes As Integer, ByVal bManualReset As Integer, _
  ByVal bInitialState As Integer, _
  <MarshalAs(UnmanagedType.LPStr)> ByVal lpName As String) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function CreateFile( _
  <MarshalAs(UnmanagedType.LPStr)> ByVal lpFileName As String, _
  ByVal dwDesiredAccess As Integer, ByVal dwShareMode As Integer, _
  ByVal lpSecurityAttributes As Integer, _
  ByVal dwCreationDisposition As Integer, _
  ByVal dwFlagsAndAttributes As Integer, _
  ByVal hTemplateFile As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function EscapeCommFunction( _
  ByVal hFile As Integer, ByVal ifunc As Long) As Boolean
End Function

<DllImport("kernel32.dll")> Private Shared Function FormatMessage( _
  ByVal dwFlags As Integer, ByVal lpSource As Integer, _
  ByVal dwMessageId As Integer, ByVal dwLanguageId As Integer, _
  <MarshalAs(UnmanagedType.LPStr)> ByVal lpBuffer As String, _
  ByVal nSize As Integer, ByVal Arguments As Integer) As Integer
End Function

Private Declare Function FormatMessage Lib "kernel32" Alias _
 "FormatMessageA" (ByVal dwFlags As Integer, ByVal lpSource As Integer, _
 ByVal dwMessageId As Integer, ByVal dwLanguageId As Integer, _
 ByVal lpBuffer As StringBuilder, ByVal nSize As Integer, _
 ByVal Arguments As Integer) As Integer

<DllImport("kernel32.dll")> Public Shared Function GetCommModemStatus( _
  ByVal hFile As Integer, ByRef lpModemStatus As Integer) As Boolean
End Function

<DllImport("kernel32.dll")> Private Shared Function GetCommState( _
  ByVal hCommDev As Integer, ByRef lpDCB As DCB) As Integer
End Function
```

```
<DllImport("kernel32.dll")> Private Shared Function GetCommTimeouts( _
  ByVal hFile As Integer, ByRef lpCommTimeouts As COMMTIMEOUTS) As
Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function GetLastError() As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function GetOverlappedResult( _
  ByVal hFile As Integer, ByRef lpOverlapped As OVERLAPPED, _
  ByRef lpNumberOfBytesTransferred As Integer, _
  ByVal bWait As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function PurgeComm( _
  ByVal hFile As Integer, ByVal dwFlags As Integer) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function ReadFile( _
  ByVal hFile As Integer, ByVal Buffer As Byte(), _
  ByVal nNumberOfBytesToRead As Integer, _
  ByRef lpNumberOfBytesRead As Integer, _
  ByRef lpOverlapped As OVERLAPPED) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function SetCommTimeouts( _
  ByVal hFile As Integer, ByRef lpCommTimeouts As COMMTIMEOUTS) As
Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function SetCommState( _
  ByVal hCommDev As Integer, ByRef lpDCB As DCB) As Integer
End Function

<DllImport("kernel32.dll")> Private Shared Function SetupComm( _
  ByVal hFile As Integer, ByVal dwInQueue As Integer, _
  ByVal dwOutQueue As Integer) As Integer
End Function
```

```
    <DllImport("kernel32.dll")> Private Shared Function SetCommMask( _
      ByVal hFile As Integer, ByVal lpEvtMask As Integer) As Integer
    End Function

    <DllImport("kernel32.dll")> Private Shared Function WaitCommEvent( _
      ByVal hFile As Integer, ByRef Mask As EventMasks, _
      ByRef lpOverlap As OVERLAPPED) As Integer
    End Function

    <DllImport("kernel32.dll")> Private Shared Function WaitForSingleObject( _
      ByVal hHandle As Integer, ByVal dwMilliseconds As Integer) As Integer
    End Function

    <DllImport("kernel32.dll")> Private Shared Function WriteFile( _
      ByVal hFile As Integer, ByVal Buffer As Byte(), _
      ByVal nNumberOfBytesToWrite As Integer, _
      ByRef lpNumberOfBytesWritten As Integer, _
      ByRef lpOverlapped As OVERLAPPED) As Integer
    End Function

#End Region

#Region "Methods"


    ' Thủ tục này thực hiện đọc không đồng bộ, thủ tục này không sử dụng trực
    ' tiếp nhưng được sử dụng cho class.
    Public Sub _R()
      Dim iRet As Integer = Read(miTmpBytes2Read)
    End Sub

    ' Gửi ra cổng COM
    Public Sub _W()
      Write(mabtTmpTxBuf)
    End Sub
```

```
' Thủ tục này sử dụng một thread khác để đọc từ cổng COM. Nó phát
' sinh RxCompleted khi đọc xong. Nó đọc một số nguyên
Public Overloads Sub AsyncRead(ByVal Bytes2Read As Integer)
  If meMode <> Mode.Overlapped Then Throw New ApplicationException( _
    "Async Methods allowed only when WorkingMode=Overlapped")
  miTmpBytes2Read = Bytes2Read
  moThreadTx = New Thread(AddressOf _R)
  moThreadTx.Start()
End Sub

' Thủ tục này sử dụng một tiểu trình khác để ghi ra cổng COM. Nó phát sinh
' TxCompleted khi hoàn tất. Nó ghi một mảng các Byte
Public Overloads Sub AsyncWrite(ByVal Buffer() As Byte)
  If meMode <> Mode.Overlapped Then Throw New ApplicationException( _
    "Async Methods allowed only when WorkingMode=Overlapped")
  If mbWaitOnWrite = True Then Throw New ApplicationException( _
    "Unable to send message because of pending transmission.")
  mabtTmpTxBuf = Buffer
  moThreadTx = New Thread(AddressOf _W)
  moThreadTx.Start()
End Sub

' Ghi một chuỗi ra cổng COM
Public Overloads Sub AsyncWrite(ByVal Buffer As String)
  Dim oEncoder As New System.Text.ASCIIEncoding()
  Dim aByte() As Byte = oEncoder.GetBytes(Buffer)
  Me.AsyncWrite(aByte)
End Sub

' Kiểm tra Modem có hoạt động không
Public Function CheckLineStatus(ByVal Line As ModemStatusBits) As Boolean
  Return Convert.ToBoolean(ModemStatus And Line)
End Function
' Xóa nội dung của buffer
Public Sub ClearInputBuffer()
  If Not mhRS = -1 Then
    PurgeComm(mhRS, PURGE_RXCLEAR)
  End If
End Sub
```

```
' Đóng cổng COM
Public Sub Close()
  If mhRS <> -1 Then
    CloseHandle(mhRS)
    mhRS = -1
  End If
End Sub

' Mở và khởi tạo Comm Port
Public Overloads Sub Open()
  Dim uDcb As DCB, iRc As Integer
  ' Quy định chế độ làm việc
  Dim iMode As Integer = Convert.ToInt32(IIf(meMode = Mode.Overlapped, _
    FILE_FLAG_OVERLAPPED, 0))
  ' Khởi tạo cổng COM
  If miPort > 0 Then
    Try
      ' Tạo một Stream Handle trên cổng COM
      mhRS = CreateFile("COM" & miPort.ToString, _
      GENERIC_READ Or GENERIC_WRITE, 0, 0, _
      OPEN_EXISTING, iMode, 0)
      If mhRS <> -1 Then
        ' Xóa tất cả các lỗi truyền thông
        Dim lpErrCode As Integer
        iRc = ClearCommError(mhRS, lpErrCode, 0&)
        ' Xóa các buffer nhập/xuất
        iRc = PurgeComm(mhRS, PurgeBuffers.RXClear Or _
          PurgeBuffers.TxClear)
        ' Lấy thông số của cổng COM
        iRc = GetCommState(mhRS, uDcb)
        ' Cập nhật thông số cổng COM
        Dim sParity As String = "NOEM"
        sParity = sParity.Substring(meParity, 1)
        ' Quy định DCB
        Dim sDCBState As String = String.Format( _
          "baud={0} parity={1} data={2} stop={3}", _
          miBaudRate, sParity, miDataBit, CInt(meStopBit))
        iRc = BuildCommDCB(sDCBState, uDcb)
```

```
            iRc = SetCommState(mhRS, uDcb)
            If iRc = 0 Then
               Dim sErrTxt As String = pErr2Text(GetLastError())
               Throw New CIOChannelException( _
                  "Unable to set COM state0" & sErrTxt)
            End If
               ' Thiết lập các buffer
            iRc = SetupComm(mhRS, miBufferSize, miBufferSize)
               ' Quy định timeout
            pSetTimeout()
         Else
               ' Phát sinh lỗi khởi tạo
            Throw New CIOChannelException( _
               "Unable to open COM" & miPort.ToString)
         End If
      Catch Ex As Exception
         Throw New CIOChannelException(Ex.Message, Ex)
      End Try
   Else
      ' Không thể mở
      Throw New ApplicationException("COM Port not defined, " + _
         "use Port property to set it before invoking InitPort")
   End If
End Sub

' Overload thủ tục Open, cho phép sử dụng tham số
Public Overloads Sub Open(ByVal Port As Integer, _
   ByVal BaudRate As Integer, ByVal DataBit As Integer, _
   ByVal Parity As DataParity, ByVal StopBit As DataStopBit, _
   ByVal BufferSize As Integer)

   Me.Port = Port
   Me.BaudRate = BaudRate
   Me.DataBit = DataBit
   Me.Parity = Parity
   Me.StopBit = StopBit
   Me.BufferSize = BufferSize
   Open()
End Sub
```

```
    ' Chuyển các mã lỗi API thành văn bản
    Private Function pErr2Text(ByVal lCode As Integer) As String
        Dim sRtrnCode As New StringBuilder(256)
        Dim lRet As Integer

        lRet = FormatMessage(&H1000, 0, lCode, 0, sRtrnCode, 256, 0)
        If lRet > 0 Then
            Return sRtrnCode.ToString
        Else
            Return "Error not found."
        End If

    End Function

    Private Sub pHandleOverlappedRead(ByVal Bytes2Read As Integer)

        Dim iReadChars, iRc, iRes, iLastErr As Integer
        muOverlapped.hEvent = CreateEvent(Nothing, 1, 0, Nothing)

        If muOverlapped.hEvent = 0 Then
            ' Không thể tạo event
            Throw New ApplicationException( _
              "Error creating event for overlapped read.")
        Else
            ' Đọc theo chế độ overlapped
            If mbWaitOnRead = False Then
                ReDim mabtRxBuf(Bytes2Read - 1)
                iRc = ReadFile(mhRS, mabtRxBuf, Bytes2Read, _
                  iReadChars, muOverlapped)
                If iRc = 0 Then
                    iLastErr = GetLastError()
                    If iLastErr <> ERROR_IO_PENDING Then
                        Throw New ArgumentException("Overlapped Read Error: " & _
                          pErr2Text(iLastErr))
                    Else
                        ' Set Flag
                        mbWaitOnRead = True
                    End If

                Else
```

```
            ' Việc đọc hoàn tất
            RaiseEvent DataReceived(Me, mabtRxBuf)
          End If
        End If
      End If

      ' Đợi cho thao tác đọc hoàn tất
      If mbWaitOnRead Then
        iRes = WaitForSingleObject(muOverlapped.hEvent, miTimeout)
        Select Case iRes
          Case WAIT_OBJECT_0
            If GetOverlappedResult(mhRS, muOverlapped, _
              iReadChars, 0) = 0 Then

              iLastErr = GetLastError()
              If iLastErr = ERROR_IO_INCOMPLETE Then
                Throw New ApplicationException( _
                  "Read operation incomplete")
              Else
                Throw New ApplicationException( _
                  "Read opetation error " & iLastErr.ToString)
              End If
            Else
              ' Thao tác đã hoàn tất
              RaiseEvent DataReceived(Me, mabtRxBuf)
              mbWaitOnRead = False
            End If
          Case WAIT_TIMEOUT
            Throw New IOTimeoutException("Timeout error")
          Case Else
            Throw New ApplicationException("Overlapped read error")
        End Select
      End If
    End Sub

    Private Function pHandleOverlappedWrite(ByVal Buffer() As Byte) As Boolean

      Dim iBytesWritten, iRc, iLastErr, iRes As Integer, bErr As Boolean
      muOverlappedW.hEvent = CreateEvent(Nothing, 1, 0, Nothing)
```

```
    If muOverlappedW.hEvent = 0 Then
      Throw New ApplicationException( _
        "Error creating event for overlapped write.")
    Else

      ' Overlap write
      PurgeComm(mhRS, PURGE_RXCLEAR Or PURGE_TXCLEAR)
      mbWaitOnRead = True
      iRc = WriteFile(mhRS, Buffer, Buffer.Length, _
        iBytesWritten, muOverlappedW)
      If iRc = 0 Then
        iLastErr = GetLastError()
        If iLastErr <> ERROR_IO_PENDING Then
          Throw New ArgumentException("Overlapped Read Error: " & _
            pErr2Text(iLastErr))
        Else
          ' Đợi
          iRes = WaitForSingleObject(muOverlappedW.hEvent, INFINITE)
          Select Case iRes
            Case WAIT_OBJECT_0
              If GetOverlappedResult(mhRS, muOverlappedW, _
                iBytesWritten, 0) = 0 Then

                bErr = True
              Else
                mbWaitOnRead = False
                RaiseEvent TxCompleted(Me)
              End If
          End Select
        End If
      Else
        bErr = False
      End If
    End If
    CloseHandle(muOverlappedW.hEvent)
    Return bErr
  End Function
```

```
' Quy định timeout cho cổng COM
Private Sub pSetTimeout()
  Dim uCtm As COMMTIMEOUTS

  If mhRS = -1 Then
    Exit Sub
  Else
    With uCtm
      .ReadIntervalTimeout = 0
      .ReadTotalTimeoutMultiplier = 0
      .ReadTotalTimeoutConstant = miTimeout
      .WriteTotalTimeoutMultiplier = 10
      .WriteTotalTimeoutConstant = 100
    End With
    SetCommTimeouts(mhRS, uCtm)
  End If
End Sub

' Hàm này trả về một số nguyên cho biết số byte đọc được từ cổng COM.
' Nó chấp nhận một tham số chỉ định số byte cần đọc.
Public Function Read(ByVal Bytes2Read As Integer) As Integer
  Dim iReadChars, iRc As Integer

  ' Nếu Bytes2Read không được chỉ định thì sử dụng Buffersize
  If Bytes2Read = 0 Then Bytes2Read = miBufferSize
  If mhRS = -1 Then
    Throw New ApplicationException( _
      "Please initialize and open port before using this method")
  Else
    ' Lấy các Byte từ cổng COM
    Try
      If meMode = Mode.Overlapped Then
        pHandleOverlappedRead(Bytes2Read)
      Else
        ReDim mabtRxBuf(Bytes2Read - 1)
        iRc = ReadFile(mhRS, mabtRxBuf, Bytes2Read, iReadChars, Nothing)
        If iRc = 0 Then
          Throw New ApplicationException( _
            "ReadFile error " & iRc.ToString)
```

```
            Else
                ' Xử lý timeout hoặc trả về các ký tự đọc được
              If iReadChars < Bytes2Read Then
                Throw New IOTimeoutException("Timeout error")
              Else
                mbWaitOnRead = True
                Return (iReadChars)
              End If
            End If
          End If
        Catch Ex As Exception
          Throw New ApplicationException("Read Error: " & Ex.Message, Ex)
        End Try
      End If
    End Function

    ' Gửi một mảng các byte ra cổng COM
    Public Overloads Sub Write(ByVal Buffer As Byte())
      Dim iBytesWritten, iRc As Integer
      If mhRS = -1 Then
        Throw New ApplicationException( _
          "Please initialize and open port before using this method")
      Else
        ' Gửi dữ liệu ra cổng COM
        Try
          If meMode = Mode.Overlapped Then
            ' Overlap write
            If pHandleOverlappedWrite(Buffer) Then
              Throw New ApplicationException( _
                "Error in overllapped write")
            End If

          Else

            ' Xóa các I/O buffer
            PurgeComm(mhRS, PURGE_RXCLEAR Or PURGE_TXCLEAR)
            iRc = WriteFile(mhRS, Buffer, Buffer.Length, _
              iBytesWritten, Nothing)
```

```
            If iRc = 0 Then
              Throw New ApplicationException( _
                "Write Error - Bytes Written " & _
                iBytesWritten.ToString & " of " & _
                Buffer.Length.ToString)
            End If
          End If
        Catch Ex As Exception
          Throw
        End Try
      End If
    End Sub

    ' Gửi một chuỗi ra cổng COM
    Public Overloads Sub Write(ByVal Buffer As String)
      Dim oEncoder As New System.Text.ASCIIEncoding()
      Dim aByte() As Byte = oEncoder.GetBytes(Buffer)
      Me.Write(aByte)
    End Sub
#End Region
End Class
```

frmMain.vb

```
Option Strict On
Imports System.Text

Public Class frmMain
    Inherits System.Windows.Forms.Form

    ' Khai báo các biến cần thiết
    Private m_CommPort As New Rs232()
    Private m_IsModemFound As Boolean = False
    Private m_ModemPort As Integer = 0

' Code phát sinh form được bỏ qua
```

```
    ' Kiểm tra các cổng COM có trên máy tính bằng cách mở các cổng từ 1 đến 4.
    Private Sub btnCheckForPorts_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCheckForPorts.Click
        ' Kiểm tra và đánh dấu vào ListBox
        Dim i As Integer
        For i = 1 To 4
            WriteMessage("Testing COM" + i.ToString())
            If IsPortAvailable(i) Then
                Me.clstPorts.SetItemChecked(i - 1, True)
            Else
                Me.clstPorts.SetItemChecked(i - 1, False)
            End If
        Next
        ' Cho phép sử dụng nút "Find Modems"
        Me.btnCheckModems.Enabled = True
    End Sub

    ' Gửi lệnh AT đến bất kỳ cổng COM nào đang hoạt động. Nếu có trả lời thì
    ' một modem có thể sử dụng được phát hiện trên cổng COM đó.
    Private Sub btnCheckModems_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnCheckModems.Click
        Dim i As Integer
        For i = 0 To 3
            If Me.clstPorts.GetItemChecked(i) Then
                If IsPortAvailable(i + 1) Then
                    ' Kiểm tra xem cổng COM có trả lời lệnh AT
                    If IsPortAModem(i + 1) Then
                        Me.m_IsModemFound = True
                        Me.m_ModemPort = i + 1
                        ' Hiển thị thông báo
                        WriteMessage("Port " + (i + 1).ToString() + _
                            " is a responsive modem.")
                    Else
                        WriteMessage("Port " + (i + 1).ToString() + _
                            " is not a responsive modem.")
                    End If
                End If
            End If
        Next
```

```
    ' Nếu tìm thấy một modem, cho phép sử dụng các nút khác
    If Me.m_IsModemFound Then
      Me.txtSelectedModem.Text = "Using Modem on COM" + _
        Me.m_ModemPort.ToString()
      Me.txtUserCommand.Enabled = True
      Me.btnSendATCommand.Enabled = True
      Me.btnSendUserCommand.Enabled = True
    End If
  End Sub

  ' Xóa nội dung TextBox
  Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnClear.Click
    Me.txtStatus.Clear()
  End Sub

  ' Gửi lệnh AT ra modem và nhận trả lời từ modem, nó phụ thuộc vào timer
  ' thực hiện việc đọc
  Private Sub btnSendATCommand_Click(ByVal sender As System.Object, ByVal e
  As System.EventArgs) Handles btnSendATCommand.Click

    Try
      ' Cho phép Timer hoạt động
      Me.tmrReadCommPort.Enabled = True
      ' Thực hiện mở port
      m_CommPort.Open(m_ModemPort, 115200, 8, _
          Rs232.DataParity.Parity_None, _
          Rs232.DataStopBit.StopBit_1, 4096)

      ' Gửi lệnh AT ra cổng COM
      m_CommPort.Write(Encoding.ASCII.GetBytes("AT" & Chr(13)))
      ' Đợi thời gian đủ lâu để modem trả lời và timer phát sinh sự kiện Tick
      System.Threading.Thread.Sleep(200)
      Application.DoEvents()
      m_CommPort.Close()

    Catch ex As Exception
      MessageBox.Show("Unable to communicate with Modem")
```

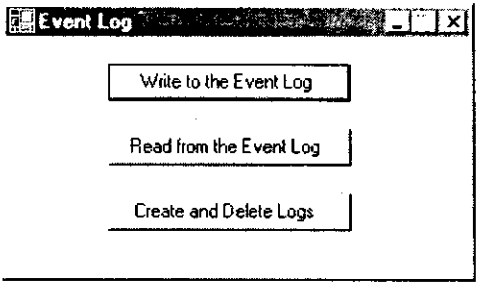
```
    Finally
        ' Không cho timer hoạt động
      Me.tmrReadCommPort.Enabled = False
    End Try
  End Sub

  ' Gửi lệnh củ người sử dụng ra modem và nhận trả lời từ nó.
  Private Sub btnSendUserCommand_Click(ByVal sender As System.Object, ByVal
  e As System.EventArgs) Handles btnSendUserCommand.Click
    Try
      Me.tmrReadCommPort.Enabled = True
      m_CommPort.Open(m_ModemPort, 115200, 8, _
          Rs232.DataParity.Parity_None, _
          Rs232.DataStopBit.StopBit_1, 4096)

        ' Gửi lệnh ra cổng COM
      m_CommPort.Write(Encoding.ASCII.GetBytes( _
                                  Me.txtUserCommand.Text & Chr(13)))
      System.Threading.Thread.Sleep(200)
      Application.DoEvents()
      m_CommPort.Close()

    Catch ex As Exception
      MessageBox.Show("Unable to communicate with Modem")
    Finally
      Me.tmrReadCommPort.Enabled = False
    End Try
  End Sub

  ' Phát sinh khi đến chu kỳ của Timer. Nó ghi tất cả dữ liệu có trong buffer
  ' của Comm Port ra TextBox.
  Private Sub tmrReadCommPort_Tick(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles tmrReadCommPort.Tick
    Try
        ' khi có dữ liệu, đọc từng byte và xuất nó ra
      While (m_CommPort.Read(1) <> -1)
        WriteMessage(Chr(m_CommPort.InputStream(0)), False)
      End While
```

```
    Catch exc As Exception
        ' Không có gì để đọc
    End Try
End Sub

' Kiểm tra cổng có gắn modem.
Private Function IsPortAModem(ByVal port As Integer) As Boolean

    Try
        m_CommPort.Open(port, 115200, 8, Rs232.DataParity.Parity_None, _
            Rs232.DataStopBit.StopBit_1, 4096)
        m_CommPort.Write(Encoding.ASCII.GetBytes("AT" & Chr(13)))
        System.Threading.Thread.Sleep(200)
        Application.DoEvents()
        ' Nhận thông tin trả về từ Comm Port
        Try
            Dim b As Byte
            ' Cố đọc một byte. Nếu đọc được thì giả sử cổng đó có modem.
            ' Xóa buffer trước khi thoát
            m_CommPort.Read(1)
            m_CommPort.ClearInputBuffer()
            m_CommPort.Close()
            Return True
        Catch exc As Exception
            ' Không đọc được gì từ Comm Port
            m_CommPort.Close()
            Return False
        End Try
    Catch exc As Exception
        ' Không thể mở port hoặc gửi dữ liệu ra nó
        Me.clstPorts.SetItemChecked(port - 1, False)
        MsgBox("Could not open port.", MsgBoxStyle.OKOnly, Me.Text)
        Return False
    End Try

End Function
```

```
    ' Kiểm tra một Port có sẵn sàng không
    Private Function IsPortAvailable(ByVal ComPort As Integer) As Boolean
        Try
            m_CommPort.Open(ComPort, 115200, 8, Rs232.DataParity.Parity_None, _
                Rs232.DataStopBit.StopBit_1, 4096)
            ' Nếu có thể chạy đến đây thì cổng COM sẵn sàng
            m_CommPort.Close()
            Return True
        Catch
            ' Nếu chạy đến đây thì mở cổng COM không thành công
            Return False
        End Try
    End Function

    ' Ghi một chuỗi ra TextBox txtStatus
    Private Sub WriteMessage(ByVal message As String)
        Me.txtStatus.Text += message + vbCrLf
    End Sub


    ' Ghi một chuỗi ra TextBox txtStatus và chuyển qua dòng mới
    Private Sub WriteMessage(ByVal message As String, ByVal linefeed As Boolean)
        Me.txtStatus.Text += message
        If linefeed Then
            Me.txtStatus.Text += vbCrLf
        End If
    End Sub
End Class
```

88. Sử dụng Event Log

Chương trình này chỉ cách tương tác với Event Log.



Các chức năng chính:

- Event Log có thể dễ dàng được sử dụng bởi nhiều ứng dụng khác nhau để lưu các lỗi ứng dụng, các khuyến cáo, và các thông tin chung. Sử dụng class EventLog chứa trong namespace System.Diagnostics, ta có thể dễ dàng ghi vào hoặc đọc thông tin từ Event Log của máy tính cục bộ hoặc của máy tính ở xa. Trong chương trình này ta sẽ thấy cách đọc, ghi vào các Log đã có cũng như cách tạo mới và xóa các Log.

Project: **Su dung Event Log**

frmMain.vb

```
Option Strict On
Public Class frmMain
    Inherits System.Windows.Forms.Form
    Private entryType As EventLogEntryType = EventLogEntryType.Error
    ' Khi ghi vào Event Log cần truyền vào tên máy tính và phải có quyền ghi vào.

    Private machineName As String = Environment.MachineName
    Private logType As String = ""
    Dim frmWrite As New frmWrite()
    Dim frmRead As New frmRead()
    Dim frmCreateDelete As New frmCreateDelete()

' Code phát sinh form được bỏ qua

    Private Sub btnWrite_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnWrite.Click
        If frmWrite Is Nothing OrElse frmWrite.IsDisposed Then
            frmWrite = New frmWrite()
        End If
        frmWrite.Show()
        frmWrite.BringToFront()
    End Sub

    Private Sub btnRead_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnRead.Click
        If frmRead Is Nothing OrElse frmRead.IsDisposed Then
            frmRead = New frmRead()
        End If
```

```
    frmRead.Show()
    frmRead.BringToFront()
  End Sub

  Private Sub btnCreateDelete_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnCreateDelete.Click
    If frmCreateDelete Is Nothing OrElse frmCreateDelete.IsDisposed Then
      frmCreateDelete = New frmCreateDelete()
    End If
    frmCreateDelete.Show()
    frmCreateDelete.BringToFront()
  End Sub
End Class
```

frmWrite.vb

```
Imports System.Security
Public Class frmWrite
  Inherits System.Windows.Forms.Form
  Private entryType As System.Diagnostics.EventLogEntryType =
  EventLogEntryType.Information

' Code phát sinh form được bỏ qua

  Private Sub cmdWriteEntry_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) _
  Handles btnWriteEntry.Click

    Try
      If IsNumeric(txtEventID.Text) Then

        ' Tham số đầu tiên là tên của Log sẽ ghi vào, tham số thứ hai là tên
        ' máy tính, tham số thứ ba là nguồn của sự kiện (thường là tên của
        ' ứng dụng)

        Dim ev As New EventLog("Application", _
                  System.Environment.MachineName, _
                  "Su dung Event Log")

        ev.WriteEntry(txtEntry.Text, entryType, CInt(txtEventID.Text))
        ev.Close()
```

```
            MessageBox.Show("Entry written to the event log", _
               "frmWrite", _
               MessageBoxButtons.OK, _
               MessageBoxIcon.Information)
         Else
            ' EventID không phải là số
          MessageBox.Show("Value entered into EventID textbox must be numeric", _
               "Event ID Entry Error", _
               MessageBoxButtons.OK, _
               MessageBoxIcon.Exclamation)
         End If

      Catch secEx As SecurityException
       MessageBox.Show("Error writing to the event log: this may be due to a lack" & _
            "of appropriate permissions." & vbCrLf & secEx.Message, _
            "Lack of Permissions", _
            MessageBoxButtons.OK, _
            MessageBoxIcon.Error)

      Catch ex As Exception
         MessageBox.Show("Error accessing logs on the local machine." & _
           vbCrLf & ex.Message, _
           "Error accessing logs.", _
           MessageBoxButtons.OK, _
           MessageBoxIcon.Error)
      End Try
   End Sub

   Private Sub rdo_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
   Handles rdoError.Click, rdoInfo.Click, rdoWarning.Click
      Dim rdo As RadioButton = CType(sender, RadioButton)
      Select Case rdo.Name
         Case "rdoError"
            entryType = EventLogEntryType.Error
         Case "rdoWarning"
            entryType = EventLogEntryType.Warning
         Case "rdoInfo"
            entryType = EventLogEntryType.Information
```

```
        Case Else
            Debug.Assert(False, "User selected an event log type that" & _
                                " is not currently handled")
    End Select
  End Sub
End Class
```

frmRead.vb

```
Imports System.Security
Public Class frmRead
  Inherits System.Windows.Forms.Form

  ' Lưu tên của Log mà người sử dụng muốn xem
  Private logType As String = ""

' Code phát sinh form được bỏ qua

  Private Sub cmdViewLogEntries_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnViewLogEntries.Click
    Try
      Const ENTRIES_TO_DISPLAY As Integer = 10
      Dim ev As New EventLog(logType, System.Environment.MachineName, _
        "Su dung EventLog")
      Dim entry As EventLogEntry

      rchEventLogOutput.Text = "Event log entries (maximum of 10)," & _
                               " newest to oldest" & vbCrLf & vbCrLf

      Dim LastLogToShow As Integer = ev.Entries.Count - ENTRIES_TO_DISPLAY
      If LastLogToShow < 0 Then
        LastLogToShow = 0
      End If
       ' Hiển thị 10 record cuối cùng của log được chọn
      Dim i As Integer
      For i = ev.Entries.Count - 1 To LastLogToShow Step -1
        Dim CurrentEntry As EventLogEntry = ev.Entries(i)

        rchEventLogOutput.Text &= "Event ID : " & _
          CurrentEntry.EventID & vbCrLf
```

```
            rchEventLogOutput.Text &= "Entry Type : " & _
              CurrentEntry.EntryType.ToString() & vbCrLf
            rchEventLogOutput.Text &= "Message : " & _
              CurrentEntry.Message & vbCrLf & vbCrLf
        Next

        ' Có thể duyệt qua tất cả record trong log sử dụng vòng lặp for như sau
        ' For Each entry In ev.Entries
        ' Next
    Catch secEx As SecurityException
        MessageBox.Show("Error reading an event log: this may be due to a lack of
appropriate permissions." & vbCrLf & secEx.Message, _
          "Lack of Permissions", _
          MessageBoxButtons.OK, _
          MessageBoxIcon.Error)
    Catch ex As Exception
        MessageBox.Show("Error accessing logs on the local machine." & vbCrLf &
ex.Message, _
          "Error accessing logs.", _
          MessageBoxButtons.OK, _
          MessageBoxIcon.Error)
    End Try
End Sub

Private Sub lstEntryType_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) _
Handles lstEntryType.SelectedIndexChanged
    ' Lưu log được chọn
    logType = CType(lstEntryType.Items(lstEntryType.SelectedIndex()), String)
End Sub

Private Sub frmRead_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Try
        Dim CurrentLog As EventLog
        For Each CurrentLog In EventLog.GetEventLogs()
            lstEntryType.Items.Add(CurrentLog.LogDisplayName)
        Next
        lstEntryType.SelectedIndex = 0
```

```
    Catch secEx As SecurityException
      MessageBox.Show("Error reading an event log: this may be due to a lack of
  appropriate permissions." & vbCrLf & secEx.Message, _
        "Lack of Permissions", _
        MessageBoxButtons.OK, _
        MessageBoxIcon.Error)
    Catch ex As Exception
      MessageBox.Show("Error accessing logs on the local machine." & _
        vbCrLf & ex.Message, _
        "Error accessing logs.", _
        MessageBoxButtons.OK, _
        MessageBoxIcon.Error)
    End Try
  End Sub
End Class
```

frmCreateDelete.vb

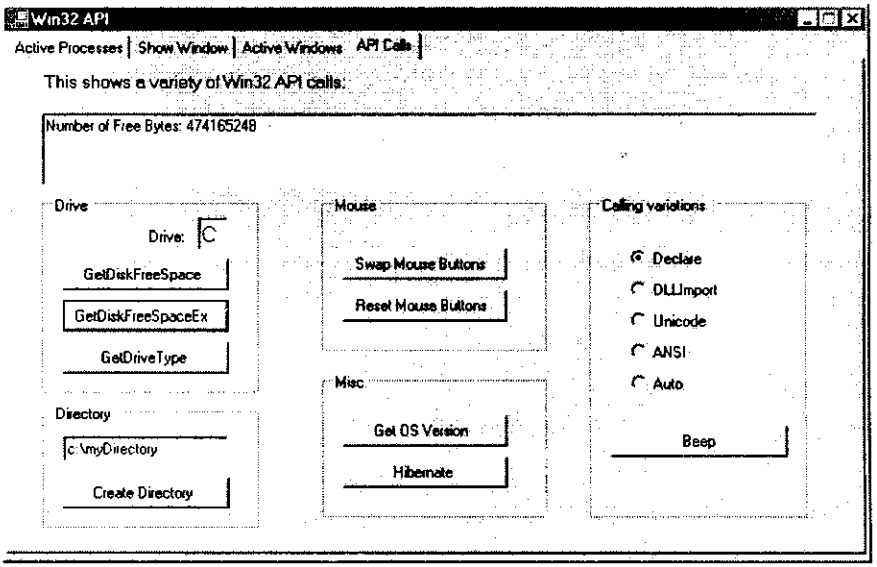
```
Public Class frmCreateDelete
  Inherits System.Windows.Forms.Form

' code phát sinh form được bỏ qua

  Private Sub cmdCreateLog_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnCreateLog.Click
    Dim ev As New EventLog()
    Try
      ' Kiểm tra xem có tồn tại log muốn tạo
      If Not ev.Exists(txtLogNameToCreate.Text) Then
        If ev.SourceExists("Su dung EventLog") Then
          ev.DeleteEventSource("Syu dung EventLog")
        End If
        ev.CreateEventSource("Su dung EventLog", txtLogNameToCreate.Text)
      Else
        MessageBox.Show("Log :" & txtLogNameToCreate.Text + _
          " already exists.", _
          "Log Exists", _
          MessageBoxButtons.OK, _
          MessageBoxIcon.Exclamation)
      End If
```

```
    Catch ex As Exception
        MessageBox.Show("Unable to Create Event Log, the message was: " &
ex.Message & " This may be due to a lack of appropriate permissions.", _
            "An Exception was thrown", _
            MessageBoxButtons.OK, _
            MessageBoxIcon.Exclamation)
    Finally
        ' Đóng Log
        ev.Close()
    End Try

    ' Ta cũng có thể tạo một Event Log như sau:

    ' Dim ev2 As New EventLog(txtLogNameToCreate.Text, _
    '       System.Environment.MachineName, _
    '       "Su dung EventLog")
    ' ev2.WriteEntry("test message")
    ' ev2.Close()
End Sub

Private Sub cmdDeleteLog_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDeleteLog.Click
    Dim ev As New System.Diagnostics.EventLog()
    Try
        If ev.Exists(txtLogNameToDelete.Text) Then
            ' Gọi phương thức xóa log
            ev.Delete(txtLogNameToDelete.Text)
        Else
            MessageBox.Show("Log " + txtLogNameToDelete.Text + _
                " does not exist, therefore it cannot be deleted.", _
                "Log Does not Exist", _
                MessageBoxButtons.OK, _
                MessageBoxIcon.Error)
        End If
    Catch ex As Exception
        MessageBox.Show("Unable to delete event log, the message was: " & _
            ex.Message & " this may be due to a lack of appropriate permissions.", _
            "An Exception was thrown", _
            MessageBoxButtons.OK, _
            MessageBoxIcon.Exclamation)
```

```
    Finally
      ev.Close()
    End Try
  End Sub

End Class
```

89. Sử dụng Win32 API

Chương trình này cho biết cách sử dụng các hàm Win32 API trong VB.NET.



Các chức năng chính:

- **Active Processes**: Liệt kê tất cả các Windows process. Sử dụng một hàm Win32 callback EnumWindows bằng cách sử dụng một đối tượng ủy quyền (delegate).
- **Active Windows**: Liệt kê tất cả các Windows đang chạy, sử dụng DllImport.
- **Show Window**: Nhập vào tựa đề của Windows hoặc tên class, và đem cửa sổ đó lên trên.
- **API Calls**: Sử dụng các hàm API: GetDiskFreeSpace, GetDiskFreeSpaceEx, GetDriveType, CreateDirectory, SwapMouseButton, IsPwrHibernateAllowed, SetSuspendState, và Beep.

Project: **Win32 API**

Win32API.vb

```
Option Strict On
Imports System.Runtime.InteropServices
Imports System.Text

' Class chứa các hằng số và hàm Windows 32 API
Public Class Win32API
    <StructLayout(LayoutKind.Sequential)> _
    Public Structure OSVersionInfo
        Public OSVersionInfoSize As Integer
        Public majorVersion As Integer
        Public minorVersion As Integer
        Public buildNumber As Integer
        Public platformId As Integer
        <MarshalAs(UnmanagedType.ByValTStr, SizeConst:=128)> _
        Public versionString As String
    End Structure

    <StructLayout(LayoutKind.Sequential)> _
    Public Structure SECURITY_ATTRIBUTES
        Public nLength As Integer
        Public lpSecurityDescriptor As Integer
        Public bInheritHandle As Integer
    End Structure

    Public Const GWL_EXSTYLE As Integer = (-20)
    Public Const GW_OWNER As Integer = 4
    Public Const SW_RESTORE As Integer = 9
    Public Const SW_SHOW As Integer = 5
    Public Const WS_EX_TOOLWINDOW As Integer = &H80
    Public Const WS_EX_APPWINDOW As Integer = &H40000

    Public Declare Function CreateDirectory Lib "kernel32" _
      Alias "CreateDirectoryA" (ByVal lpPathName As String, _
                        ByVal lpSecurityAttributes _
                          As SECURITY_ATTRIBUTES) As Boolean
```

```
Public Delegate Function EnumWindowsCallback(ByVal hWnd As Integer, _
                                              ByVal lParam As Integer) As Boolean

Public Declare Function EnumWindows Lib "user32.dll" _
  Alias "EnumWindows" (ByVal callback As EnumWindowsCallback, _
                       ByVal lParam As Integer) As Integer

<DllImport("user32.dll", EntryPoint:="EnumWindows", SetLastError:=True, _
CharSet:=CharSet.Ansi, ExactSpelling:=True, _
CallingConvention:=CallingConvention.StdCall)> _

Public Shared Function EnumWindowsDllImport(ByVal callback As
EnumWindowsCallback, ByVal lParam As Integer) As Integer
End Function

Public Declare Auto Function FindWindow Lib "user32.dll" _
  Alias "FindWindow" (ByVal lpClassName As String, _
                      ByVal lpWindowName As String) As Integer

Public Declare Auto Function FindWindowAny Lib "user32.dll" _
Alias "FindWindow" (ByVal lpClassName As Integer, _
                    ByVal lpWindowName As Integer) As Integer

Public Declare Auto Function FindWindowNullClassName Lib "user32.dll" _
  Alias "FindWindow" (ByVal lpClassName As Integer, _
                      ByVal lpWindowName As String) As Integer

Public Declare Auto Function FindWindowNullWindowCaption Lib "user32.dll" _
  Alias "FindWindow" (ByVal lpClassName As String, _
                      ByVal lpWindowName As Integer) As Integer

Public Declare Function GetClassName Lib "user32.dll" _
  Alias "GetClassNameA" (ByVal hwnd As Integer, _
                         ByVal lpClassName As StringBuilder, _
                         ByVal cch As Integer) As Integer

Public Declare Function GetDiskFreeSpace Lib "kernel32" _
  Alias "GetDiskFreeSpaceA" (ByVal lpRootPathName As String, _
                             ByRef lpSectorsPerCluster As Integer, _
                             ByRef lpBytesPerSector As Integer, _
```

```
                    ByRef lpNumberOfFreeClusters As Integer, _
                    ByRef lpTotalNumberOfClusters As Integer) As Integer

Public Declare Function GetDiskFreeSpaceEx Lib "kernel32" _
  Alias "GetDiskFreeSpaceExA" (ByVal lpRootPathName As String, _
                    ByRef lpFreeBytesAvailableToCaller As Integer, _
                    ByRef lpTotalNumberOfBytes As Integer, _
                    ByRef lpTotalNumberOfFreeBytes As UInt32) As Integer

Public Declare Function GetDriveType Lib "kernel32" _
  Alias "GetDriveTypeA" (ByVal nDrive As String) As Integer

Public Declare Function GetParent Lib "user32.dll" _
  Alias "GetParent" (ByVal hwnd As Integer) As Integer

Declare Ansi Function GetVersionEx Lib "kernel32.dll" _
  Alias "GetVersionExA" (ByRef osvi As OSVersionInfo) As Boolean

Public Declare Function GetWindow Lib "user32.dll" _
  Alias "GetWindow" (ByVal hwnd As Integer, _
            ByVal wCmd As Integer) As Integer

Public Declare Function GetWindowLong Lib "user32.dll" _
  Alias "GetWindowLongA" (ByVal hwnd As Integer, _
                ByVal nIndex As Integer) As Integer

Public Declare Sub GetWindowText Lib "user32.dll" _
  Alias "GetWindowTextA" (ByVal hWnd As Integer, _
                ByVal lpString As StringBuilder, _
                ByVal nMaxCount As Integer)

Public Declare Function IsIconic Lib "user32.dll" _
  Alias "IsIconic" (ByVal hwnd As Integer) As Boolean

Public Declare Function IsPwrHibernateAllowed Lib "Powrprof.dll" _
  Alias "IsPwrHibernateAllowed" () As Integer

Public Declare Function IsWindowVisible Lib "user32.dll" _
  Alias "IsWindowVisible" (ByVal hwnd As Integer) As Boolean
```

```
    Public Declare Function SetForegroundWindow Lib "user32.dll" _
        Alias "SetForegroundWindow" (ByVal hwnd As Integer) As Integer

    Public Declare Function SetSuspendState Lib "Powrprof.dll" _
        Alias "SetSuspendState" (ByVal Hibernate As Integer, _
                                 ByVal ForceCritical As Integer, _
                                 ByVal DisableWakeEvent As Integer) As Integer

    Public Declare Function ShowWindow Lib "user32.dll" _
        Alias "ShowWindow" (ByVal hwnd As Integer, ByVal nCmdShow As Integer) As
    Integer

    Declare Function SwapMouseButton Lib "user32.dll" _
        Alias "SwapMouseButton" (ByVal bSwap As Integer) As Integer

End Class
```

CallingVariations.vb

```
Option Strict On
Imports System.Runtime.InteropServices

' Class này chứa các kiểu gọi khác nhau của hàm Beep
Public Class CallingVariations

    ' Khai báo
    Public Declare Function DeclareBeep Lib "kernel32" Alias "Beep" _
        (ByVal dwFreq As Integer, ByVal dwDuration As Integer) As Integer

    ' DLLImport
    <DllImport("kernel32.dll", EntryPoint:="Beep")> _
    Public Shared Function DLLImportBeep(ByVal dwFreq As Integer, _
        ByVal dwDuration As Integer) As Integer
    End Function

    ' Chỉ định Unicode
    Public Declare Unicode Function UnicodeBeep Lib "kernel32" Alias "Beep" _
        (ByVal dwFreq As Integer, ByVal dwDuration As Integer) As Integer
```

```
    ' Chỉ định Ansi
    Public Declare Ansi Function ANSIBeep Lib "kernel32" Alias "Beep" _
        (ByVal dwFreq As Integer, ByVal dwDuration As Integer) As Integer

    ' Chỉ định Auto
    Public Declare Auto Function AutoBeep Lib "kernel32" Alias "Beep" _
        (ByVal dwFreq As Integer, ByVal dwDuration As Integer) As Integer

    ' Sử dụng Exact Spelling
    ' Mặc nhiên là False, nếu FALSE một ký tự A nối sau ANSI và W nối sau Unicode
    <DllImport("kernel32.dll", ExactSpelling:=True, CharSet:=CharSet.Ansi)> _
    Public Shared Function ExactSpellingBeep(ByVal dwFreq As Integer, _
        ByVal dwDuration As Integer) As Integer
    End Function

End Class
```

frmMain.vb

```
Option Strict On

Imports System
Imports System.Text
Imports System.Runtime.InteropServices

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Const STRING_BUFFER_LENGTH As Integer = 255

' Code phát sinh form được bỏ qua

#Region "API Calls"
    ' Gọi hàm Beep với nhiều cách gọi khác nhau
    Private Sub btnBeep_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnBeep.Click
        If rbDeclare.Checked Then
            CallingVariations.DeclareBeep(1000, 1000)
        ElseIf rbDLLImport.Checked Then
            CallingVariations.DLLImportBeep(1000, 1000)
```

```
    ElseIf rbANSI.Checked Then
      CallingVariations.ANSIBeep(1000, 1000)
    ElseIf rbUnicode.Checked Then
      CallingVariations.UnicodeBeep(1000, 1000)
    ElseIf rbAuto.Checked Then
      CallingVariations.AutoBeep(1000, 1000)
    End If
  End Sub

  ' Tạo một thư mục nếu có thể, và cập nhật thông báo
  Private Sub btnCreateDirectory_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnCreateDirectory.Click
    Dim security As New Win32API.SECURITY_ATTRIBUTES()
    If Win32API.CreateDirectory(txtDirectory.Text, security) Then
      txtFunctionOutput.Text = "Directory created."
    Else
      txtFunctionOutput.Text = "Directory not created."
    End If
  End Sub

  ' Lấy số cluster còn trống trên đĩa, sử dụng hàm API GetDiskFreeSpace
  Private Sub btnGetFreeSpace_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnGetFreeSpace.Click
    Dim rootPathName As String
    Dim sectorsPerCluster As Integer
    Dim bytesPerSector As Integer
    Dim numberOfFreeClusters As Integer
    Dim totalNumberOfClusters As Integer

    rootPathName = txtDriveLetter.Text + ":\"

    Win32API.GetDiskFreeSpace(rootPathName, sectorsPerCluster, _
      bytesPerSector, _
      numberOfFreeClusters, totalNumberOfClusters)

    txtFunctionOutput.Text = "Number of Free Clusters: " & _
      numberOfFreeClusters.ToString()
  End Sub
```

```
' Lấy số byte còn trống trên đĩa, sử dụng API GetDiskFreeSpaceEx
Private Sub btnGetDiskFreeSpaceEx_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnGetDiskFreeSpaceEx.Click
    Dim rootPathName As String
    Dim freeBytesToCaller As Integer
    Dim totalNumberOfBytes As Integer
    Dim totalNumberOfFreeBytes As UInt32

    rootPathName = txtDriveLetter.Text + ":\"

    Win32API.GetDiskFreeSpaceEx(rootPathName, freeBytesToCaller, _
      totalNumberOfBytes, _
      totalNumberOfFreeBytes)

    txtFunctionOutput.Text = "Number of Free Bytes: " & _
      totalNumberOfFreeBytes.ToString()
End Sub

' Lấy thông tin kiểu ổ đĩa, sử dụng API GetDriveType
Private Sub btnGetDriveType_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnGetDriveType.Click
    Dim rootPathName As String
    rootPathName = txtDriveLetter.Text + ":\"

    Select Case Win32API.GetDriveType(rootPathName)
      Case 2
        txtFunctionOutput.Text = "Drive type: Removable"
      Case 3
        txtFunctionOutput.Text = "Drive type: Fixed"
      Case Is = 4
        txtFunctionOutput.Text = "Drive type: Remote"
      Case Is = 5
        txtFunctionOutput.Text = "Drive type: Cd-Rom"
      Case Is = 6
        txtFunctionOutput.Text = "Drive type: Ram disk"
      Case Else
        txtFunctionOutput.Text = "Drive type: Unrecognized"
    End Select

End Sub
```

```
    ' Lấy phiên bản của hệ điều hành
    Private Sub btnGetOSVersion_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnGetOSVersion.Click
        Dim versionInfo As New Win32API.OSVersionInfo()

        versionInfo.OSVersionInfoSize = Marshal.SizeOf(versionInfo)
        Win32API.GetVersionEx(versionInfo)

        txtFunctionOutput.Text = "Build Number is: " & _
                    versionInfo.buildNumber.ToString() & Chr(13) & Chr(10)
        txtFunctionOutput.Text += "Major Version Number is: " & _
                    versionInfo.majorVersion.ToString()
    End Sub

    ' Hàm này cho máy tính tạm ngưng nếu có thể
    Private Sub btnHibernate_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnHibernate.Click
        If Win32API.IsPwrHibernateAllowed() <> 0 Then
            Win32API.SetSuspendState(1, 0, 0)
        Else
            txtFunctionOutput.Text = "Your computer doesn't support hibernation.  " & _
                "This may be due to system settings or simply a computer bios that does not
    support hibernation."
        End If
    End Sub

    ' Quy định lại các nút của chuột theo ban đầu
    Private Sub btnResetMouseButton_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnResetMouseButton.Click
        Win32API.SwapMouseButton(0)
    End Sub

    ' Chuyển chức năng các nút chuột
    Private Sub btnSwapMouseButton_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnSwapMouseButton.Click
        Win32API.SwapMouseButton(1)
    End Sub

#End Region
```

```
' Xóa nội dung và cho các process đang hoạt động vào ListView
Private Sub btnRefreshActiveProcesses_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnRefreshActiveProcesses.Click
    lvwProcessList.Items.Clear()

    Win32API.EnumWindows(New Win32API.EnumWindowsCallback(AddressOf _
        FillActiveProcessList), 0)
End Sub

' Xóa ListView và cho các cửa sổ đang hoạt động vào ListView
Private Sub btnRefreshActiveWindows_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnRefreshActiveWindows.Click
    lbActiveWindows.Items.Clear()

    Win32API.EnumWindowsDllImport(New _
        Win32API.EnumWindowsCallback(AddressOf _
        FillActiveWindowsList), 0)
End Sub

' Thủ tục này tìm một cửa sổ có tựa đề nhập vào và đưa nó lên trên
Private Sub btnShow_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShow.Click
    Dim hWnd As Integer

    If txtWindowCaption.Text = "" And txtClassName.Text = "" Then
        ' Tìm bất kỳ cửa sổ nào có sẵn
        hWnd = Win32API.FindWindowAny(0, 0)
    ElseIf txtWindowCaption.Text = "" And txtClassName.Text <> "" Then
        ' Tìm cửa sổ theo tên class
        hWnd = Win32API.FindWindowNullWindowCaption(txtClassName.Text, 0)
    ElseIf txtWindowCaption.Text <> "" And txtClassName.Text = "" Then
        ' Tìm cửa sổ theo tên của tiêu đề
        hWnd = Win32API.FindWindowNullClassName(0, txtWindowCaption.Text)
    Else

        ' Tìm cửa sổ theo cả tên class và tên tiêu đề
        hWnd = Win32API.FindWindow(txtClassName.Text, txtWindowCaption.Text)
    End If
```

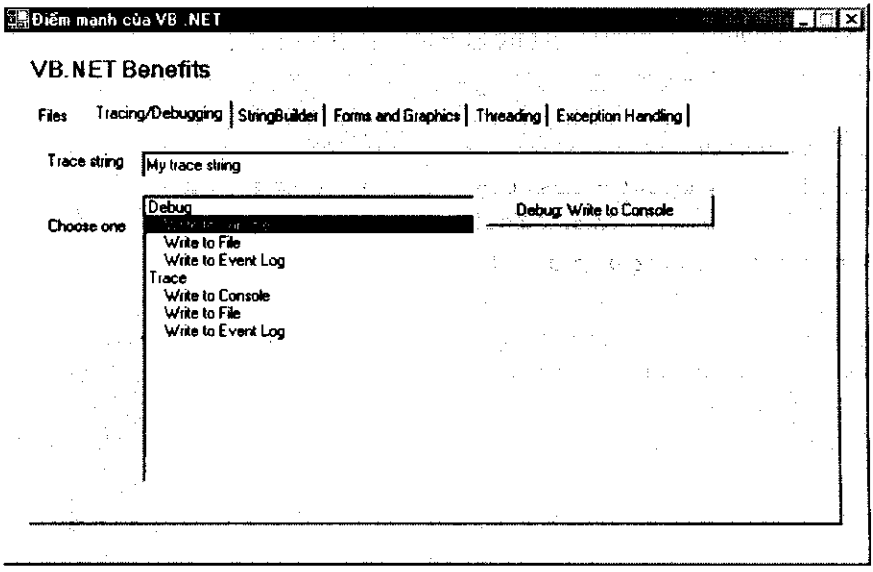
```
        ' Nếu không tìm thấy, FindWindow trả về 0.
        If hWnd = 0 Then
            MsgBox("Specified window is not running.", _
                   MsgBoxStyle.Exclamation, Me.Text)
        Else
            ' Đưa cửa sổ lên trên
            Win32API.SetForegroundWindow(hWnd)

            If Win32API.IsIconic(hWnd) Then
                Win32API.ShowWindow(hWnd, Win32API.SW_RESTORE)
            Else
                Win32API.ShowWindow(hWnd, Win32API.SW_SHOW)
            End If
        End If
    End Sub

    ' Kiểm tra các TextBox tên class và tên tiêu đề và cập nhật label lblFunctionCalled
    Private Sub ChangeFunctionCalledLabel()
        If txtWindowCaption.Text = "" And txtClassName.Text = "" Then
            lblFunctionCalled.Text = _
            "FindWindow (ClassName As Integer, WindowName As Integer) will be called."
        ElseIf txtWindowCaption.Text = "" And txtClassName.Text <> "" Then
            lblFunctionCalled.Text = _
            "FindWindow (ClassName As String, WindowName As Integer) will be called."
        ElseIf txtWindowCaption.Text <> "" And txtClassName.Text = "" Then
            lblFunctionCalled.Text = _
            "FindWindow (ClassName As Integer, WindowName As String) will be called."
        Else
            lblFunctionCalled.Text = _
            "FindWindow (ClassName As String, WindowName As String) will be called."
        End If
    End Sub

    ' Hàm này được gọi (cho mỗi process đang hoạt động) bởi EnumWindows
    Function FillActiveProcessList(ByVal hWnd As Integer, ByVal lParam As Integer)
    As Boolean

        Dim windowText As New StringBuilder(STRING_BUFFER_LENGTH)
        Dim className As New StringBuilder(STRING_BUFFER_LENGTH)
```

```
        ' Lấy tên tiêu đề và tên class.
        Win32API.GetWindowText(hWnd, windowText, STRING_BUFFER_LENGTH)
        Win32API.GetClassName(hWnd, className, STRING_BUFFER_LENGTH)

        ' Cho process vào ListView Processes
        Dim processItem As New ListViewItem(windowText.ToString, 0)
        processItem.SubItems.Add(className.ToString)
        processItem.SubItems.Add(hWnd.ToString)
        lvwProcessList.Items.Add(processItem)

        Return True
    End Function

    Function FillActiveWindowsList(ByVal hWnd As Integer, ByVal lParam As Integer)
    As Boolean
        Dim windowText As New StringBuilder(STRING_BUFFER_LENGTH)

        ' Lấy Window Caption.
        Win32API.GetWindowText(hWnd, windowText, STRING_BUFFER_LENGTH)

        ' Chỉ cho các cửa sổ vào danh sách Windows
        If ProcessIsActiveWindow(hWnd) Then
            lbActiveWindows.Items.Add(windowText)
        End If

        Return True
    End Function

    ' Kiểm tra một process có phải là một cửa sổ đang hoạt động
    Function ProcessIsActiveWindow(ByVal hWnd As Integer) As Boolean
        Dim windowText As New StringBuilder(STRING_BUFFER_LENGTH)
        Dim windowIsOwned As Boolean
        Dim windowStyle As Integer

        ' Lấy Windows Caption, thông tin, và kiểu cửa sổ
        Win32API.GetWindowText(hWnd, windowText, STRING_BUFFER_LENGTH)
        windowIsOwned = Win32API.GetWindow(hWnd, Win32API.GW_OWNER) <> 0
        windowStyle = Win32API.GetWindowLong(hWnd, Win32API.GWL_EXSTYLE)
```

```
        ' Không cho phép các process không phải là cửa sổ
        If Not Win32API.IsWindowVisible(hWnd) Then
            Return False
        End If

        ' Cửa sổ phải có tiêu đề
        If windowText.ToString.Equals("") Then
            Return False
        End If

        ' Không có cửa sổ cha
        If Win32API.GetParent(hWnd) <> 0 Then
            Return False
        End If

        ' Không chấp nhận ToolTip
        If (windowStyle And Win32API.WS_EX_TOOLWINDOW) <> 0 And Not _
                windowIsOwned Then
            Return False
        End If

        If (windowStyle And Win32API.WS_EX_APPWINDOW) = 0 And _
                windowIsOwned Then
            Return False
        End If
        Return True
    End Function

    ' Khi thay đổi txtClassName, cập nhật lblFunctionCalled
    Private Sub txtClassName_TextChanged(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles txtClassName.TextChanged
        ChangeFunctionCalledLabel()
    End Sub

    ' Khi thay đổi txtWindowCaption, cập nhật lblFunctionCalled
    Private Sub txtWindowCaption_TextChanged(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles txtWindowCaption.TextChanged
        ChangeFunctionCalledLabel()
    End Sub
End Class
```

90. Một số lợi điểm của Visual Basic .NET

Chương trình này biểu diễn một số lợi điểm chính của VB.NET.



Các chức năng chính:

- Sử dụng đối tượng StringBuider, cung cấp các khả năng hiệu quả để xử lý các chuỗi.
- Các hàm thao tác với file, bao gồm đọc và ghi vào file, sao chép, di chuyển và xóa file, và truy xuất các thuộc tính của file.
- Bắt lỗi và theo dõi, sử dụng các đối tượng Debug và Trace, tốt hơn so với VB 6.
- Cấu trúc bắt lỗi Try...Catch...Finally...End Try, dùng để thay thế cho On Error Goto và linh động hơn chỉ dẫn này.
- Thừa kế, bao gồm cả các thủ tục overload, override các phương thức của class cơ sở trong class thừa kế... (xem project: 82. Scope-Override-Overload)
- Làm việc với tiểu trình (threading).
- Khả năng của Windows Form, kể cả anchoring và docking, dễ dàng với thứ tự tab, và các khả năng đồ họa mới.

Project: **Diem manh cua VB.NET**

frmMain.vb

```
Option Strict On
Imports System.Text
Imports System.IO
Imports System.Threading

Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        lstFileFunction.SelectedIndex = 0
        lstDebugFunction.SelectedIndex = 0
        txtSBAppend.Text = "My string"
        lstExceptionHandling.SelectedIndex = 0
    End Sub

    Private Sub btnOpenDemoForm_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnOpenDemoForm.Click
        ' Tạo một instance mới
        Dim frm As New frmControls()
        ' Hiển thị theo chế độ hộp thoại
        frm.ShowDialog()
    End Sub

    Private Sub btnShowGraphics_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnShowGraphics.Click
        Dim frmG As New frmGraphics()
        frmG.ShowDialog()
    End Sub

    Private Sub btnStringBuilderFunction_Click(ByVal sender As System.Object, ByVal
    e As System.EventArgs) Handles btnStringBuilderFunction.Click

        Dim strSBOrig As String = txtSBOrig.Text.Trim
        Dim strConcatenated As String
        Dim strSBAppend As String = txtSBAppend.Text.Trim
        Dim intStrIterations As Integer = CInt(txtStrIterations.Text)
```

```
    Dim intSBIterations As Integer = CInt(txtSBIterations.Text)
    Dim sb As New StringBuilder(strSBAppend.Length * intSBIterations)
    Dim tmr As New TimeCheck()

    Dim i As Integer
    Try
      txtStrResults.Text = vbNullString
      txtSBResults.Text = vbNullString
      Cursor.Current = System.Windows.Forms.Cursors.WaitCursor

      ' Nối chuỗi theo kiểu VB6
      tmr.Begin()
      strConcatenated = strSBOrig
      For i = 1 To intStrIterations
        strConcatenated = strConcatenated & strSBAppend
      Next
      tmr.End()
      txtStrResults.Text = tmr.Milliseconds.ToString()

      ' Sử dụng StringBuilder
      tmr.Begin()
      For i = 1 To intSBIterations
        sb.Append(strSBAppend)
      Next
      tmr.End()
      txtSBResults.Text = tmr.Milliseconds.ToString()
    Catch exp As Exception
      MsgBox(exp.ToString(), MsgBoxStyle.OKOnly Or MsgBoxStyle.Critical, _
                             Me.Text)
    Finally
      Cursor.Current = System.Windows.Forms.Cursors.Default
    End Try
  End Sub

  ' Chỉ chấp nhận số trong TextBox
  Private Sub txtStrIterations_KeyPress(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyPressEventArgs) Handles txtStrIterations.KeyPress
    Dim KeyStroke As Char
    Dim chrNullCharacter As Char = Chr(0)
```

```
    KeyStroke = e.KeyChar
    Select Case KeyStroke
      Çase "0"c To "9"c, CChar(vbBack), CChar(vbCr)
          ' Số từ 0 đến 9, backspace, return thì không làm gì cả
      Case Else
          ' Bỏ tất cả các phím khác
        KeyStroke = chrNullCharacter
    End Select

    If KeyStroke = chrNullCharacter Then
      e.Handled = True
    Else
      e.Handled = False
    End If
  End Sub

  Private Sub lstFileFunction_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
lstFileFunction.SelectedIndexChanged
    Select Case lstFileFunction.SelectedIndex
      Case 0 ' Đọc một file
        btnFileFunction.Text = "&Read file"
      Case 1 ' Ghi vào một file
        btnFileFunction.Text = "&Write file"
      Case 2 ' Kích thước của một file
        btnFileFunction.Text = "&Get file size"
      Case 3 ' Thông tin File Version
        btnFileFunction.Text = "&Get file version"
      Case 4 ' Tên file tạm
        btnFileFunction.Text = "&Get temp file name"
    End Select
  End Sub

  Private Sub btnFileFunction_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnFileFunction.Click
    ' Các class trong namespace System.IO cho phép đọc và ghi vào các file
    ' cũng như các stream khác, như các memory stream, network stream...
    Dim strFile As String = "..\DemoText.txt"
    Dim strFileWrite As String = "..\DemoText_Write.txt"
```

```
    Select Case lstFileFunction.SelectedIndex
      Case 0 ' Đọc một file
        Dim srFile As New StreamReader(strFile)
        txtFileResult.Text = srFile.ReadToEnd()
        srFile.Close()
      Case 1 ' Ghi vào một file
        Dim swFile As StreamWriter = File.CreateText(strFileWrite)
        swFile.WriteLine("The quick brown fox jumped over the lazy dogs.")
        swFile.Flush()
        swFile.Close()
        Dim sr As New StreamReader(strFileWrite)
        txtFileResult.Text = sr.ReadToEnd()
        sr.Close()
      Case 2 ' Kích thước file
        Dim fiFile As New FileInfo(strFileWrite)
        txtFileResult.Text = "Size of " & _
          strFileWrite.Substring(InStr("/", strFileWrite)) & ": " & _
          fiFile.Length.ToString + " bytes."
      Case 3 ' File version info
        Dim env As System.Environment
        strFile = env.SystemDirectory & "\Notepad.exe"
        Dim viVersionInfo As FileVersionInfo = _
            FileVersionInfo.GetVersionInfo(strFile)
        txtFileResult.Text = "Version of " & strFile & ": " & _
            viVersionInfo.FileVersion & vbCrLf
        txtFileResult.Text &= "Description of " & strFile & ": " & _
            viVersionInfo.FileDescription
      Case 4 ' Tên file tạm
        txtFileResult.Text = "Temp file name: " & Path.GetTempFileName
    End Select

  End Sub

  Private Sub lstExceptionHandling_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
lstExceptionHandling.SelectedIndexChanged
    Select Case lstExceptionHandling.SelectedIndex
      Case 0
        btnExceptionHandlingFunction.Text = "&Basic Exception Handling"
```

```
        Case 1
            btnExceptionHandlingFunction.Text = "&Focused Exception Handling"
    End Select
End Sub

Private Sub btnExceptionHandlingFunction_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnExceptionHandlingFunction.Click
    Select Case lstExceptionHandling.SelectedIndex
        Case 0 ' Basic
            BasicException()
        Case 1 ' Focused
            ExceptionReadingFromFile()
    End Select
End Sub

Private Sub BasicException()
    ' Exception Handling thay thế cho On Error. Exception handling mạnh hơn và
    ' linh động hơn nhiều so với On Error!

    Dim dblNum1 As Double = 5
    Dim dblNum2 As Double = 0
    Dim intResult As Integer
    Try
        intResult = CInt(dblNum1 / dblNum2)
    Catch exp As Exception
        ' Thông báo lỗi gặp phải
        ' exp có thể thay bằng tên hợp lệ khác
        txtExceptionHandlingResult.Text = exp.Message
    End Try
End Sub

Private Sub ExceptionReadingFromFile()
    Try
        ' Cố mở một file trong một thư mục không tồn tại
        Dim sw As New StreamWriter("c:\12345678asdf\baddirectory.txt")
    Catch dirNotFoundEX As DirectoryNotFoundException
        ' Bắt cụ thể một exception
        txtExceptionHandlingResult.Text = "Message: " &
                    dirNotFoundEX.Message & vbCrLf & vbCrLf
```

```
        txtExceptionHandlingResult.Text &= "Stack Trace: " & _
                    dirNotFoundEX.StackTrace

    Catch exp As Exception
        ' Bắt các lỗi khác
        MsgBox(exp.ToString(), MsgBoxStyle.OKOnly Or MsgBoxStyle.Critical, _
                                Me.Text)
    End Try
End Sub

' Sử dụng class Debug nếu muốn xuất trong phiên bản Debug của chương trình.
' Sử dụng class Trace nếu muốn xuất trong cả hai phiên bản Debug và Release.
Private Sub lstDebugFunction_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
lstDebugFunction.SelectedIndexChanged
    Select Case lstDebugFunction.SelectedIndex
        Case 0 ' Debug:
            lstDebugFunction.SelectedIndex = 1
        Case 1 ' Debug: Ghi ra cửa sổ Console (Ouput)
            btnDebugFunction.Text = "&Debug: Write to Console"
        Case 2 ' Debug: Ghi vào File
            btnDebugFunction.Text = "&Debug: Write to File"
        Case 3 ' Debug: Ghi vào Event Log
            btnDebugFunction.Text = "&Debug: Write to Event Log"
        Case 4 ' Trace:
            lstDebugFunction.SelectedIndex = 5
        Case 5 ' Trace: Ghi ra Console (Output)
            btnDebugFunction.Text = "&Trace: Write to Console"
        Case 6 ' Trace: Ghi vào File
            btnDebugFunction.Text = "&Trace: Write to File"
        Case 7 ' Trace: Ghi vào Event Log
            btnDebugFunction.Text = "&Trace: Write to Event Log"
    End Select
End Sub

Private Sub btnDebugFunction_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDebugFunction.Click
    Dim strDebug As String = txtDebugInfo.Text
    Debug.Indent()
```

```
Select Case lstDebugFunction.SelectedIndex
  Case 0 ' Debug:
  Case 1 ' Debug: Ghi vào cửa sổ Console (Ouput)
    Debug.WriteLine(strDebug)
    Debug.Flush()
    sbrPanel2.Text = "Wrote debug output to console"
  Case 2 ' Debug: Ghi vào file
      ' Tạo một file để lưu
    Dim strFile As String = "C:\DebugOutput.txt"
    Dim stmFile As Stream = File.Create(strFile)
      ' Sử dụng Listener
    Dim twTextListener As New TextWriterTraceListener(stmFile)
    With Debug.Listeners
      .Clear()
      .Add(twTextListener)
    End With

      ' Ghi vào file
    Debug.Write(strDebug)
    Debug.Flush()
    sbrPanel2.Text = "Wrote debug output to " & strFile
  Case 3 ' Debug: Ghi vào Event Log
      ' Tạo một debug listener cho event log
    Dim logdebugListener As New _
            EventLogTraceListener("ContrastVBNETwithVB6")
    With Debug.Listeners
      .Clear()
      .Add(logdebugListener)
    End With
    Debug.Write(strDebug)
    Debug.Flush()
    sbrPanel2.Text = "Wrote debug output to Event Log"

  Case 4 ' Trace:
  Case 5 ' Trace: Ghi ra console (Ouput)
    With Trace.Listeners
      .Clear()
      .Add(New TextWriterTraceListener(Console.Out))
    End With
```

```
            Trace.WriteLine("Testing Trace to Console.")
            Console.WriteLine(strDebug)
            Console.WriteLine("Note that the debug/trace strings are indented for ease
of reading.")
            Console.WriteLine("Option: Debug.Indent()")
            Trace.WriteLine("Done testing Trace to Console.")
            Trace.Flush()
            sbrPanel2.Text = "Wrote trace output to console"
        Case 6 ' Trace: Ghi vào file
            Dim strFile As String = "C:\TraceOutput.txt"
            Dim stmFile As Stream = File.Create(strfile)
            Dim twTextListener As New TextWriterTraceListener(stmFile)
            With Trace.Listeners
                .Clear()
                .Add(twTextListener)
            End With
            Trace.Write(strDebug)
            Trace.Flush()
            sbrPanel2.Text = "Wrote trace output to " & strFile
        Case 7 ' Trace: Ghi vào Event Log
            Dim logTraceListener As New _
                        EventLogTraceListener("ContrastVBNETwithVB6")
            With Trace.Listeners
                .Clear()
                .Add(logTraceListener)
            End With
            Trace.Write(strDebug)
            Trace.Flush()
            sbrPanel2.Text = "Wrote trace output to Event Log"
    End Select
    Debug.Unindent()
End Sub

Private Sub btnThreadFunction_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnThreadFunction.Click
    ' Tạo một tiểu trình để chạy một thủ tục, nghĩa là trong lúc thread hoạt
    ' động ta vẫn có thể làm việc khác.
    Dim newThread As New Thread(New ThreadStart(AddressOf CalledByThread))
    newThread.Start()
End Sub
```

```
    ' Thủ tục được gọi bởi tiểu trình
    Private Sub CalledByThread()
        txtThreadResult.Text = "This method is running on another thread, started at " &
    DateTime.Now.ToShortTimeString & vbCrLf
        txtThreadResult.Text &= "Gone to sleep for 5 seconds." & vbCrLf
        Thread.Sleep(5000)
        txtThreadResult.Text &= "Now resuming." & vbCrLf
        txtThreadResult.Text &= "Method called by thread now ended."
    End Sub
End Class
```

TimeCheck.vb

```
Option Strict On

Imports System.TimeSpan
Imports System.DateTime

Public Class TimeCheck

    Private m_dtmBegin As DateTime
    Private m_EndSpan As TimeSpan

    ' Bắt đầu tính thời gian
    Public Sub [Begin]()
        m_dtmBegin = DateTime.Now
    End Sub

    ' Kết thúc tính thời gian
    Public Sub [End]()
        m_EndSpan = DateTime.Now.Subtract(m_dtmBegin)
    End Sub

    Public ReadOnly Property Milliseconds() As Long
        Get
            Milliseconds = CType(m_EndSpan.TotalMilliseconds, Long)
        End Get
    End Property

End Class
```

frmControls.vb

```
Option Strict On
' Chú ý các thuộc tính của các control
Public Class frmControls
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "
    Public Sub New()
        MyBase.New()
        'This call is required by the Windows Form Designer.
        InitializeComponent()
        'Add any initialization after the InitializeComponent() call
    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents txtDockedBottom As System.Windows.Forms.TextBox
    Friend WithEvents TextBox2 As System.Windows.Forms.TextBox
    Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents txtCSZ As System.Windows.Forms.TextBox
    Friend WithEvents txtAddress As System.Windows.Forms.TextBox
    Friend WithEvents txtName As System.Windows.Forms.TextBox
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
```

```
    Me.txtDockedBottom = New System.Windows.Forms.TextBox()
    Me.TextBox2 = New System.Windows.Forms.TextBox()
    Me.TextBox1 = New System.Windows.Forms.TextBox()
    Me.Label3 = New System.Windows.Forms.Label()
    Me.Label2 = New System.Windows.Forms.Label()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.txtCSZ = New System.Windows.Forms.TextBox()
    Me.txtAddress = New System.Windows.Forms.TextBox()
    Me.txtName = New System.Windows.Forms.TextBox()
    Me.SuspendLayout()
    '
    'txtDockedBottom
    '
    Me.txtDockedBottom.Dock = System.Windows.Forms.DockStyle.Bottom
    Me.txtDockedBottom.Location = New System.Drawing.Point(0, 165)
    Me.txtDockedBottom.Multiline = True
    Me.txtDockedBottom.Name = "txtDockedBottom"
    Me.txtDockedBottom.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
    Me.txtDockedBottom.Size = New System.Drawing.Size(568, 184)
    Me.txtDockedBottom.TabIndex = 8
    Me.txtDockedBottom.Text = ""
    '
    'TextBox2
    '
    Me.TextBox2.Anchor = (System.Windows.Forms.AnchorStyles.Bottom Or
System.Windows.Forms.AnchorStyles.Right)
    Me.TextBox2.Location = New System.Drawing.Point(384, 128)
    Me.TextBox2.Name = "TextBox2"
    Me.TextBox2.Size = New System.Drawing.Size(56, 20)
    Me.TextBox2.TabIndex = 6
    Me.TextBox2.Text = ""
    '
    'TextBox1
    '
    Me.TextBox1.Anchor = (System.Windows.Forms.AnchorStyles.Bottom Or
System.Windows.Forms.AnchorStyles.Right)
    Me.TextBox1.Location = New System.Drawing.Point(448, 128)
    Me.TextBox1.Name = "TextBox1"
    Me.TextBox1.Size = New System.Drawing.Size(104, 20)
    Me.TextBox1.TabIndex = 7
```

```
    Me.TextBox1.Text = ""
    '
    'Label3
    '
    Me.Label3.Anchor = (System.Windows.Forms.AnchorStyles.Bottom Or
System.Windows.Forms.AnchorStyles.Left)
    Me.Label3.AutoSize = True
    Me.Label3.Location = New System.Drawing.Point(8, 128)
    Me.Label3.Name = "Label3"
    Me.Label3.Size = New System.Drawing.Size(78, 13)
    Me.Label3.TabIndex = 4
    Me.Label3.Text = "City, State, Zip"
    '
    'Label2
    '
    Me.Label2.AutoSize = True
    Me.Label2.Location = New System.Drawing.Point(8, 40)
    Me.Label2.Name = "Label2"
    Me.Label2.Size = New System.Drawing.Size(46, 13)
    Me.Label2.TabIndex = 2
    Me.Label2.Text = "Address"
    '
    'Label1
    '
    Me.Label1.AutoSize = True
    Me.Label1.Location = New System.Drawing.Point(8, 16)
    Me.Label1.Name = "Label1"
    Me.Label1.Size = New System.Drawing.Size(34, 13)
    Me.Label1.TabIndex = 0
    Me.Label1.Text = "Name"
    '
    'txtCSZ
    '
    Me.txtCSZ.Anchor = ((System.Windows.Forms.AnchorStyles.Bottom Or
System.Windows.Forms.AnchorStyles.Left) _
            Or System.Windows.Forms.AnchorStyles.Right)
    Me.txtCSZ.Location = New System.Drawing.Point(88, 128)
    Me.txtCSZ.Name = "txtCSZ"
    Me.txtCSZ.Size = New System.Drawing.Size(288, 20)
    Me.txtCSZ.TabIndex = 5
```

```
        Me.txtCSZ.Text = ""
        '
        'txtAddress
        '
        Me.txtAddress.Anchor = (((System.Windows.Forms.AnchorStyles.Top Or
    System.Windows.Forms.AnchorStyles.Bottom) _
                    Or System.Windows.Forms.AnchorStyles.Left) _
                    Or System.Windows.Forms.AnchorStyles.Right)
        Me.txtAddress.Location = New System.Drawing.Point(88, 40)
        Me.txtAddress.Multiline = True
        Me.txtAddress.Name = "txtAddress"
        Me.txtAddress.Size = New System.Drawing.Size(464, 80)
        Me.txtAddress.TabIndex = 3
        Me.txtAddress.Text = ""
        '
        'txtName
        '
        Me.txtName.Anchor = ((System.Windows.Forms.AnchorStyles.Top Or
    System.Windows.Forms.AnchorStyles.Left) _
                    Or System.Windows.Forms.AnchorStyles.Right)
        Me.txtName.Location = New System.Drawing.Point(88, 16)
        Me.txtName.Name = "txtName"
        Me.txtName.Size = New System.Drawing.Size(464, 20)
        Me.txtName.TabIndex = 1
        Me.txtName.Text = ""
        '
        'frmControls
        '
        Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
        Me.ClientSize = New System.Drawing.Size(568, 349)
        Me.Controls.AddRange(New System.Windows.Forms.Control() {Me.TextBox2,
    Me.TextBox1, Me.Label3, Me.Label2, Me.Label1, Me.txtCSZ, Me.txtAddress,
    Me.txtName, Me.txtDockedBottom})
        Me.Name = "frmControls"
        Me.Text = "Anchoring and Docked Controls"
        Me.ResumeLayout(False)

    End Sub
#End Region
End Class
```

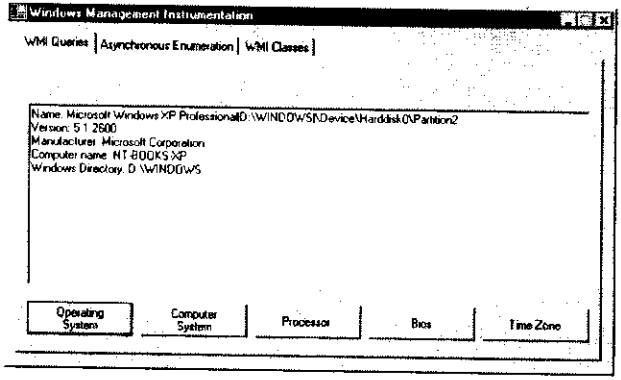
frmGraphics.vb

```
Imports System.Drawing
Imports System.Drawing.Drawing2D

Public Class frmGraphics
    Inherits System.Windows.Forms.Form
    Private drawX As Integer
    Private drawY As Integer

' Code phát sinh form được bỏ qua

    Private Sub miDrawLine_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles miDrawLine.Click
        picDrawing.CreateGraphics().Clear(Me.BackColor)
        Dim g As Graphics = picDrawing.CreateGraphics()
        Dim p As New Pen(Color.Blue, 6)
        g.DrawLine(p, 100, 100, 50, 150)
        g.Dispose()
    End Sub

    Private Sub frmGraphics_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        drawX = CInt((ClientSize.Width / 2) + 50)
        drawY = CInt((ClientSize.Height / 2) + 50)
    End Sub

    Private Sub miDrawCircle_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles miDrawCircle.Click
        picDrawing.CreateGraphics().Clear(Me.BackColor)
        Dim g As Graphics = picDrawing.CreateGraphics()
        Dim p As New Pen(Color.Red, 3)
        g.DrawEllipse(p, 120, 120, 100, 100)
        g.Dispose()
    End Sub

    Private Sub miDrawRectangle_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles miDrawRectangle.Click
        picDrawing.CreateGraphics().Clear(Me.BackColor)
        Dim g As Graphics = picDrawing.CreateGraphics()
```

```
        Dim p As New Pen(Color.Green, 3)
        g.DrawRectangle(p, 150, 150, 100, 100)
        g.Dispose()
    End Sub

    Private Sub miDrawText_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles miDrawText.Click
        picDrawing.CreateGraphics().Clear(Me.BackColor)
        Dim g As Graphics = picDrawing.CreateGraphics()
        g.DrawString("VB.NET", New Font("Arial", 20), Brushes.Blue, 135, 135)
        g.Dispose()
    End Sub

    Private Sub miDrawFancyText_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles miDrawFancyText.Click
        picDrawing.CreateGraphics().Clear(Me.BackColor)
        Dim g As Graphics = picDrawing.CreateGraphics()
        Dim b1 As New SolidBrush(Color.FromArgb(80, 10, 255, 100))
        g.DrawString("VB.NET", New Font("Arial", 30), b1, 135, 135)
        Dim b2 As New SolidBrush(Color.FromArgb(80, 100, 255, 100))
        g.DrawString("VB.NET", New Font("Arial", 20), b2, 150, 150)
        Dim b3 As New SolidBrush(Color.FromArgb(80, 10, 255, 100))
        g.DrawString("VB.NET", New Font("Arial", 10), b3, 170, 170)
        g.Dispose()
    End Sub
End Class
```

91. Windows Management Instrumentation

Chương trình này chỉ cách sử dụng Windows Management Instrumentation (WMI) trong Visual Basic .NET.



Các chức năng chính:

- **WMI Queries:** Sử dụng class ManagementObjectSearcher để truy vấn các thông tin hệ thống: Operating System Name, Version, Manufacturer, Computer name, và Windows Directory; Computer System Manufacturer, Model, System Type, và Total Physical Memory; Processor name; Bios version; và Time zone.
- **Asynchronous Enumerations:** Trong nhiều trường hợp WMI được sử dụng để lấy một lượng lớn dữ liệu. Trong các trường hợp như vậy thao tác được làm một cách không đồng bộ, cho phép giao diện người sử dụng và các công việc khác được tiếp tục. Phần này chỉ cách lấy danh sách các đĩa logic và các thuộc tính của chúng sử dụng class ManagementOperationObserver, với một sự kiện callback.
- **WMI Classes:** Liệt kê tất cả các class WMI cha và tất cả các class con sử dụng các class ManagementClass và EnumerationOptions.

Project: **Windows Management Instrumentation**

frmMain.vb

```
Option Strict On

Imports System

' Thêm một Reference đến namespace, và Imports để sử dụng các đối tượng WMI
Imports System.Management

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Const CRLF As String = Chr(13) & Chr(10)

' Code phát sinh form được bỏ qua

    ' Lấy thông tin BIOS
    Private Sub btnBios_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnBios.Click
        ' Sử dụng câu lệnh SelectQuery ở vị trí của lệnh SELECT
        Dim query As New SelectQuery("Win32_bios")
```

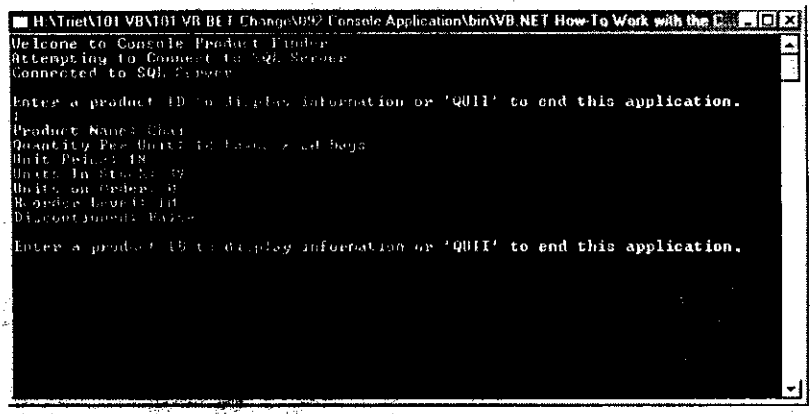
```
        ' ManagementObjectSearcher lấy một tập hợp các đối tượng WMI dựa trên query
        Dim search As New ManagementObjectSearcher(query)

        ' Hiển thị từng mục của Win32_bios
        Dim info As ManagementObject
        For Each info In search.Get()
            txtOutput.Text = "Bios version: " & info("version").ToString() & CRLF
        Next
    End Sub

    ' Lấy thông tin hệ thống
    Private Sub btnComputerSystem_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnComputerSystem.Click
        ' Trong trường hợp này sử dụng lệnh SELECT thay cho SelectQuery
        Dim search As New ManagementObjectSearcher( _
                            "SELECT * FROM Win32_ComputerSystem")

        ' Hiển thị
        Dim info As ManagementObject
        For Each info In search.Get()
            txtOutput.Text = "Manufacturer: " & info("manufacturer").ToString() & CRLF
            txtOutput.Text += "Model: " & info("model").ToString() & CRLF
            txtOutput.Text += "System Type: " & info("systemtype").ToString() & CRLF
            txtOutput.Text += "Total Physical Memory: " & _
                info("totalphysicalmemory").ToString() & CRLF
        Next
    End Sub

    ' Lấy các class WMI
    Private Sub btnClassEnum_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnClassEnum.Click
        lstWMIClasses.Items.Clear()
        Dim root As New ManagementClass()

        ' Nếu có đánh dấu checkbox Subclasses thì ta lấy tất cả các class con
        Dim options As New EnumerationOptions()
        options.EnumerateDeep = chkIncludeSubclasses.Checked

        ' Hiển thị các class WMI
        Dim info As ManagementObject
```

```
    For Each info In root.GetSubclasses(options)
      lstWMIClasses.Items.Add(info("__Class"))
    Next
  End Sub

  ' Lấy thông tin hệ điều hành
  Private Sub btnOperatingSytem_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnOperatingSytem.Click
    Dim search As New ManagementObjectSearcher( _
                    "SELECT * FROM Win32_OperatingSystem")

    Dim info As ManagementObject
    For Each info In search.Get()
      txtOutput.Text = "Name: " & info("name").ToString() & CRLF
      txtOutput.Text += "Version: " & info("version").ToString() & CRLF
      txtOutput.Text += "Manufacturer: " & info("manufacturer").ToString() & CRLF
      txtOutput.Text += "Computer name: " & info("csname").ToString() & CRLF
      txtOutput.Text += "Windows Directory: " & _
        info("windowsdirectory").ToString() & CRLF
    Next
  End Sub

  ' Thông tin bộ vi xử lý
  Private Sub btnProcessor_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnProcessor.Click
    Dim query As New SelectQuery("Win32_processor")
    Dim search As New ManagementObjectSearcher(query)
    Dim info As ManagementObject
    For Each info In search.Get()
      txtOutput.Text = "Processor: " & info("caption").ToString() & CRLF
    Next
  End Sub

  ' Thực hiện việc lấy các ổ đĩa không đồng bộ
  Private Sub btnStartEnum_Click(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles btnStartEnum.Click
    lvDriveList.Items.Clear()
    Dim disks As New ManagementClass("Win32_LogicalDisk")
```

```
        ' ManagementOperationObserver đảm nhiệm việc không đồng bộ bằng cách
        ' sử dụng một hàm callback
        Dim observer As New ManagementOperationObserver()
        ' Thêm một event handler cho observer để gọi cho mỗi ổ đĩa logic
        AddHandler observer.ObjectReady, AddressOf OnEnumObjectReady
        ' Trả về tập hợp tất cả các đĩa logic
        disks.GetInstances(observer)
    End Sub

    ' Lấy thông tin Time Zone
    Private Sub btnTimeZone_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnTimeZone.Click
        Dim query As New SelectQuery("Win32_timezone")
        Dim search As New ManagementObjectSearcher(query)
        Dim info As ManagementObject
        For Each info In search.Get()
            txtOutput.Text = "Time zone: " & info("caption").ToString() & CRLF
        Next
    End Sub

    ' Thủ tục callback sử dụng trong btnStartEnum_Click
    Sub OnEnumObjectReady(ByVal sender As Object, ByVal e As
    ObjectReadyEventArgs)
        ' Tạo item mới để cho vào DriveList ListView, và ký tự ổ đĩa
        Dim item As New ListViewItem(e.NewObject("Name").ToString(), 0)
        ' Nếu không có tên đĩa (A:, CD-ROM không có đĩa) thì hiển thị ký tự ổ đĩa
        If Not IsNothing(e.NewObject("VolumeName")) Then
            item.SubItems.Add(e.NewObject("VolumeName").ToString())
            item.SubItems.Add(e.NewObject("Size").ToString() & " bytes")
            ' Nếu là 0 byte, hiển thị (none)
            If e.NewObject("FreeSpace").ToString() <> "0" Then
                item.SubItems.Add(e.NewObject("FreeSpace").ToString() & " bytes")
            Else
                item.SubItems.Add("(none)")
            End If
        End If
        lvDriveList.Items.Add(item)
    End Sub
End Class
```

92. Làm việc với các ứng dụng Console

Chương trình này tạo một ứng dụng console (giao tiếp bằng dòng lệnh) sử dụng cơ sở dữ liệu.



Các chức năng chính:

- Chương trình console này sử dụng SQL Server, đọc bảng Products trong database Northwind vào một dataset. Khi chương trình chạy, nó nhận input từ người sử dụng có thể là Product ID hoặc 'quit'. Nếu nhập vào 'quit' thì chương trình sẽ thoát. Nếu nhập vào một ID đúng (số nguyên) thì thông tin của Product được hiển thị.

Project: **Console Application**

ModMain.vb

```
Option Strict On
Imports System.Data.SqlClient

Module modMain
    Private dsProducts As New DataSet()
    Private dvProducts As DataView

    Private PROMPT_MESSAGE As String = "Enter a product ID to display information
or 'QUIT' to end this application."
    Private WELCOME_MESSAGE As String = "Welcome to Console Product Finder"

    Private Const SQL_CONNECTION_STRING As String = _
        "Server=localhost;" & _
        "DataBase=Northwind;" & _
        "Integrated Security=SSPI"
```

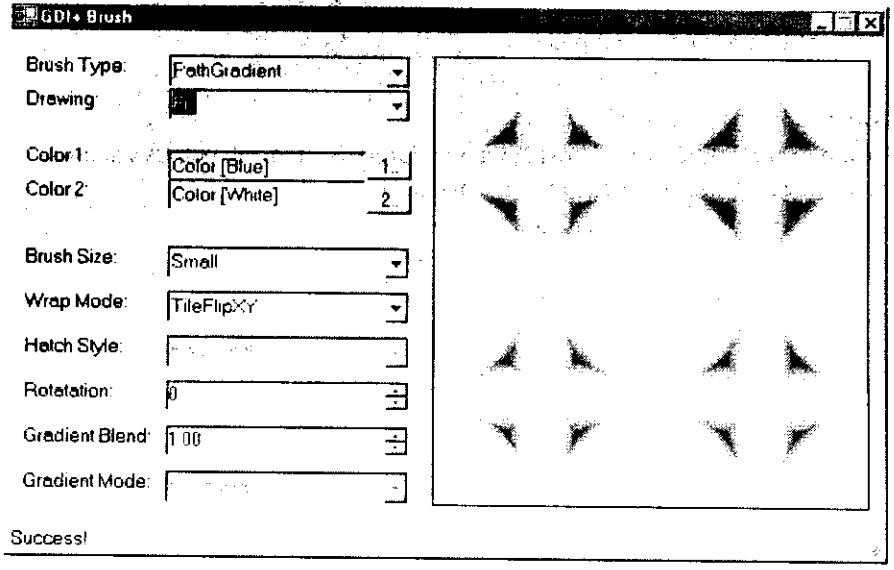
```
Private Const MSDE_CONNECTION_STRING As String = _
    "Server=(local)\NetSDK;" & _
    "DataBase=Northwind;" & _
    "Integrated Security=SSPI"

Sub Main()
    Dim strInput As String
    Dim indexData As Integer

    Console.WriteLine(WELCOME_MESSAGE)
     ' Kết nối vào SQL Server, thiết lập DataSet chứa bảng Products và tạo
     ' một Stored View của bảng Products trên trường ProductID
    myConnect()

    Console.WriteLine()
    Console.WriteLine(PROMPT_MESSAGE)

    strInput = UCase(Console.ReadLine())
    While strInput <> "QUIT"

        ' Kiểm tra để bảo đảm Product ID là một số
        While Not (IsNumeric(strInput) Or UCase(strInput) = "QUIT")
            Console.WriteLine("A numeric product ID is required.")
            Console.WriteLine("Please reenter the Product ID or QUIT to continue.")
            strInput = Console.ReadLine()
        End While

        ' Thoát
        If UCase(strInput) = "QUIT" Then
            End
        End If

        ' Hiển thị thông tin Product nếu số nhập vào là một Product ID
        indexData = dvProducts.Find(strInput)
        If indexData = -1 Then
            Console.WriteLine("No product found.")
        Else
            Console.Write("Product Name: ")
            Console.WriteLine(dvProducts(indexData)("ProductName"))
```

```
            Console.Write("Quantity Per Unit: ")
            Console.WriteLine(dvProducts(indexData)("QuantityPerUnit"))
            Console.Write("Unit Price: ")
            Console.WriteLine(dvProducts(indexData)("UnitPrice"))
            Console.Write("Units In Stock: ")
            Console.WriteLine(dvProducts(indexData)("UnitsInStock"))
            Console.Write("Units on Order: ")
            Console.WriteLine(dvProducts(indexData)("UnitsOnOrder"))
            Console.Write("Reorder Level: ")
            Console.WriteLine(dvProducts(indexData)("ReorderLevel"))
            Console.Write("Discontinued: ")
            If CBool(dvProducts(indexData)("Discontinued")) = False Then
               Console.WriteLine("False")
            Else
               Console.WriteLine("True")
            End If
         End If
         Console.WriteLine()
         Console.WriteLine(PROMPT_MESSAGE)
         strInput = UCase(Console.ReadLine())
      End While
      End
   End Sub

   Sub myConnect()
      Dim strConnection As String = SQL_CONNECTION_STRING
       ' Hiển thị thông báo cho biết đang kết nối vào SQL Server
      Console.WriteLine("Attempting to Connect to SQL Server ")

       ' Kết nối vào SQL Server trước, nếu không thành công thì kết nối vào MSDE
      Dim IsConnecting As Boolean = True
      While IsConnecting
         Try

            ' Class SqlConnection cho phép trao đổi với SQL Server
            Dim northwindConnection As New SqlConnection(strConnection)

            ' SqlDataAdapter được sử dụng để thiết lập một DataSet
            Dim ProductAdapter As New SqlDataAdapter( _
```

```
            "SELECT * " & "FROM products", northwindConnection)

            ' Thiết lập DataSet với thông tin trong bảng Products. Vì DataSet có
            ' thể chứa nhiều tập kết quả, do đó ta cần đặt tên cho mỗi tập kết quả.
            ProductAdapter.Fill(dsProducts, "Products")

            ' Tạo DataView, sử dụng constructor để chỉ định điều kiện sắp xếp
            dvProducts = New DataView(dsProducts.Tables("products"), "", _
                                      "ProductID ASC", DataViewRowState.OriginalRows)

            ' Dữ liệu đã được lấy thành công, thoát vòng lặp
            IsConnecting = False

            ' Xử lý khi kết nối xảy ra lỗi
        Catch exc As Exception
            If strConnection = SQL_CONNECTION_STRING Then
                strConnection = MSDE_CONNECTION_STRING
                Console.WriteLine("Attempting to Connecting to MSDE")
            Else
                Console.WriteLine("To run this sample, you must have SQL " & _
                "or MSDE with the Northwind database installed. ")
                End
            End If
        End Try
    End While

    Console.WriteLine("Connected to SQL Server")
  End Sub
End Module
```

93. Sử dụng các GDI+ Brush

Chương trình này cho thấy nhiều tính năng khi sử dụng đối Brush trong GDI+.



Các chức năng chính:

- Người sử dụng có thể chọn nhiều tham số khác nhau để hiển thị Brush. Sau mỗi lần chọn tham số, hình ảnh được vẽ lại trong một PictureBox. Một số tính năng cao cấp được kết hợp với các đối tượng PathLinearGradient.

Project: **GDI+ Brush**

frmMain.vb

```
Option Strict On

' Drawing2D cần sử dụng cho các Brush sẽ sử dụng
Imports System.Drawing.Drawing2D

Public Class frmMain
    Inherits System.Windows.Forms.Form

    Dim m_Brush As Brush
    Dim m_Color1 As Color = Color.Blue ' foreground color
```

```
    Dim m_Color2 As Color = Color.White ' background color
    Dim m_Pen As Pen
    Dim m_BrushSize As Rectangle
    Dim myGraphics As Graphics

' Code phát sinh form được bỏ qua

    ' Chọn màu cho m_Color1
    Private Sub btnSetColor1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSetColor1.Click
        Dim cdlg As New ColorDialog()

        If cdlg.ShowDialog() = DialogResult.OK Then
            m_Color1 = cdlg.Color
            txtColor1.Text = cdlg.Color.ToString()
            txtColor1.BackColor = cdlg.Color
        End If

    End Sub

    ' Chọn màu cho m_Color2
    Private Sub btnSetColor2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSetColor2.Click
        Dim cdlg As New ColorDialog()

        If cdlg.ShowDialog() = DialogResult.OK Then
            m_Color2 = cdlg.Color
            txtColor2.Text = cdlg.Color.ToString()
            txtColor2.BackColor = cdlg.Color
        End If
    End Sub

    ' Thay đổi kích thước của brush
    Private Sub cboBrushSize_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cboBrushSize.SelectedIndexChanged
        Select Case cboBrushSize.Text
            Case "Large"
                m_BrushSize = New Rectangle(0, 0, _
                    picDemoArea.Width, picDemoArea.Height)
```

```
        Case "Medium"
            m_BrushSize = New Rectangle(0, 0, _
                picDemoArea.Width \ 2, picDemoArea.Height \ 2)
        Case "Small"
            m_BrushSize = New Rectangle(0, 0, _
                picDemoArea.Width \ 4, picDemoArea.Height \ 4)
    End Select

    ' Gọi RedrawPicture
    RedrawPicture(cboBrushSize, New EventArgs())
End Sub

Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Dim i As Integer
    ' Gán BrushSize bằng với toàn PictureBox
    m_BrushSize = New Rectangle(0, 0, picDemoArea.Width, picDemoArea.Height)

    ' Cho các giá trị thích hợp vào các ComboBox
    ' cboWrapMode
    cboWrapMode.Items.Add(WrapMode.Clamp)
    cboWrapMode.Items.Add(WrapMode.Tile)
    cboWrapMode.Items.Add(WrapMode.TileFlipX)
    cboWrapMode.Items.Add(WrapMode.TileFlipY)
    cboWrapMode.Items.Add(WrapMode.TileFlipXY)

    ' Giá trị HatchStyle.Max không đúng trong .NET. Giá trị chính xác là 52
    Const maxHatchStyle As Integer = 52
    For i = HatchStyle.Min To maxHatchStyle
        cboHatchStyle.Items.Add(CType(i, HatchStyle))
    Next

    ' GradientStyles
    cboGradientMode.Items.Add(LinearGradientMode.BackwardDiagonal)
    cboGradientMode.Items.Add(LinearGradientMode.ForwardDiagonal)
    cboGradientMode.Items.Add(LinearGradientMode.Horizontal)
    cboGradientMode.Items.Add(LinearGradientMode.Vertical)
End Sub
```

```
Private Sub RedrawPicture(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cboBrushType.SelectedIndexChanged,
cboDrawing.SelectedIndexChanged, txtColor1.TextChanged,
cboWrapMode.SelectedIndexChanged, cboHatchStyle.SelectedIndexChanged,
txtColor2.TextChanged, cboGradientMode.SelectedIndexChanged,
nudRotation.ValueChanged, nudGradientBlend.ValueChanged

    ' Reset PictureBox
    picDemoArea.CreateGraphics().Clear(Color.White)
    picDemoArea.Refresh()

    ' Reset Status Bar
    sbrDrawingStatus.Text = ""

    ' Xây dựng Brush với các thuộc tính do người sử dụng chọn.
    Select Case cboBrushType.Text

        Case "Solid" ' sử dụng SolidBrush
            ' Cập nhật UI
            Me.cboBrushSize.Enabled = False
            Me.cboHatchStyle.Enabled = False
            Me.cboWrapMode.Enabled = False
            Me.txtColor2.Enabled = False
            Me.btnSetColor2.Enabled = False
            Me.nudGradientBlend.Enabled = False
            Me.nudRotation.Enabled = False
            Me.cboGradientMode.Enabled = False

            ' Tạo một SolidBrush dựa trên màu được chọn
            Dim mySolidBrush As New SolidBrush(m_Color1)

            ' Gán brush vừa tạo cho m_Brush
            m_Brush = mySolidBrush

        Case "Hatch" ' Sử dụng HatchBrush
            Me.cboBrushSize.Enabled = False
            Me.cboHatchStyle.Enabled = True
            Me.cboWrapMode.Enabled = False
            Me.txtColor2.Enabled = True
            Me.btnSetColor2.Enabled = True
```

```
        Me.nudGradientBlend.Enabled = False
        Me.nudRotation.Enabled = False
        Me.cboGradientMode.Enabled = False

        ' Tạo mới một HatchBrush sử dụng hai màu. Bởi vì thuộc tính HatchStyle
        ' là Read-Only nên HatchStyle phải được quy định lúc tạo HatchBrush
        Dim myHatchBrush As New HatchBrush( _
          CType(cboHatchStyle.SelectedItem, HatchStyle), _
                m_Color1, m_Color2)

        m_Brush = myHatchBrush

    Case "Texture" ' Sử dụng TextureBrush

        Me.cboBrushSize.Enabled = True
        Me.cboHatchStyle.Enabled = False
        Me.cboWrapMode.Enabled = True
        Me.txtColor2.Enabled = False
        Me.btnSetColor2.Enabled = False
        Me.nudGradientBlend.Enabled = False
        Me.nudRotation.Enabled = True
        Me.cboGradientMode.Enabled = False

        ' Tạo một TextureBrush dựa trên một bitmap.
        Dim myTextureBrush As New TextureBrush( _
          New Bitmap("..\waterlilies.jpg"), m_BrushSize)

        ' WrapMode quyết định cách trải brush
        myTextureBrush.WrapMode = CType(cboWrapMode.SelectedItem, _
                                        WrapMode)

        ' Phương thức RotateTransform xoay Brush
        myTextureBrush.RotateTransform(nudRotation.Value)

        ' Có thể sử dụng ScaleTransform để biến dạng brush
        ' myTextureBrush.ScaleTransform(0.5F, 2.0F)

        m_Brush = myTextureBrush
```

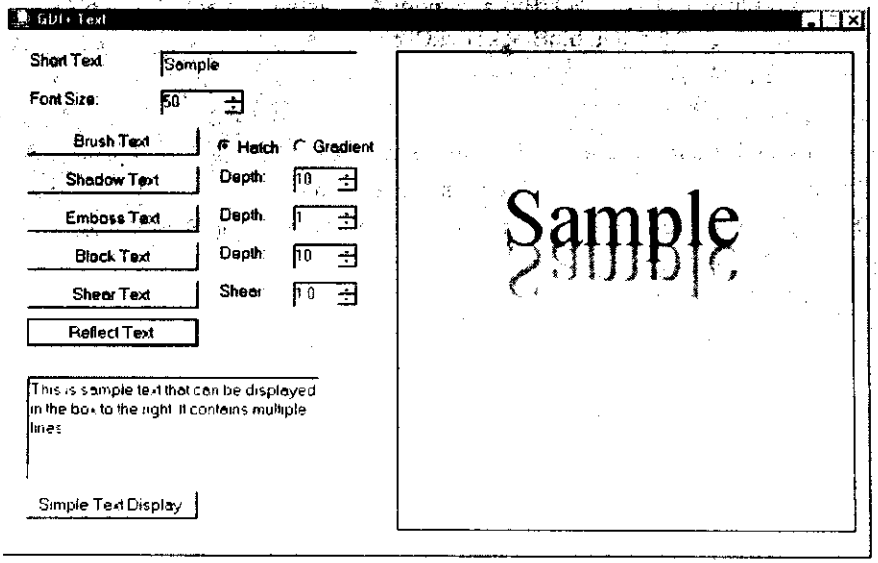
```
    Case "LinearGradient" ' Sử dụng LinearGradientBrush

      Me.cboBrushSize.Enabled = True
      Me.cboHatchStyle.Enabled = False
      Me.cboWrapMode.Enabled = True
      Me.txtColor2.Enabled = True
      Me.btnSetColor2.Enabled = True
      Me.nudGradientBlend.Enabled = True
      Me.nudRotation.Enabled = True
      Me.cboGradientMode.Enabled = True

        ' Tạo mới một LinearGradientBrush
      Dim myLinearGradientBrush As New LinearGradientBrush( _
        m_BrushSize, m_Color1, m_Color2, _
        CType(cboGradientMode.SelectedItem, LinearGradientMode))

        ' LinearGradientBrush không thể sử dụng giá trị Clamp cho WrapMode
    If CType(cboWrapMode.SelectedItem, WrapMode) <> WrapMode.Clamp Then
        myLinearGradientBrush.WrapMode = _
          CType(cboWrapMode.SelectedItem, WrapMode)
      Else
        Me.sbrDrawingStatus.Text += _
          "A Linear Gradient Brush cannot use the Clamp WrapMode."
      End If

      myLinearGradientBrush.RotateTransform(nudRotation.Value)

      myLinearGradientBrush.SetBlendTriangularShape( _
        nudGradientBlend.Value)

        ' Có thể sử dụng SetSigmaBellShape để chỉ định tâm của vùng tô
        ' myLinearGradientBrush.SetSigmaBellShape(0.2)

      m_Brush = myLinearGradientBrush

    Case "PathGradient" ' Sử dụng PathGradientBrush
      Me.cboBrushSize.Enabled = True
      Me.cboHatchStyle.Enabled = False
      Me.cboWrapMode.Enabled = True
```

```
        Me.txtColor2.Enabled = True
        Me.btnSetColor2.Enabled = True
        Me.nudGradientBlend.Enabled = True
        Me.nudRotation.Enabled = True
        Me.cboGradientMode.Enabled = False

        ' Định nghĩa một tập các điểm làm đường dẫn cho kiểu tô này.
        Dim pathPoint() As Point = {New Point(0, m_BrushSize.Height), _
            New Point(m_BrushSize.Width, m_BrushSize.Height), _
            New Point(m_BrushSize.Width, 0)}

        ' Tạo mới một PathGradientBrush dựa trên đường dẫn đã tạo
        Dim myPathGradientBrush As New PathGradientBrush(pathPoint)

        myPathGradientBrush.CenterColor = m_Color1
        myPathGradientBrush.SurroundColors = New Color() {m_Color2}


        myPathGradientBrush.WrapMode = _
          CType(cboWrapMode.SelectedItem, WrapMode)

        myPathGradientBrush.RotateTransform(nudRotation.Value)

        myPathGradientBrush.SetBlendTriangularShape( _
          nudGradientBlend.Value)

        m_Brush = myPathGradientBrush
    End Select

    ' Sử dụng m_Brush để vẻ ảnh thích hợp vào picDemoArea
    myGraphics = picDemoArea.CreateGraphics()

    ' Chọn kiểu vẽ
    Select Case cboDrawing.Text
      Case "Fill"
          ' Tô toàn bộ PictureBox
        myGraphics.FillRectangle(m_Brush, 0, 0, _
          picDemoArea.Width, picDemoArea.Height)
```

```
    Case "Ellipses"
            ' Vẽ hai elip giao nhau
          myGraphics.FillEllipse(m_Brush, picDemoArea.Width \ 10, _
            picDemoArea.Height \ 10, picDemoArea.Width \ 2, _
            picDemoArea.Height \ 2)
          myGraphics.FillEllipse(m_Brush, picDemoArea.Width \ 3, _
            picDemoArea.Height \ 3, picDemoArea.Width \ 2, _
            picDemoArea.Height \ 2)
        Case "Lines"
            ' Vẽ các đường thẳng cắt nhau
            ' Tạo Pen dựa trên Brush
          Dim myPen As New Pen(m_Brush, 40)

            ' Vẽ từ góc này đến góc kia
          myGraphics.DrawLine( _
            myPen, 0, 0, picDemoArea.Width, picDemoArea.Height)
          myGraphics.DrawLine(myPen, 0, 0, 0, picDemoArea.Height)
          myGraphics.DrawLine(myPen, 0, 0, picDemoArea.Width, 0)
          myGraphics.DrawLine(myPen, picDemoArea.Width, 0, _
            picDemoArea.Width, picDemoArea.Height)
          myGraphics.DrawLine(myPen, 0, picDemoArea.Height, _
            picDemoArea.Width, picDemoArea.Height)
          myGraphics.DrawLine( _
            myPen, picDemoArea.Width, 0, 0, picDemoArea.Height)
    End Select

    If Me.sbrDrawingStatus.Text = "" Then
      Me.sbrDrawingStatus.Text = "Success!"
    End If
  End Sub
End Class
```

94. GDI+ Text

Chương trình này cho thấy nhiều khả năng khi sử dụng GDI+ để làm việc với văn bản.



Các chức năng chính:

- Tạo văn bản dựa trên các đối tượng Brush.
- Tạo hiệu ứng đổ bóng (shadow).
- Tạo hiệu ứng trạm nổi hoặc khắc (embossed hoặc engrave).
- Tạo văn bản khối.
- Tạo hiệu ứng bị kéo (shear).
- Tạo hiệu ứng phản chiếu (reflect).
- Sử dụng DrawString để vẽ văn bản vào PictureBox.

Project: **GDI+ Text**

frmMain.vb

```
Option Strict On
Imports System.Drawing.Drawing2D

Public Class frmMain
    Inherits System.Windows.Forms.Form
```

```
' Code phát sinh form được bỏ qua

    ' Vẽ văn bản dạng khối
  Private Sub btnBlockText_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBlockText.Click
    Dim textSize As SizeF
    Dim g As Graphics
    Dim myBackBrush As Brush = Brushes.Black
    Dim myForeBrush As Brush = Brushes.Aquamarine
    Dim myFont As New Font("Times New Roman", Me.nudFontSize.Value,
FontStyle.Regular)
    Dim xLocation, yLocation As Single
    Dim i As Integer

     ' Tạo đối tượng Graphics từ PictureBox
    g = picDemoArea.CreateGraphics()
    g.Clear(Color.White)

     ' Lấy kích thước để vẽ Sample Text
    textSize = g.MeasureString(Me.txtShortText.Text, myFont)
    xLocation = (picDemoArea.Width - textSize.Width) / 2
    yLocation = (picDemoArea.Height - textSize.Height) / 2

    For i = CInt(nudBlockDepth.Value) To 0 Step -1
      g.DrawString(txtShortText.Text, myFont, myBackBrush, _
          xLocation - i, yLocation + i)
    Next
    g.DrawString(txtShortText.Text, myFont, myForeBrush, xLocation, yLocation)
  End Sub

    ' Vẽ Sample Text bằng Brush (Hatch hoặc Gradient)
  Private Sub btnBrushText_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnBrushText.Click

    Dim textSize As SizeF
    Dim g As Graphics
    Dim myBrush As Brush
    Dim gradientRectangle As RectangleF
```

```
    Dim myFont As New Font("Times New Roman", Me.nudFontSize.Value,
FontStyle.Regular)

    g = picDemoArea.CreateGraphics()
    g.Clear(Color.White)

    textSize = g.MeasureString(Me.txtShortText.Text, myFont)

     ' Tạo Brush
    If Me.optHatch.Checked Then
      myBrush = New HatchBrush(HatchStyle.DiagonalBrick, _
        Color.Blue, Color.Yellow)
    Else
       ' Tạo LinearGradientBrush
      gradientRectangle = New RectangleF(New PointF(0, 0), textSize)
      myBrush = New LinearGradientBrush(gradientRectangle, Color.Blue, _
        Color.Yellow, LinearGradientMode.ForwardDiagonal)
    End If

     ' Vẽ văn bản
    g.DrawString(txtShortText.Text, myFont, myBrush, _
        (picDemoArea.Width - textSize.Width) / 2, _
        (picDemoArea.Height - textSize.Height) / 2)
  End Sub

  ' Tạo văn bản Emboss. Đầu tiên vẽ màu đen và một khoảng cách, sau đó vẽ
  ' màu trắng, là màu nền hiện hành.
Private Sub btnEmboss_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnEmboss.Click
    Dim textSize As SizeF
    Dim g As Graphics
    Dim myBackBrush As Brush = Brushes.Black
    Dim myForeBrush As Brush = Brushes.White
    Dim myFont As New Font("Times New Roman", Me.nudFontSize.Value,
FontStyle.Regular)
    Dim xLocation, yLocation As Single

    g = picDemoArea.CreateGraphics()
    g.Clear(Color.White)
```

```
    textSize = g.MeasureString(Me.txtShortText.Text, myFont)

    xLocation = (picDemoArea.Width - textSize.Width) / 2
    yLocation = (picDemoArea.Height - textSize.Height) / 2

    ' Vẽ màu Den làm màu nền của văn bản, nếu thay bằng giá trị mudEmbossDepth
    ' bằng giá trị âm sẽ cho hiệu ứng Engrave
    g.DrawString(txtShortText.Text, myFont, myBackBrush, _
        xLocation + Me.nudEmbossDepth.Value, _
        yLocation + Me.nudEmbossDepth.Value)

    ' Vẽ văn bản màu trắng phía trên văn bản màu đen
    g.DrawString(txtShortText.Text, myFont, myForeBrush, xLocation, yLocation)
End Sub

' Phản chiếu văn bản
Private Sub btnReflectedText_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnReflectedText.Click
    Dim textSize As SizeF
    Dim g As Graphics
    Dim myBackBrush As Brush = Brushes.Gray
    Dim myForeBrush As Brush = Brushes.Black
    Dim myFont As New Font("Times New Roman", Me.nudFontSize.Value,
FontStyle.Regular)
    ' Sử dụng để lưu trạng thái hiện hành của Graphics
    Dim myState As GraphicsState
    ' Dùng để lưu vị trí
    Dim xLocation, yLocation As Single
    Dim textHeight As Single
    Dim i As Integer
    g = picDemoArea.CreateGraphics()
    g.Clear(Color.White)

    textSize = g.MeasureString(Me.txtShortText.Text, myFont)

    xLocation = (picDemoArea.Width - textSize.Width) / 2
    yLocation = (picDemoArea.Height - textSize.Height) / 2

    g.TranslateTransform(xLocation, yLocation)
```

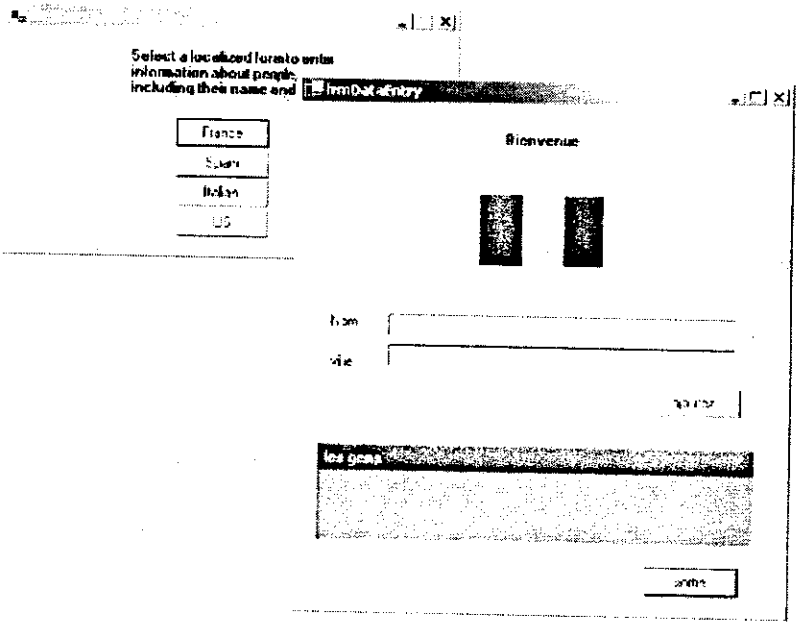
```
    Dim lineAscent As Integer
    Dim lineSpacing As Integer
    Dim lineHeight As Single

    lineAscent = myFont.FontFamily.GetCellAscent(myFont.Style)
    lineSpacing = myFont.FontFamily.GetLineSpacing(myFont.Style)
    lineHeight = myFont.GetHeight(g)
    textHeight = lineHeight * lineAscent / lineSpacing

    ' Bỏ các ghi chú bên dưới để phản chiếu thấp hơn các ký tự
    'Dim lineDescent As Integer
    'lineDescent = myFont.FontFamily.GetCellDescent(myFont.Style)
    'textHeight = lineHeight * (lineAscent + lineDescent) / lineSpacing

    ' Lưu trạng thái Graphics
    myState = g.Save()

    ' Để vẽ phản chiếu, sử dụng ScaleTransform với giá trị âm. Sử dụng -1 sẽ
    ' phản chiếu mà không có sự méo mó.
    g.ScaleTransform(1, -1.0F) ' Chỉ phản chiếu hướng Y
    g.DrawString(txtShortText.Text, myFont, myBackBrush, 0, -textHeight)

    ' Reset Graphics trước khi vẽ
    g.Restore(myState)
    ' Vẽ văn bản chính
    g.DrawString(txtShortText.Text, myFont, myForeBrush, 0, -textHeight)
End Sub

' Tạo Sample Text với một brush và đổ bóng
Private Sub btnShadowText_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShadowText.Click
    Dim textSize As SizeF
    Dim g As Graphics
    Dim myShadowBrush As Brush = Brushes.Gray
    Dim myForeBrush As Brush = Brushes.Black
    Dim myFont As New Font("Times New Roman", Me.nudFontSize.Value,
FontStyle.Regular)
    Dim xLocation, yLocation As Single ' Used for the location
```

```
    g = picDemoArea.CreateGraphics()
    g.Clear(Color.White)

    textSize = g.MeasureString(Me.txtShortText.Text, myFont)

    xLocation = (picDemoArea.Width - textSize.Width) / 2
    yLocation = (picDemoArea.Height - textSize.Height) / 2

     ' Vẽ bóng trước
    g.DrawString(txtShortText.Text, myFont, myShadowBrush, _
        xLocation + Me.nudShadowDepth.Value, _
        yLocation + Me.nudShadowDepth.Value)

     ' Vẽ văn bản chính phía trên bóng
    g.DrawString(txtShortText.Text, myFont, myForeBrush, xLocation, yLocation)
End Sub

 ' Tạo hiệu ứng kéo văn bản
Private Sub btnShearText_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnShearText.Click
    Dim textSize As SizeF
    Dim g As Graphics
    Dim myForeBrush As Brush = Brushes.Black
    Dim myFont As New Font("Times New Roman", Me.nudFontSize.Value,
FontStyle.Regular)
    Dim myTransform As Matrix ' Sử dụng để kéo văn bản
    Dim xLocation, yLocation As Single
    Dim textHeight As Single
    Dim i As Integer

    g = picDemoArea.CreateGraphics()
    g.Clear(Color.White)

    textSize = g.MeasureString(Me.txtShortText.Text, myFont)

    xLocation = (picDemoArea.Width - textSize.Width) / 2
    yLocation = (picDemoArea.Height - textSize.Height) / 2

    g.TranslateTransform(xLocation, yLocation)
```

```
    ' Thực hiện Shear
        myTransform = g.Transform
        myTransform.Shear(nudSkew.Value, 0)
        g.Transform = myTransform

        ' Vẽ văn bản
        g.DrawString(txtShortText.Text, myFont, myForeBrush, 0, 0)

    End Sub

    ' Hiển thị các dòng văn bản trên PictureBox
    Private Sub btnSimpleText_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnSimpleText.Click
        Dim textSize As SizeF
        Dim g As Graphics
        Dim myForeBrush As Brush = Brushes.Black
        Dim myFont As New Font("Times New Roman", Me.nudFontSize.Value,
    FontStyle.Regular)
        Dim textHeight As Single
        Dim i As Integer
        g = picDemoArea.CreateGraphics()
        g.Clear(Color.White)
        textSize = g.MeasureString(txtLongText.Text, myFont)
        ' Vẽ văn bản
        g.DrawString(txtLongText.Text, myFont, myForeBrush, _
            New RectangleF(0, 0, picDemoArea.Width, picDemoArea.Height))
    End Sub
End Class
```

95. Sử dụng các file Resource

Chương trình này chỉ cách sử dụng các file resource (file tài nguyên) và thuộc tính "localizable" của form để bản địa hóa ứng dụng.



Các chức năng chính:

- Chương trình cho thấy ta có thể địa phương hóa một ứng dụng bằng cách sử dụng các thuộc tính language và localizable của form. Chương trình sẽ hiển thị một form tương ứng với thông tin do người sử dụng chọn.

Chú ý:

Trước tiên, chọn thuộc tính localizable là True. Tiếp theo, tạo file resource cho từng địa phương bằng cách chọn thuộc tính Language. Trình thiết kế sẽ tự động tạo ra một file .rsx, sử dụng ResEditor.exe để soạn thảo file resource (Xem thêm về Resource trong MSDN).

Project: **Resource Files**

frmMain.vb

```
Option Strict On
Imports System.Threading
Imports System.Globalization


Public Class frmMain
    Inherits System.Windows.Forms.Form

' Code phát sinh form được bỏ qua

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button2.Click
        Thread.CurrentThread.CurrentUICulture = New CultureInfo("fr-FR")
        Dim frmData As New frmDataEntry()
        frmData.ShowDialog()
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button3.Click
        Thread.CurrentThread.CurrentUICulture = New CultureInfo("es-ES")
        Dim frmData As New frmDataEntry()
        frmData.ShowDialog()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button4.Click
        Thread.CurrentThread.CurrentUICulture = New CultureInfo("it-IT")
        Dim frmData As New frmDataEntry()
        frmData.ShowDialog()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click
        Thread.CurrentThread.CurrentUICulture = New CultureInfo("en-US")
        Dim frmData As New frmDataEntry()
        frmData.ShowDialog()
    End Sub
End Class
```

frmDataEntry.vb

```
Option Strict On
Imports System.Globalization

Public Class frmDataEntry
    Inherits System.Windows.Forms.Form
    Private dt As New DataTable("Names")

' Code phát sinh form được bỏ qua

    Private Sub cmdQuit_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles cmdQuit.Click
        Me.Close()
    End Sub

    Private Sub cmdSave_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles cmdSave.Click
        Dim myDataRow As DataRow

        ' Tạo một dòng mới cho DataTable
        myDataRow = dt.NewRow()
        ' Cho dữ liệu vào
        myDataRow.Item("Name") = txtName.Text
        myDataRow.Item("City") = txtCity.Text
        ' Cho dòng mới vào DataTable
        dt.Rows.Add(myDataRow)
    End Sub

    Private Sub frmDataEntry_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
        ' Tạo dao diện cho Grid
        dt.Columns.Add("Name")
        dt.Columns.Add("City")
        grdPeople.DataSource = dt
    End Sub
End Class
```

--- Hết ---

Coding Techniques

* Kết hợp Help vào ứng dụng * Tạo ứng dụng theo kiểu Explorer * Crystal Report Print Preview * Tạo Control động * Điều khiển Office * Các hộp thoại thông dụng * System Tray * Exception tùy biến * Truy xuất và duyệt dữ liệu * Định dạng DataGrid * Sắp xếp và lọc dữ liệu trên DataGrid * In ấn đơn giản * ListBox và ComboBox * Sử dụng các menu * Quản lý nhiều form cấp cao nhất (Top-Level) * Drag and Drop * Thừa kế Windows Forms * Validating TextBox * Tự vẽ Menu * Định dạng dữ liệu trong chuỗi. * Regular Expression * Clipboard * Tạo hộp thoại Login * Mã hóa và giải mã dữ liệu * Windows Role-Based Security * Thuật toán mã hóa Hash (băm) * Điều khiển Internet Explorer * Truy xuất File System * Theo dõi File System * Tạo ứng dụng Web nhập dữ liệu * Web Service * Master-Details Web Forms * Phân trang DataGrid trên Web Forms * Sử dụng các Web Service * Truyền dữ liệu nhị phân thông qua HTTP * Sử dụng mảng Array * Callback * DateTime và TimeSpan * Hướng đối tượng trong Visual Basic .NET * Chuỗi String * Try Catch Finally * Multithreading * TCP Remoting * Serializing và Deserializing các đối tượng * Kết dữ liệu vào ComboBox * Tạo ứng dụng Master-Details Windows Forms * Tạo ứng dụng Offline * Tạo một Form nhập liệu * Tạo và sử dụng một Database * Tạo ứng dụng theo mô hình 3-Tier * Đọc và ghi các hình ảnh vào Database * DataReader * DataView * Typed DataSet * ADO 2.6 trong .NET * Định dạng dữ liệu trước khi hiển thị và lưu vào database * Stored Procedure * Làm hoạt hình với GDI+ * Windows Service * Screen Saver với GDI+ * Trace Listener * Enterprise Service Component * File tạm thời * Cấu hình cho ứng dụng * Lấy thông tin hệ thống và môi trường. * MSMQ (Microsoft Message Queuing) * Process Viewer * Reflection * Gửi và nhận dữ liệu thông qua HTTP * Gửi Email * Service Manager * Stack Frame * System Events * XMLDocument Class * [GDI+] Pen * Tương tác với Windows Service * Xử lý hình ảnh với GDI+ * Phân chia một ứng dụng * Class Process * Đọc và ghi file Text * Tương tác với các Performance Counter * Scope, Override và Overload * Thread Pooling * Garbage Collector * Lập trình Socket * So sánh các control dùng trên Web Forms và Windows Forms * Công COM * Event Log * Win32 API * Một số lợi điểm của Visual Basic .NET * Windows Management Instrumentation * Ứng dụng Console * GDI+ Brush * GDI+ Text * Files Resource *

Giá : 125.000đ