



NHỮNG BÀI THỰC HÀNH

ISUAL BASIC

CĂN BẢN

NHÀ XUẤT BẢN THỐNG KÊ

01010101010101
01010101010101
10101010101010

Những bài thực hành Visual Basic .NET căn bản

NHỮNG BÀI THỰC HÀNH
Visual Basic
.NET
CĂN BẢN

Tổng hợp và biên soạn:

KS. ĐÌNH XUÂN LÂM

VN-GUIDE

NHÀ XUẤT BẢN THỐNG KÊ

Bài 1.

Cài đặt VB.NET trên WINDOWS 2000 Server

Yêu cầu hệ thống

Nếu bạn sử dụng toàn bộ phiên bản Visual Studio .NET Enterprise Architect, cấu hình tối thiểu đòi hỏi với máy tính bạn phải là:

- Processor: Pentium II, 450 MHz trở lên
- RAM: Nếu sử dụng Windows NT 4.0 Workstation cần 64Mb RAM trở lên, với Windows NT 4.0 Server là 160Mb RAM hoặc Microsoft Windows 2000 Professional là 96Mb, Microsoft Windows 2000 Server là 192 Mb, Windows XP Professional 160 Mb, Windows XP Home 96 Mb.
- Dung lượng đĩa cứng còn trống: 600 Mb trên đĩa hệ thống và 3Gb dùng để cài đặt ứng dụng.
- Ổ đĩa CD-ROM hoặc DVD.
- Video: Màn hình phải đặt ở chế độ phân giải tối thiểu là 256 Colors, 800x600.
- Mouse và các thiết bị tương thích khác.

Nếu bạn chỉ muốn cài đặt riêng Visual Basic .Net trong bộ Microsoft Visual Studio .NET, cấu hình tối thiểu máy tính của bạn phải là:

- Processor: Pentium II, 450 MHz trở lên

- RAM: Nếu sử dụng Windows NT 4.0 Workstation cần 64Mb RAM trở lên, với Windows NT 4.0 Server là 160Mb RAM hoặc Microsoft Windows 2000 Professional là 96Mb, Microsoft Windows 2000 Server là 192 Mb, Windows XP Professional 160 Mb, Windows XP Home 96 Mb.
- Dung lượng đĩa cứng còn trống: 600 Mb trên đĩa hệ thống và 1.5 Gb dùng để cài đặt ứng dụng.
- Ổ đĩa CD-ROM hoặc DVD.
- Video: Màn hình phải đặt ở chế độ phân giải tối thiểu là 256 Colors, 800x600.
- Mouse và các thiết bị tương thích khác.

Chuẩn bị cài đặt

Trước khi tiến hành cài đặt, bạn cần chuẩn bị cho mình một bộ đĩa cài đặt Microsoft Studio.Net Enterprise Architect gồm 7 đĩa CD.

Tiến hành kiểm tra xem máy tính của bạn đã cài đặt hệ điều hành Windows 2000 hay chưa.

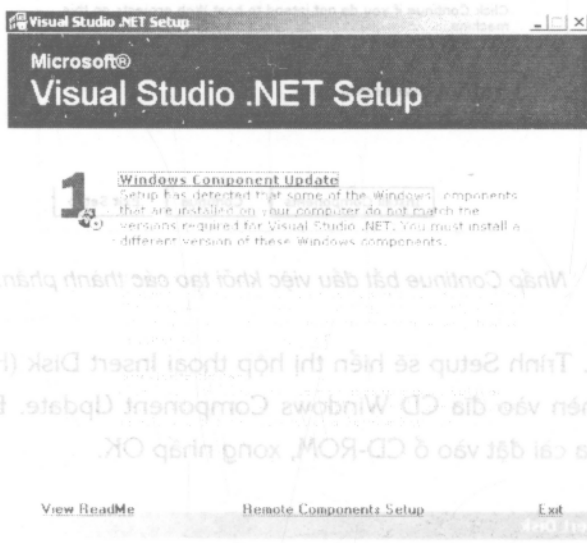
MÁCH NƯỚC

- ◆ Phiên bản Microsoft Studio.Net Enterprise Architect không thể cài đặt trên Microsoft Windows 95, Windows 98 hay Windows ME. Chúng chỉ có thể chạy được trên Windows NT 4.0, Windows XP, Windows 2000,.
 - ◆ Microsoft Studio.Net có nhiều phiên bản khác nhau như là Enterprise Architect, Enterprise Developer, Professional và Academic. Trong phạm vi tập sách này, chúng tôi sẽ sử dụng phiên bản Enterprise Architect.
 - ◆ Đối với những bạn thích sử dụng Visual Basic 6.0, vẫn có thể khởi tạo cùng lúc hai bộ Visual Studio .NET và Visual Studio 6.0 trên cùng một máy tính.
-

Tiến hành cài đặt

Bước 1. Khởi động Windows 2000 Server.

Bước 2. Đặt đĩa CD1 của bộ đĩa cài đặt Microsoft Studio.Net vào ổ đĩa CDROM. Trình Autorun trên đĩa CD sẽ tự động nạp chương trình Setup với màn hình như Hình 1.1.



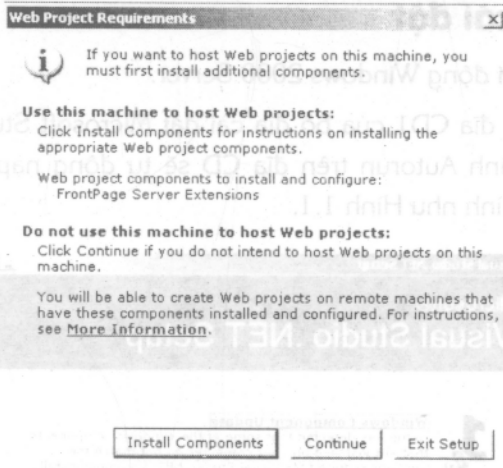
Hình 1.1 Màn hình bắt đầu cài đặt Microsoft Studio.Net.

Nhấp mục **Windows Component Update** để trình Setup dò và gỡ bỏ các thành phần không thích hợp với phiên bản Visual Studio .Net sẽ cài đặt, xong tự động cài đặt các thành phần mới.

MÁCH NƯỚC

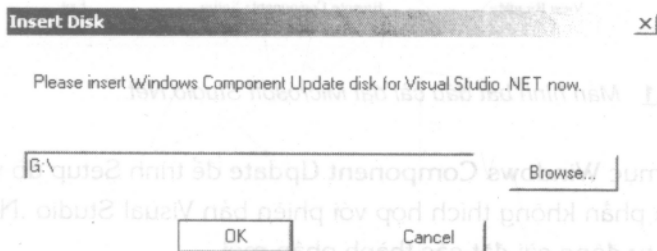
- ♦ Nếu trình Autorun không khởi động thì chọn lệnh Start ➡ Run và chạy tập tin Setup.exe từ đĩa CD1.

Bước 3. Nhấp nút Continue để tiếp tục công việc cài đặt (Hình 1.2).



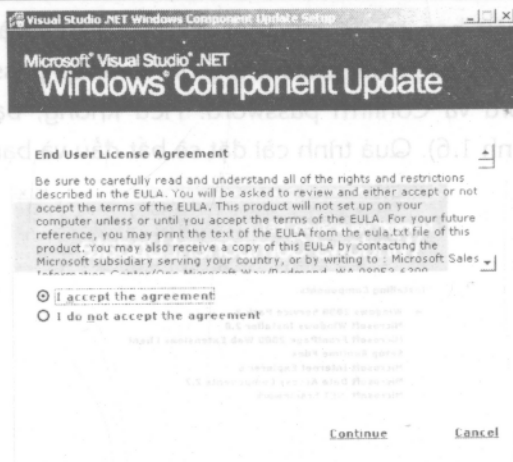
Hình 1.2 Nhấp *Continue* bắt đầu việc khởi tạo các thành phần.

Bước 4. Trình Setup sẽ hiển thị hộp thoại Insert Disk (Hình 1.3) yêu cầu bạn chèn vào đĩa CD Windows Component Update. Đưa đĩa CD5 trong bộ đĩa cài đặt vào ổ CD-ROM, xong nhấp OK.



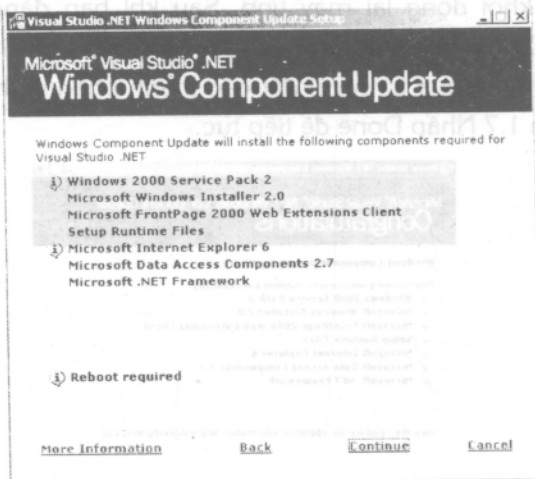
Hình 1.3 Hộp thoại Insert Disk.

Bước 5. Trong cửa sổ Visual Studio .Net Windows Component Update Setup, chọn mục I accept the agreement, xong nhấp Continue (Hình 1.4).



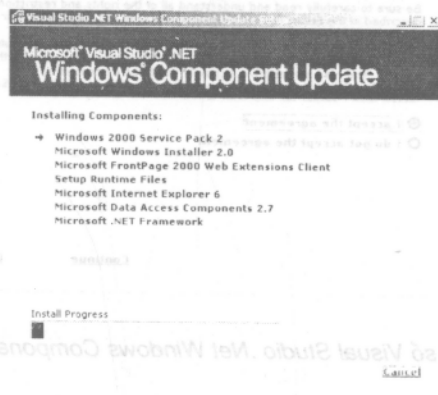
Hình 1.4 Cửa sổ Visual Studio .Net Windows Component Update Setup.

Bước 6. Tên các thành phần cần khởi tạo sẽ được liệt kê như Hình 1.5. Nhấp Continue để tiếp tục quá trình cài đặt.



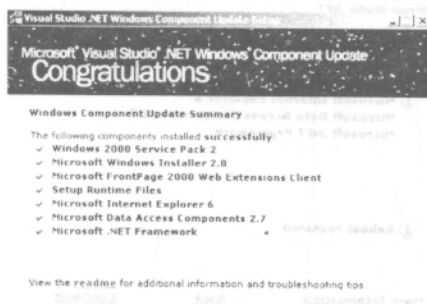
Hình 1.5 Tên các thành phần sẽ khởi tạo.

Bước 7. Nếu bạn muốn tạo một password truy cập ứng dụng thì đánh dấu chọn mục Automatically log on và nhập password 2 lần vào các ô Password và Confirm password. Nếu không, bạn có thể nhấp Install now (Hình 1.6). Quá trình cài đặt sẽ bắt đầu và bạn cứ việc đợi.



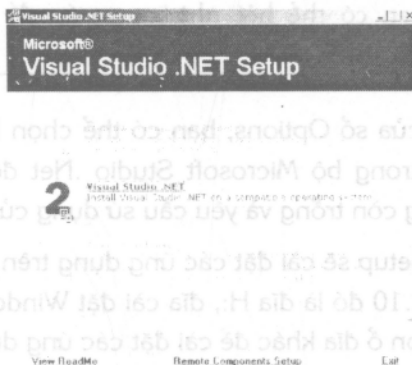
Hình 1.6 Cửa sổ cho biết tiến trình cài đặt các thành phần.

Sau khi cài đặt xong thành phần Windows 2000 Service Pack 2, trình Setup sẽ tự khởi động lại máy tính. Sau khi bạn đăng nhập vào lại Windows 2000 Server, quá trình cài đặt mới tiếp tục. Sau vài lần khởi động lại máy tính, quá trình cài đặt các thành phần sẽ hoàn tất với kết quả như Hình 1.7 Nhấp Done để tiếp tục.



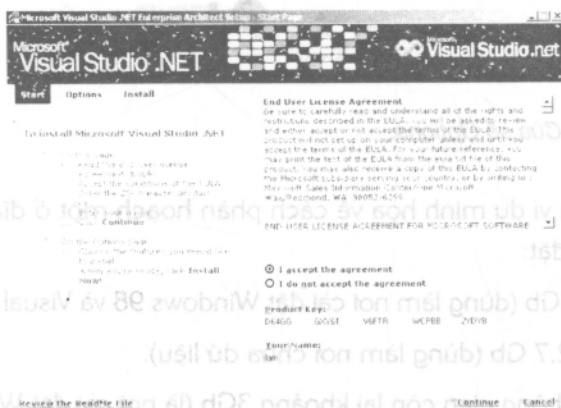
Hình 1.7 Quá trình cài đặt các thành phần đã hoàn tất.

Bước 8. Nhấp mục Visual Studio .Net (Hình 1.8) để bắt đầu phần cài đặt chính.



Hình 1.8 Bắt đầu cài đặt Visual Studio .NET.

Trình Setup sẽ hiển thị hộp thoại Insert Disk, yêu cầu bạn chèn trở lại đĩa CD1. Xong nhấp OK. Trang Start sẽ xuất hiện (Hình 1.9) để chờ bạn nhập Product Key và tên. Chọn mục I accept the agreement, nhập Product Key và tên của bạn, xong nhấp Continue.



Hình 1.9 Nhập Product Key và tên người dùng.

MÁCH NƯỚC

- ♦ Tùy thuộc phiên bản đã mua mà bạn cần nhập Product Key cho thích hợp. Bạn có thể hỏi nhà cung cấp để biết chính xác Product Key cần nhập.

Bước 9. Trong cửa sổ Options, bạn có thể chọn hết hoặc chỉ chọn một số ứng dụng trong bộ Microsoft Studio .Net để cài đặt, tùy vào dung lượng đĩa cứng còn trống và yêu cầu sử dụng của bạn.

Mặc định, trình Setup sẽ cài đặt các ứng dụng trên đĩa hệ thống (ứng với hình minh họa 1.10 đó là đĩa H:, đĩa cài đặt Windows 2000 Server), bạn cũng có thể chọn ổ đĩa khác để cài đặt các ứng dụng nếu muốn.



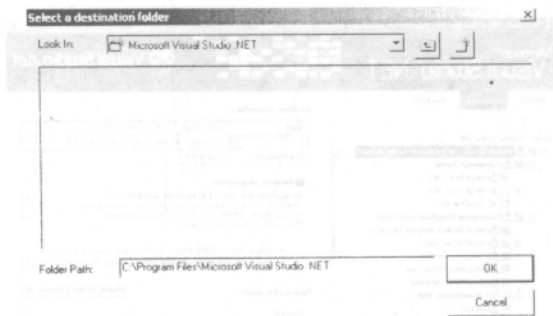
Hình 1.10 Cửa sổ Options.

Sau đây là ví dụ minh họa về cách phân hoạch một ổ đĩa cứng 10Gb trước khi cài đặt:

- Ổ C: 4Gb (dùng làm nơi cài đặt Windows 98 và Visual Studio .Net).
- Ổ D: 2.7 Gb (dùng làm nơi chứa dữ liệu).
- Ổ E: không gian còn lại khoảng 3Gb (là nơi cài đặt Windows 2000 Server và các thành phần hỗ trợ trong bộ Visual Studio .Net).

Ví dụ trong Hình 1.10 cho thấy để cài các ứng dụng trên đĩa H:, bạn cần 2218.18Mb, trong khi không gian trống trên đĩa H: còn lại không đủ để cài đặt, cụ thể là thiếu 1Gb. Trong trường hợp này, bạn có thể thực hiện một trong hai phương án sau: loại bỏ bớt một số ứng dụng chưa cần cài đặt để giảm yêu cầu dung lượng đĩa hay chuyển sang partition có dung lượng trống nhiều hơn.

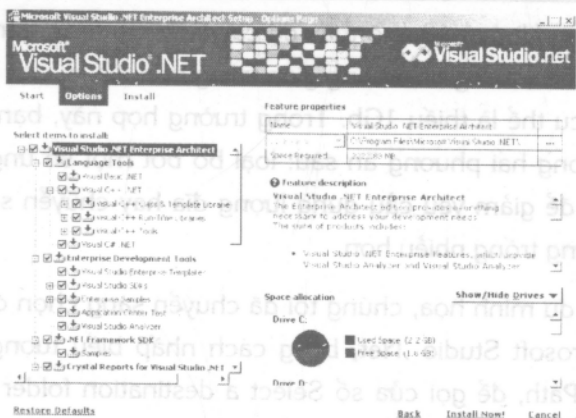
Trong ví dụ minh họa, chúng tôi đã chuyển sang chọn đĩa C làm nơi cài đặt Microsoft Studio .Net, bằng cách nhấp biểu tượng ... ở sau mục Local Path, để gọi cửa sổ Select a destination folder (Hình 1.11). Sửa lại đường dẫn cài đặt là C:\Program Files\Microsoft Visual .Net, xong nhấp OK.



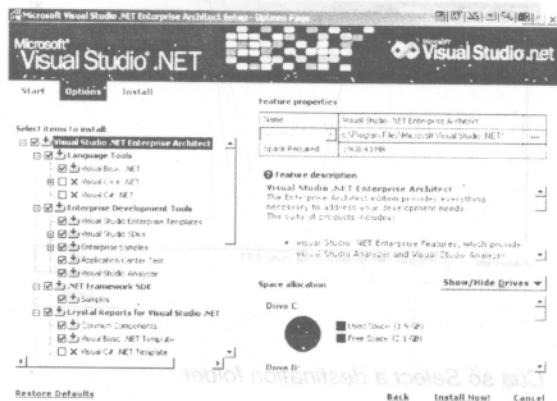
Hình 1.11 Cửa sổ *Select a destination folder*.

Trong Hình 1.12 cho thấy, dù đã chọn sang ổ C, bạn cũng còn thiếu chỗ vì chỉ còn 1.8 Gb trống trên đĩa C, trong khi không gian trống tối thiểu yêu cầu là 2.2 Gb. Trong tình huống này, cần loại bớt một số ứng dụng trong bộ Microsoft Visual Studio .Net mà bạn chưa cần sử dụng, chẳng hạn Visual C++ .Net, Visual C# .Net.

Bỏ chọn một số ứng dụng không cài đặt bằng cách nhấp vào ô chọn phía trước mỗi mục chọn (Hình 1.13a và Hình 1.13b).



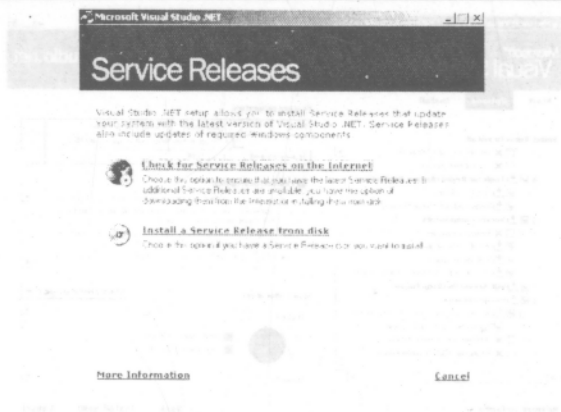
Hình 1.12 Đĩa C vẫn thiếu chỗ để cài đặt cả bộ Microsoft Visual Studio .Net.



Hình 1.13a Chọn lựa ứng dụng muốn cài đặt trong Microsoft Studio .Net.

MÁCH NƯỚC

- ◆ Nếu cài đặt trên đĩa C, bạn cần đảm bảo sau khi cài đặt xong, đĩa C vẫn còn một không gian trống đủ lớn. Trường hợp bạn ních đầy đĩa C, Windows của bạn có thể sẽ không khởi động được hoặc chạy rất chậm.



Hình 1.15 Nhấp “Install a Service Release from disk” để tiếp tục.

Giờ đây, công việc của bạn là chuẩn bị cho mình một ly café và đợi máy tính yêu cầu chèn đĩa CD kế tiếp cho đến khi quá trình cài đặt hoàn tất, vì bạn có thể phải đợi từ 1 đến vài giờ đồng hồ, tùy theo tốc độ máy tính của bạn.



Hình 1.16 Trình Setup đang tiến hành cài đặt.

CHÚ Ý

- ◆ Khi phân hoạch partition để cài đặt Windows Server 2000, bạn cần chú ý phân hoạch sao cho kích thước partition này đủ lớn, đảm bảo sau khi cài đặt xong Windows 2000 Server, partition phải có không gian trống trên 600 Mb trước khi cài đặt Visual Studio .Net.

Bài 2.

Chương trình Lunar

Mục đích chương trình

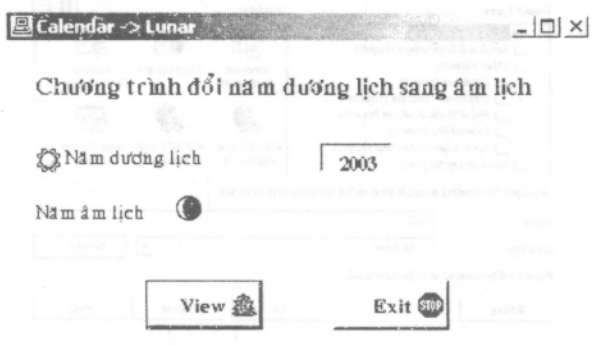
Thiết kế project Lunar thực hiện đổi một năm dương lịch bất kỳ thành năm âm lịch.

Bài thực hành này giúp bạn làm quen với việc thiết kế một project trong VB .Net, hiểu một số điều khiển căn bản trong Visual Basic .Net như Text box, Label, Button. Từ đó, rút ra điểm giống và khác nhau của các điều khiển này trong VB 6 so với Visual Basic .Net.

Hoạt động chương trình

Mỗi khi thi hành, chương trình sẽ hiển thị giao diện như Hình 2.1 và chờ người dùng nhập vào năm dương lịch. Chương trình sẽ tự đổi ra thành năm âm lịch và hiển thị lên màn hình khi người dùng nhấp nút View hay tổ hợp phím ALT+V.

Nhấn nút EXIT hay nhấn tổ hợp phím ALT+E để kết thúc chương trình.



Hình 2.1 Giao diện chương trình khi bắt đầu.

Hướng dẫn thiết kế

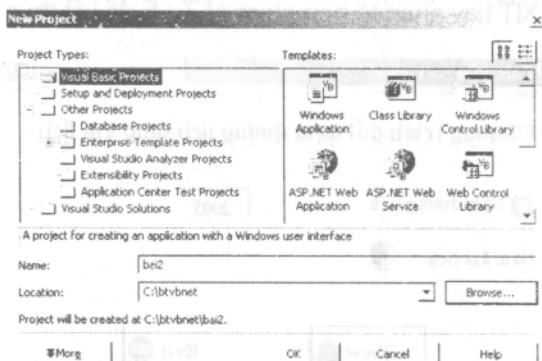
Bước 1. Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net.

Màn hình làm việc ban đầu của bạn sẽ giống như Hình 2.2.



Hình 2.2 Màn hình làm việc sau khi khởi động Microsoft Visual Studio .Net.

Bước 2. Nhấp nút New Project hay chọn lệnh File ➡ New Project để gọi cửa sổ New Project (Hình 2.3).



Hình 2.3 Cửa sổ New Project.

Chọn Project Types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.

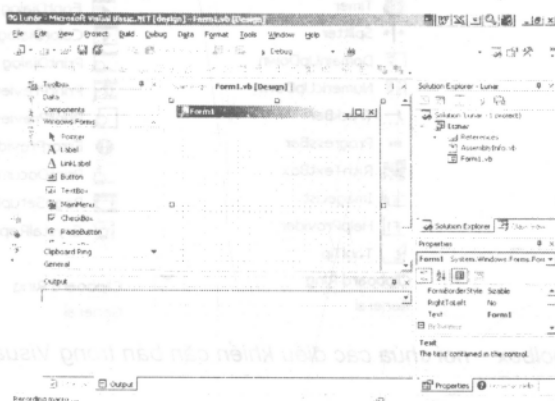
Đặt tên tập tin lưu project: Name = **LUNAR**

Định thư mục lưu project: Location = **C:\BTVBNET**

Nhấp tùy chọn Close Solution.

Xong nhấp OK.

Màn hình Project VB .Net ban đầu sẽ như Hình 2.4.



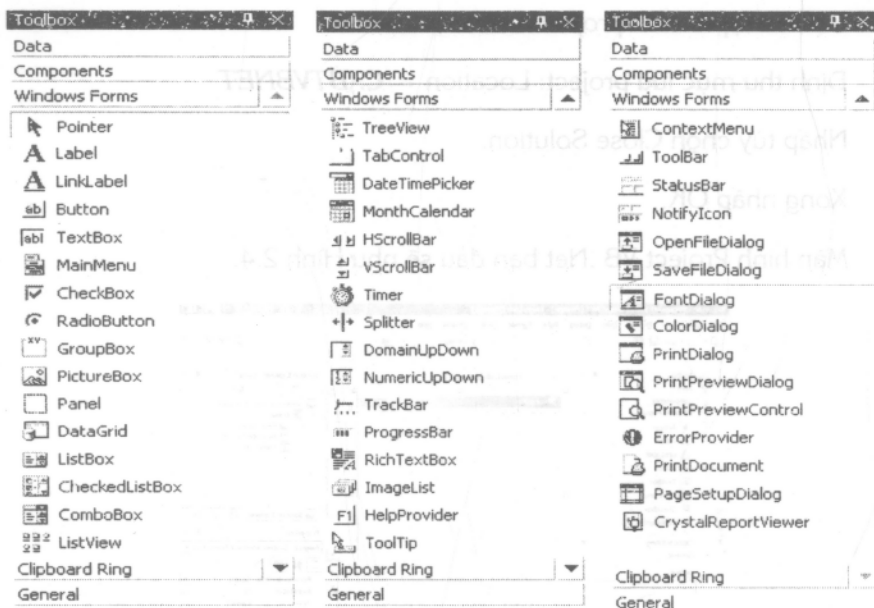
Hình 2.4 Màn hình ban đầu của một project Visual Basic .Net.

Có thể thấy project mới được tạo ban đầu chỉ chứa một form với tên là Form1.

GHI CHÚ

- ◆ Mặc định các project được tạo sẽ lưu trong thư mục H:\Documents and Settings\Administrator\My Documents\Visual Studio Projects, với H là tên ổ đĩa hệ thống (ổ đĩa cài đặt Windows 2000 Server).

Bước 3. Chọn lệnh View ➡ Toolbox để mở thanh công cụ nếu chưa được mở (Hình 2.5).



Hình 2.5 Toolbox – nơi chứa các điều khiển căn bản trong Visual Basic .Net.

Bước 4. Chọn form Form1. Xong chọn lệnh View ➡ Properties Windows hay nhấn F4 để bật cửa sổ Properties nếu chưa bật (Hình 2.6).

Định lại các thuộc tính của form như sau:

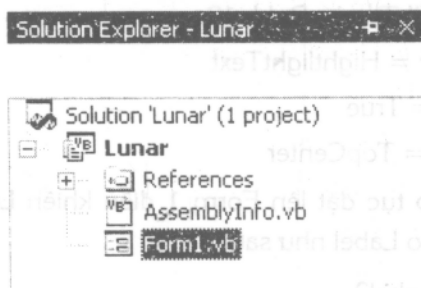
- Text: Calendar -> Lunar
- BackColor = ControlLight
- Name = FrmLunar
- StartPosition = CenterScreen
- Size = 384, 224
- Icon = C:\Program Files\Microsoft Visual Studio .Net\Common7\Graphics\Icons\Computer\PC01.ico

(để gắn biểu tượng cái máy tính trên dòng tiêu đề đầu form, chọn thuộc tính icon, nhấp nút ba chấm để chọn tập tin icon chứa biểu tượng máy tính).



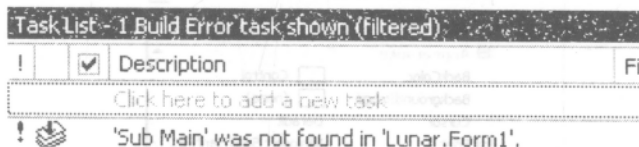
Hình 2.6 Cửa sổ Properties dùng để định thuộc tính đối tượng.

Bước 5. Chọn lệnh View ➡ Solution Explorer mở cửa sổ Solution Explorer (Hình 2.7). Chọn Form1, và định lại thuộc tính Filename = Lunar.vb trong cửa sổ Properties.



Hình 2.7 Cửa sổ Solution Explorer.

Bước 6. Mặc định Project VB .Net sẽ gọi Form1 khi thi hành. Nhưng do khi thiết kế, bạn đã đổi tên Form1 thành FrmLunar, nên nếu bạn thực hiện lệnh Build ➡ Build Solution, bạn sẽ thấy thông báo sau trong cửa sổ Task List (Hình 2.8).



Hình 2.8 Lỗi xảy ra khi biên dịch Project do đổi tên Form1.

Nhấp đúp lên thông báo lỗi “Sub Main was not found in Lunar.Form1” để gọi hộp thoại Startup Object, chọn form khởi động mới là Lunar.frmLunar, xong nhấp OK.

Giờ đây, bạn có thể nhấn F5 để biên dịch lại và chạy thử chương trình. Một form trống sẽ được hiển thị. Nhấp nút Close ở góc phải form để trở lại màn hình thiết kế.

Bước 7. Đặt lên form 1 điều khiển Label, bằng cách nhấp đúp biểu tượng Label trên Toolbox. Điều khiển có tên mặc định là Name = Label1.

Định lại các thuộc tính cho Label như sau:

- Font = VNI-Times, Bold, 12
- BackColor = HighlightText
- AutoSize = True
- TextAlign = TopCenter

Bước 8. Tiếp tục đặt lên Form 1 điều khiển Label nữa và định lại các thuộc tính cho Label như sau:

- Name = Label2
- Font = VNI-Times, Regular, 10

- BackColor = InactiveBorder
- AutoSize = False
- Size = 112,22
- TextAlign = BottomLeft
- Text = Năm dương lịch
- ImageAlign = BottomRight
- Image = C:\Program Files\Microsoft Visual Studio .Net\Common7\Graphics\Icons\Elements\Sun.ico

Bước 9. Sao chép Label2 thành 1 Label nữa và định lại các thuộc tính sau:

- Name = Label3
- Text = Năm âm lịch
- Image = C:\Program Files\Microsoft Visual Studio .Net\Common7\Graphics\Icons\Elements\Moon08.ico

Bước 10. Đặt lên form 1 Text box và định lại thuộc tính như sau:

- Name = txtYear
- Text = 2003
- TextAlign = Center
- Font = VNI-Times, Bold, 10

Bước 11. Đặt lên form thêm 1 Label nữa và định lại thuộc tính như sau:

- Name = lblLunar
- Text = (rỗng)
- TextAlign = TopCenter
- Font = VNI-Times, Bold, 10

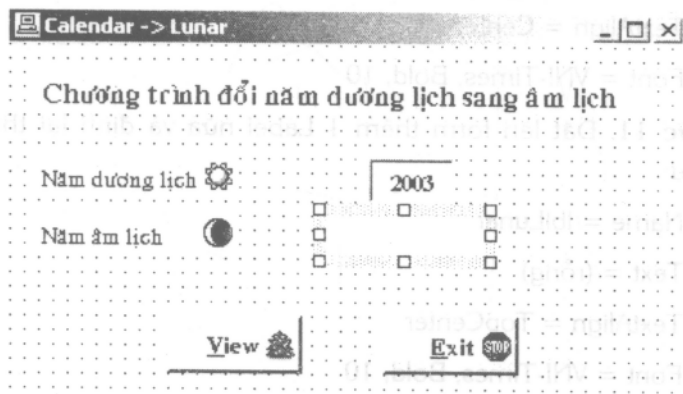
Bước 12. Đặt lên form 1 Button bằng cách nhấp đúp biểu tượng Button trên Toolbox và định lại thuộc tính như sau:

- Name = bView
- Text = &View
- Font = VNI-Times, Bold, 10
- TextAlign = MiddleCenter
- BackColor = ControlLight
- Size = 75, 32
- ImageAlign = MiddleRight
- Image = C:\Program Files\Microsoft Visual Studio .Net\Common7\Graphics\Icons\Misc\Misc44.ico

Bước 13. Sao chép bView thành một Button nữa và định lại các thuộc tính sau:

- Name = vExit
- Text = &Exit
- Image = C:\Program Files\Microsoft Visual Studio .Net\Common7\Graphics\Icons\Traffic\Trffc14.ico

Kết quả giao diện thiết kế sẽ như Hình 2.9.



Hình 2.9 Màn hình thiết kế form FrmLunar.

Bước 14. Viết mã lệnh chương trình.

Public Class FrmLunar

Inherits System.Windows.Forms.Form

Dim can(10) As String

Dim chi(12) As String

' Mã lệnh phần này do Windows Form Designer tự phát sinh trong quá trình thiết kế

#Region " Windows Form Designer generated code "

Public Sub New()

MyBase.New()

InitializeComponent()

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

Private components As System.ComponentModel.IContainer

' Chú ý: Các thủ tục sau tự động phát sinh và sửa đổi trong quá trình bạn

' thiết kế Form. Không nên điều chỉnh bằng tay đoạn mã lệnh này

Friend WithEvents FontDialog1 As System.Windows.Forms.FontDialog

Friend WithEvents Label1 As System.Windows.Forms.Label

Friend WithEvents Label2 As System.Windows.Forms.Label

Friend WithEvents Label3 As System.Windows.Forms.Label

```

Friend WithEvents BVIEW As System.Windows.Forms.Button
Friend WithEvents BeXIT As System.Windows.Forms.Button
Friend WithEvents txtYear As System.Windows.Forms.TextBox
Friend WithEvents lblLunar As System.Windows.Forms.Label
<System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
    Dim resources As System.Resources.ResourceManager = New
System.Resources.ResourceManager(GetType(FrmLunar))

    Me.FontDialog1 = New System.Windows.Forms.FontDialog()
    Me.Label1 = New System.Windows.Forms.Label()
    Me.Label2 = New System.Windows.Forms.Label()
    Me.Label3 = New System.Windows.Forms.Label()
    Me.BVIEW = New System.Windows.Forms.Button()
    Me.BeXIT = New System.Windows.Forms.Button()
    Me.txtYear = New System.Windows.Forms.TextBox()
    Me.lblLunar = New System.Windows.Forms.Label()
    Me.SuspendLayout()

    ' Label1

    Me.Label1.AutoSize = True

    Me.Label1.BackColor = System.Drawing.SystemColors.HighlightText

    Me.Label1.Font = New System.Drawing.Font("VNI-Times", 12.0!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))

    Me.Label1.Location = New System.Drawing.Point(16, 16)
    Me.Label1.Name = "Label1"

    Me.Label1.Size = New System.Drawing.Size(338, 22)
    Me.Label1.TabIndex = 0

    Me.Label1.Text = "Chương trình đổi năm dương lịch sang âm lịch"

```

```

Me.Label1.TextAlign = System.Drawing.ContentAlignment.TopCenter
' Label2
Me.Label2.BackColor = System.Drawing.SystemColors.InactiveBorder
Me.Label2.Font = New System.Drawing.Font("VNI-Times", 9.749999!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
Me.Label2.Image = CType(resources.GetObject("Label2.Image"),
System.Drawing.Bitmap)
Me.Label2.ImageAlign =
System.Drawing.ContentAlignment.BottomRight
Me.Label2.Location = New System.Drawing.Point(16, 64)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(112, 22)
Me.Label2.TabIndex = 1
Me.Label2.Text = "Năm dương lịch"
Me.Label2.TextAlign = System.Drawing.ContentAlignment.BottomLeft
' Label3
Me.Label3.BackColor = System.Drawing.SystemColors.InactiveBorder
Me.Label3.Font = New System.Drawing.Font("VNI-Times", 9.749999!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
Me.Label3.Image = CType(resources.GetObject("Label3.Image"),
System.Drawing.Bitmap)
Me.Label3.ImageAlign =
System.Drawing.ContentAlignment.BottomRight
Me.Label3.Location = New System.Drawing.Point(16, 96)
Me.Label3.Name = "Label3"

```

```
Me.Label3.Size = New System.Drawing.Size(112, 22)
Me.Label3.TabIndex = 2
Me.Label3.Text = "Năm âm lịch"
Me.Label3.TextAlign = System.Drawing.ContentAlignment.BottomLeft
' BVIEW
Me.BVIEW.Font = New System.Drawing.Font("VNI-Times", 9.749999!,
    System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
    CType(0, Byte))
Me.BVIEW.Image = CType(resources.GetObject("BVIEW.Image"),
    System.Drawing.Bitmap)
Me.BVIEW.ImageAlign = System.Drawing.ContentAlignment.MiddleRight
Me.BVIEW.Location = New System.Drawing.Point(88, 152)
Me.BVIEW.Name = "BVIEW"
Me.BVIEW.Size = New System.Drawing.Size(75, 32)
Me.BVIEW.TabIndex = 3
Me.BVIEW.Text = "&View"
' Bexit
Me.BeXIT.Font = New System.Drawing.Font("VNI-Times", 9.749999!,
    System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
    CType(0, Byte))
Me.BeXIT.Image = CType(resources.GetObject("BeXIT.Image"),
    System.Drawing.Bitmap)
Me.BeXIT.ImageAlign = System.Drawing.ContentAlignment.MiddleRight
Me.BeXIT.Location = New System.Drawing.Point(208, 152)
Me.BeXIT.Name = "BeXIT"
Me.BeXIT.Size = New System.Drawing.Size(75, 32)
Me.BeXIT.TabIndex = 4
```

```
Me.BeXIT.Text = "&Exit"
```

```
' txtYear
```

```
Me.txtYear.Font = New System.Drawing.Font("VNI-Times", 9.749999!,  
    System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,  
    CType(0, Byte))
```

```
Me.txtYear.Location = New System.Drawing.Point(200, 64)
```

```
Me.txtYear.Name = "txtYear"
```

```
Me.txtYear.Size = New System.Drawing.Size(48, 25)
```

```
Me.txtYear.TabIndex = 5
```

```
Me.txtYear.Text = "2003"
```

```
Me.txtYear.TextAlign = System.Windows.Forms.HorizontalAlignment.Center
```

```
' lblLunar
```

```
Me.lblLunar.BackColor = System.Drawing.SystemColors.InactiveBorder
```

```
Me.lblLunar.Font = New System.Drawing.Font("VNI-Times", 9.749999!,  
    System.Drawing.FontStyle.Regular,  
    System.Drawing.GraphicsUnit.Point, CType(0, Byte))
```

```
Me.lblLunar.ImageAlign = System.Drawing.ContentAlignment.TopCenter
```

```
Me.lblLunar.Location = New System.Drawing.Point(176, 96)
```

```
Me.lblLunar.Name = "lblLunar"
```

```
Me.lblLunar.Size = New System.Drawing.Size(88, 22)
```

```
Me.lblLunar.TabIndex = 6
```

```
Me.lblLunar.TextAlign = System.Drawing.ContentAlignment.BottomCenter
```

```
' FrmLunar
```

```
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
```

```
Me.BackColor = System.Drawing.SystemColors.ControlLight
```

```
Me.ClientSize = New System.Drawing.Size(376, 197)
```

```
Me.Controls.AddRange(New System.Windows.Forms.Control()  
    {Me.IblLunar, Me.txtYear, Me.BeXIT, Me.BVIEW, Me.Label3,  
    Me.Label2, Me.Label1})  
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)  
Me.Name = "FrmLunar"  
Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen  
Me.Text = "Calendar -> Lunar"  
Me.ResumeLayout(False)
```

End Sub

#End Region

' Thủ tục được gọi khi form vừa nạp

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

```
    chi(1) = "Thân"  
    chi(2) = "Dậu"  
    chi(3) = "Tuất"  
    chi(4) = "Hợi"  
    chi(5) = "Tý"  
    chi(6) = "Sửu"  
    chi(7) = "Dần"  
    chi(8) = "Mẹo"  
    chi(9) = "Thìn"  
    chi(10) = "Tỵ"  
    chi(11) = "Ngọ"  
    chi(12) = "Mùi"  
    can(1) = "Canh"
```

```

can(2) = "Tân"
can(3) = "Nhâm"
can(4) = "Quý"
can(5) = "Giáp"
can(6) = "Ất"
can(7) = "Bính"
can(8) = "Đinh"
can(9) = "Mậu"
can(10) = "Kỷ"

```

End Sub

' Thủ tục được gọi khi người dùng nhấp nút Exit

Private Sub BeXIT_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BeXIT.Click

End

End Sub

' Thủ tục được gọi khi người dùng nhấp nút View

Private Sub BVIEW_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BVIEW.Click

Dim v_chi, v_can As Integer

Dim v_result As String

v_chi = Val(txtYear.Text) Mod 12

v_can = Val(txtYear.Text) Mod 10

v_result = can(v_can + 1) & " " & chi(v_chi + 1)

lblLunar.Text = v_result

txtYear.Focus()

End Sub

End Class

LƯU Ý

- ♦ Các lệnh định nghĩa Class do VB .Net tự động thêm vào trong quá trình thiết kế Form. Bạn chỉ cần nhập vào các dòng lệnh in nghiêng mà thôi.

Bước 15. Biên dịch và chạy thử chương trình.

Chọn lệnh File ➡ Save All để lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Xong nhấn F5 hoặc chọn lệnh Debug ➡ Start để chạy thử chương trình.

Nhập năm 1971, xong nhấn View. Kết quả sẽ như Hình 2.10.

 Calendar -> Lunar _ □ ×

Chương trình đổi năm dương lịch sang âm lịch

Năm dương lịch 

1971

Năm âm lịch



Tân Hợi

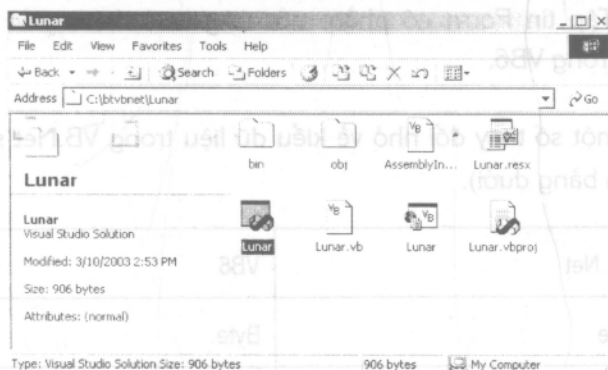
View 

Exit 

Hình 2.10 Kết quả form FrmLunar.

MÁCH NƯỚC

- ♦ Mỗi Project được tạo trong VB.Net sẽ tự động được lưu vào cùng một Folder có cùng tên với Project (Hình 2.11).



Hình 2.11 Các tập tin được tạo ứng với project Lunar.

Kỹ thuật lập trình

Nếu bạn đã từng làm quen với Visual Basic 6, có thể thấy tuy có vài điểm khác biệt, nhưng nhìn chung cách thiết kế ứng dụng Lunar trong Visual Basic .Net là tương tự như Visual Basic 6 về phương diện định nghĩa biến, sử dụng thuộc tính và xây dựng các thủ tục biến cố cho các điều khiển đã sử dụng.

Sau đây là một số điểm khác biệt đầu tiên bạn cần lưu ý khi làm việc với VB .Net:

- VB .Net là ngôn ngữ lập trình hướng đối tượng, nên tất cả các điều khiển sử dụng và mã lệnh xây dựng đều được định nghĩa trong các lớp. Đây là một thế mạnh của VB .Net mà bạn sẽ có dịp tìm hiểu kỹ hơn trong phần sau.
- Phần mở rộng của các tập tin trong VB.Net:
 - Tập tin Project có phần mở rộng là *.vbproj thay vì *.vb như trong VB6.

- Tập tin Form có phần mở rộng là *.vb thay vì *.frm như trong VB6.
- Có một số thay đổi nhỏ về kiểu dữ liệu trong VB.Net so với VB6 (xem bảng dưới).

VB .Net	VB6
Byte	Byte
Boolean	Boolean
Decimal	Không có
Không có	Currency
Short	Integer
Integer	Long
Long (<i>số nguyên 8 bytes</i>)	Không có
Single	Single
Double	Double
String	String
Date	Date
Không có	Variant
Object	Không có

- Có thể sử dụng điều khiển Label để hiển thị văn bản lẫn hình ảnh cùng lúc. Sau đây là một số thuộc tính chính của điều khiển Label trong VB.Net:
 - Name: Tên điều khiển.
 - BackColor: Định màu nền label.
 - ForeColor: Định màu chữ trong label.
 - Font: Định phong chữ, kiểu chữ, kích cỡ chữ hiển thị trên Label.
 - Text: nội dung văn bản hiển thị trên Label (*thay cho thuộc tính Caption trong VB6*).
 - TextAlign: canh trái, giữa hay phải nội dung trong Label (*thay cho thuộc tính Alignment trong VB6*). Có thể chọn một trong các giá trị sau: TopLeft, TopCenter, TopRight, MiddleLeft, MiddleCenter, MiddleRight, BottomLeft, BottomCenter, BottomRight.
 - Image: tên tập tin hình icon sẽ hiển thị trong Label (*đây là thuộc tính mới so với VB6*).
 - ImageAlign: canh trái, giữa hay phải hình ảnh trong Label (*đây là thuộc tính mới so với VB6*).
 - ImageList: xác định tên điều khiển ImageList cung cấp hình cho Label (*none nếu không sử dụng*).
 - ImageIndex: số thứ tự của hình sẽ hiển thị trên Label trong nhóm hình chứa trong điều khiển ImageList.
 - Size: kích cỡ label (*thay cho 2 thuộc tính Width và Height trong VB6*).

- Locked: Khóa hay không khóa điều khiển Label (trong VB6 điều khiển Label không có thuộc tính Locked do không nhận Focus).
 - TabIndex: thứ tự nhận focus của Label trong form.
- Có thể thay đổi biểu tượng icon trong dòng tiêu đề form. Sau đây là một số thuộc tính chính của Form trong VB.Net.
 - Name: Tên form.
 - Icon: Tên tập tin icon chứa biểu tượng sẽ hiển thị ở góc trái thanh tiêu đề form.
 - StartPosition: xác định vị trí ban đầu của form trên màn hình khi form vừa nạp (Manual, CenterScreen, WindowsDefaultLocation, WindowsDefaultBounds, CenterParent).
 - WindowState: Xác định trạng thái ban đầu của form (Normal, Minimized, Maximized).
 - Text: Văn bản hiển thị trên thanh tiêu đề form (tương ứng với thuộc tính Caption trong VB6).
 - BackgroundImage: xác định tập tin hình làm nền form (tương ứng với thuộc tính Picture trong VB6).
 - BackColor: Màu nền form.
 - FormBorderStyle: Kiểu đường viền form.
 - Opacity: Cho phép thay đổi độ sáng tối của form, có thể làm mờ form khi hiển thị. Giá trị mặc định là 100%. Đây là thuộc tính mới so với VB6.

- Có thể sử dụng điều khiển Button để hiển thị văn bản lẫn hình ảnh cùng lúc. Sau đây là một số thuộc tính chính của điều khiển Button:
 - Name: Tên điều khiển.
 - BackColor: Định màu nền button.
 - ForeColor: Định màu chữ trong button.
 - Font: Định phông chữ, kiểu chữ, kích cỡ chữ hiển thị trên button.
 - Text: nội dung văn bản hiển thị trên button (*tương ứng với thuộc tính Caption trong VB6*).
 - TextAlign: Canh trái, giữa hay phải nội dung văn bản trong button (so với VB6, bạn không nhúng canh văn bản trong button theo chiều ngang, mà còn có thể canh theo chiều dọc).
 - Image: Tên tập tin hình icon sẽ hiển thị trong button.
 - ImageAlign: Canh trái, giữa hay phải hình ảnh trong button.
 - ImageList: Xác định tên điều khiển ImageList cung cấp hình cho button.
 - ImageIndex: Số thứ tự của hình sẽ hiển thị trên button trong nhóm hình chứa trong điều khiển ImageList
 - Size: Kích cỡ button (thay cho hai thuộc tính Width và Height trong VB6).
 - Locked: Khóa hay không khóa điều khiển Button.
 - TabIndex: Thứ tự nhận focus của điều khiển Button trong form.

- Có thể sử dụng điều khiển Textbox để hiển thị văn bản trong form. Sau đây là một số thuộc tính chính của điều khiển:
 - Name: Tên điều khiển Textbox.
 - BackColor: Định màu nền Textbox.
 - ForeColor: Định màu chữ trong Textbox.
 - Font: Định phong chữ, kiểu chữ, kích cỡ chữ hiển thị trên Textbox.
 - Text: Nội dung văn bản hiển thị trên Textbox.
 - Wordwrap: Giá trị True cho phép văn bản dài tự cuộn xuống dòng.
 - Readonly: Giá trị True, chỉ có thể xem văn bản trong Textbox, chứ không được điều chỉnh.
 - PasswordChar: Ký tự thay thế cho các ký tự nhập vào Textbox. Thường dùng khi người dùng nhập mật mã login chương trình.
 - CharacterCasing: Giá trị Normal (mặc định), Upper (tự động đổi văn bản đã nhập thành chữ hoa), Lower (tự động đổi văn bản đã nhập thành chữ thường).
 - Cursor: Dạng con trỏ sẽ hiển thị lúc nhập liệu vào Textbox.
 - Multiline: Giá trị True cho phép nhập, hiển thị văn bản trong Textbox trên nhiều dòng.
 - Scrollbars: Cho phép sử dụng hay không sử dụng các thanh cuộn ngang và dọc (None: mặc định, Horizontal, Vertical, Both)

- Lines: Định nghĩa mảng chuỗi hiển thị trên Textbox. Mỗi thành phần mảng hiển thị trên một dòng.
- Autosize: Giá trị true cho phép Textbox tự định lại kích thước để hiển thị toàn bộ nội dung văn bản.
- BorderStyle: Xác định kiểu đường viền bao quanh Textbox.
- TabStop: Giá trị True cho phép Textbox có thể nhận focus từ điều khiển trước với phím TAB.
- Locked: Giá trị True sẽ khóa text box, không cho sửa đổi nội dung Textbox.

Bài 3.

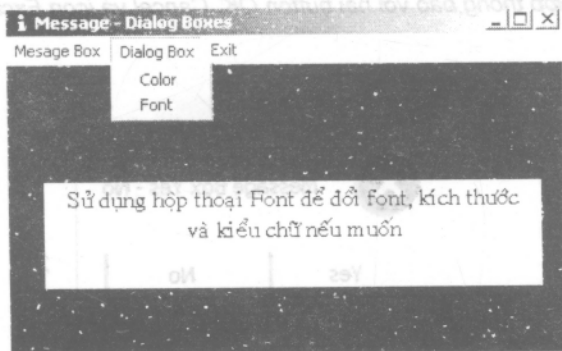
Chương trình Menu

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với việc thiết kế menu trong một project với VB .Net, ngoài ra bạn có thể tìm hiểu xem có sự khác biệt gì giữa VB.Net với VB6 khi thực hiện các lệnh định nghĩa hộp thông báo và hộp thoại.

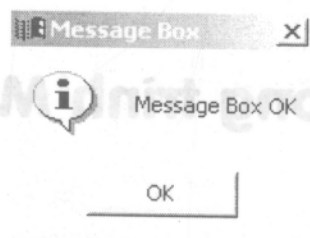
Hoạt động chương trình

Mỗi khi thi hành, chương trình sẽ hiển thị giao diện như Hình 3.1 và chờ người dùng chọn một trong các lệnh trên menu.

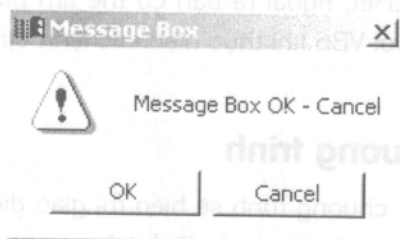


Hình 3.1 Giao diện chương trình khi bắt đầu form.

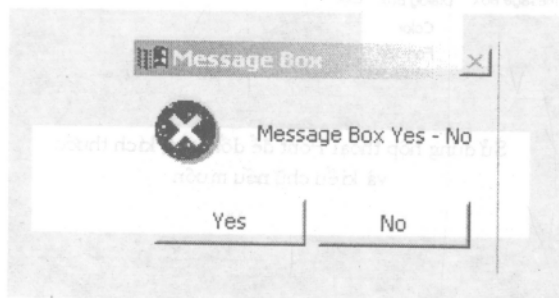
Các hộp thông báo tương ứng sẽ xuất hiện khi người dùng chọn một trong các lệnh thuộc mục “Message Box” (Hình 3.2 đến Hình 3.7).



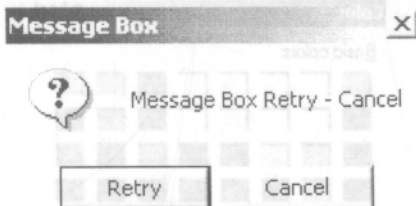
Hình 3.2 Hộp thông báo với một button OK và icon Information.



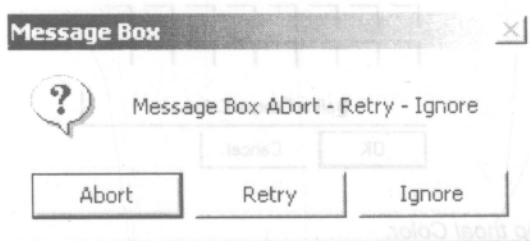
Hình 3.3 Hộp thông báo với hai button OK, Cancel và icon Exclamation.



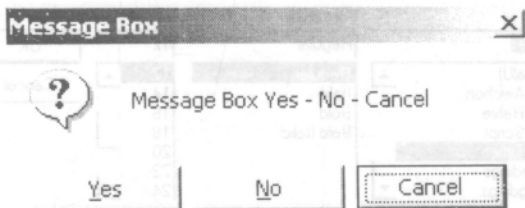
Hình 3.4 Hộp thông báo với hai button Yes, No và icon Critical.



Hình 3.5 Hộp thông báo với hai button *Retry*, *Cancel* và icon *Question*.

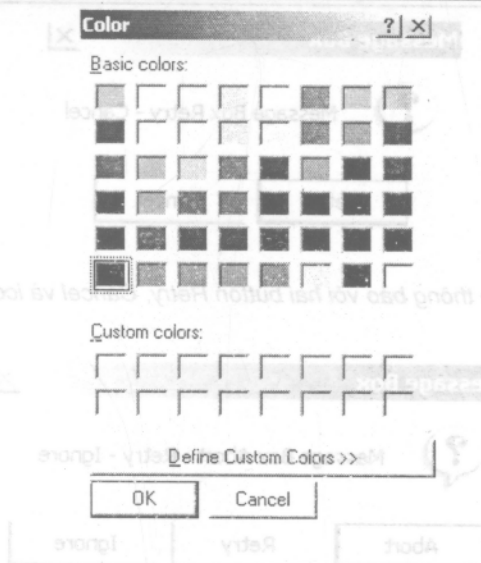


Hình 3.6 Hộp thông báo với ba button *Abort*, *Retry*, *Ignore* và icon *Question*.

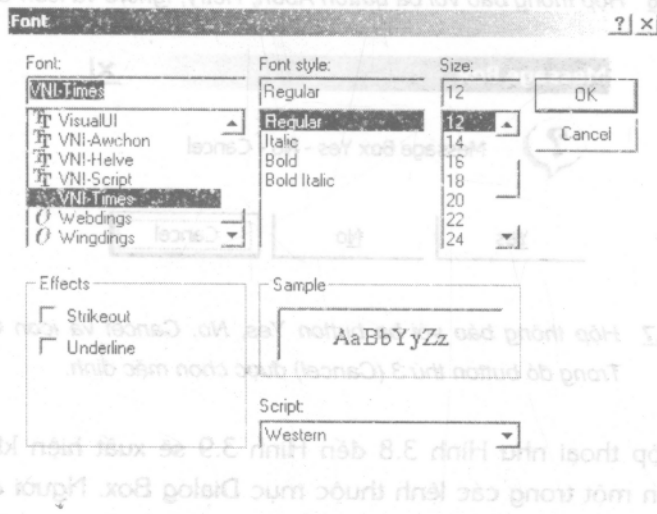


Hình 3.7 Hộp thông báo với ba button *Yes*, *No*, *Cancel* và icon *Question*. Trong đó button thứ 3 (*Cancel*) được chọn mặc định.

Các hộp thoại như Hình 3.8 đến Hình 3.9 sẽ xuất hiện khi người dùng chọn một trong các lệnh thuộc mục *Dialog Box*. Người dùng có thể dùng hộp thoại *Color* để đổi màu nền label hiển thị văn bản, dùng hộp thoại *Font* để đổi font chữ, kích cỡ, kiểu chữ cho văn bản hiển thị trong *Label*.

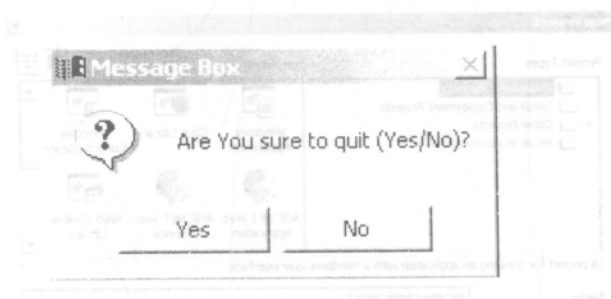


Hình 3.8 Hộp thoại Color.



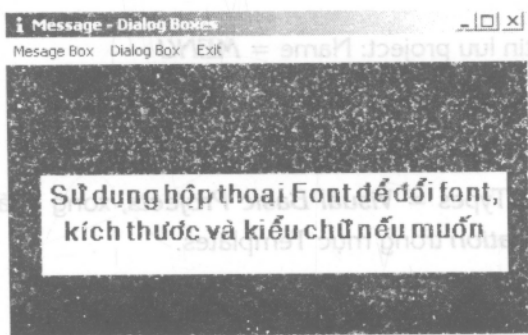
Hình 3.9 Hộp thoại Font.

Khi nhấp mục Exit, sẽ xuất hiện hộp thoại hỏi ý kiến người dùng như Hình 3.10. Chọn Yes để kết thúc chương trình, chọn No để trở lại menu chính.



Hình 3.10 Hộp thoại hỏi ý người dùng có muốn thoát chương trình không.

Hình 3.11 là một ví dụ minh họa trong trường hợp người dùng chọn màu xanh Green trong hộp thoại Color và chọn font Vni-Helve, size 14, style Bold trong hộp thoại Font.



Hình 3.11 Kết quả thay đổi màu nền và font chữ trong Label1.

Hướng dẫn thiết kế

Bước 1. Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net.

Bước 2. Nhấp nút New Project hay chọn lệnh File ➡ New Project để gọi cửa sổ New Project (Hình 3.12).



Hình 3.12 Cửa sổ New Project.

Đặt tên tập tin lưu project: Name = **MENU**

Định thư mục lưu project: Location = **C:\BTVBNET**

Nhấp tùy chọn **Close Solution**.

Chọn Project Types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.

Xong nhấp OK.

Bước 3. Chọn lệnh View ➡ Toolbox để mở thanh công cụ nếu chưa được mở.

Bước 4. Chọn lệnh View ➡ Solution Explorer để mở cửa sổ Solution Explorer nếu chưa mở. Xong chọn form **Form1.vb** và nhấn F4 hay chọn lệnh View ➡ Properties Windows để mở cửa sổ Properties nếu chưa mở.

Định lại thuộc tính Filename = **Menu.vb** trong cửa sổ Properties.

Bước 5. Nhấp màn hình Form trên vùng làm việc và định lại các thuộc tính của Form như sau:

- Text: Message – Dialog Boxes
- BackgroundImage = C:\Windows\Blue Lace 16.bmp
- Name = FrmMenu
- StartPosition = CenterScreen
- Size = 384, 224
- Icon = C:\Program Files\Microsoft Visual Studio .Net\Common7\Graphics\Icons\Computer\Msgbox04.ico

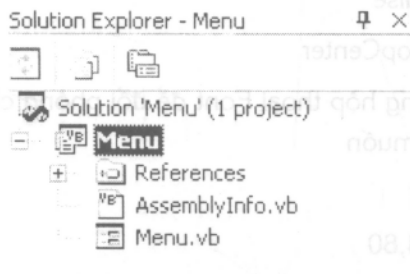
MÁCH NƯỚC

- ◆ Tập tin ảnh Blue Lace 16.bmp chỉ tồn tại trong Windows 2000, bạn có thể thay thế bằng một tập tin ảnh khác tùy thích.

Bước 6. Mặc định Project VB .Net sẽ gọi Form1 khi thi hành. Nhưng do khi thiết kế, bạn đã đổi tên Form1 thành **FrmMenu**, nên bạn cần định lại tên form sẽ được nạp khi thi hành Project.

Nhấp nút phải mouse lên project Menu trong cửa sổ Solution Explorer (Hình 3.13), xong chọn lệnh Properties trong menu nhanh để gọi hộp thoại *Menu Property Pages*. Định lại form khởi động:

- Startup Object = **frmMenu**, xong nhấp Apply (Hình 3.14).
- Nhấp OK để trở lại màn hình thiết kế Form.



Hình 3.13 Cửa sổ Solution Explorer.



Hình 3.14 Hộp thoại Menu Property Pages.

Giờ đây, bạn có thể nhấn F5 để biên dịch và chạy thử chương trình. Một form trống sẽ được hiển thị. Nhấp nút Close ở góc phải form để trở lại màn hình thiết kế.

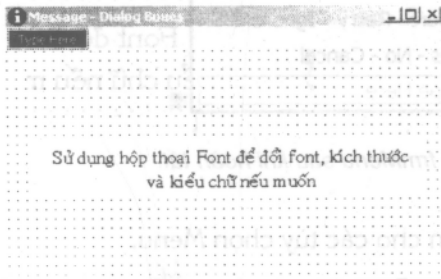
Bước 7. Đặt lên form 1 điều khiển Label, bằng cách nhấp đúp biểu tượng Label trên Toolbox. Điều khiển có tên mặc định là Name = Label1.

Định lại các thuộc tính cho Label như sau:

- Name = Label1
- Font = VNI-Times, Regular, 12
- BackColor = HighlightText
- AutoSize = False
- TextAlign = TopCenter
- Text = Sử dụng hộp thoại Font để đổi phông chữ, kích thước và kiểu chữ nếu muốn
- Size = 336,72
- Location = 24,80

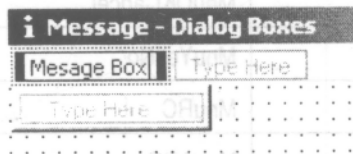
Bước 8. Thiết kế Menu cho form frmMenu.

Nhấp biểu tượng MainMenu trên Toolbox. Xong nhấp vào vùng làm việc form frmMenu (Hình 3.15).



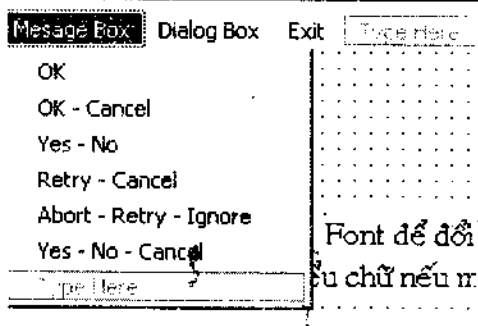
Hình 3.15 Màn hình form frmMenu lúc bắt đầu thiết kế Menu.

Nhấp vào mục Type Here và nhập Message Box. VB.Net tự động thêm các mục Type Here bên phải và dưới mục menu bạn vừa nhập để chờ bạn định nghĩa các lệnh trên menu ngang và các tùy chọn trong menu xổ xuống ứng với từng lệnh (Hình 3.16).



Hình 3.16 Định nghĩa tiếp các lệnh và tùy chọn menu.

Tiếp tục thực hiện việc định nghĩa menu để được kết quả như Hình 3.17.



Hình 3.17 Menu form frmMenu sau khi hoàn tất.

Bước 9. Định lại tên cho các tùy chọn Menu.

Chọn mục Message Box, trong cửa sổ Properties, đổi thuộc tính **Name** từ MenuItem1 thành mnuMessage. Tương tự, bạn định lại tên cho các mục chọn trên menu như sau:

Tên mục chọn menu	Thuộc tính Name
Message Box	MnuMessage
OK	MnuOK
OK - Cancel	MnuOKCancel
Yes - No	MnuYesNo
Retry - Cancel	MnuRC
Abort - Retry - Cancel	MnuARC
Yes - No - Cancel	MnuYNC
Dialog Box	MnuDialog
Color	MnuColor
Font	MnuFont
Exit	MnuExit



Hình 3.18 Màn hình thiết kế form frmMenu.

Định lại thuộc tính Menu cho form frmMenu trong cửa sổ Properties như sau:

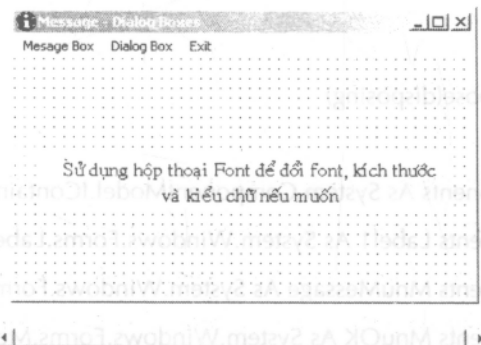
- Menu = MainMenu1

MÁCH NƯỚC

- ◆ Mỗi lần bạn nhấp biểu tượng MainMenu trên Toolbox, một đối tượng Menu mới sẽ xuất hiện dưới cửa sổ thiết kế form.

Bước 10. Bổ sung hai điều khiển FontDialog và ColorDialog từ Toolbox vào form frmMenu bằng cách lần lượt nhấp đúp trên biểu tượng của hai điều khiển đó. Khi đó, vùng bên dưới màn hình thiết kế form sẽ xuất hiện thêm hai đối tượng với tên FontDialog1 và ColorDialog1.

Màn hình thiết kế của bạn bây giờ sẽ như Hình 3.19.



Hình 3.19 Kết quả thiết kế form frmMenu.

Bước 11. Viết mã lệnh chương trình.

Để viết lệnh xử lý cho từng mục chọn menu, nhấp đúp mục chọn menu tương ứng. Sau đây mã lệnh của toàn bộ lớp **frmMenu** (mã lệnh trong các thủ tục in chữ đậm là phần bạn cần nhập vào).

Public Class frmMenu

*' Lớp **frmMenu** kế thừa lớp **Form***

Inherits System.Windows.Forms.Form

' Vùng mã lệnh tự phát sinh trong quá trình bạn thiết kế form

#Region " Windows Form Designer generated code "

Public Sub New()

MyBase.New()

InitializeComponent()

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

Private components As System.ComponentModel.IContainer

Friend WithEvents Label1 As System.Windows.Forms.Label

Friend WithEvents MnuMessage As System.Windows.Forms.MenuItem

Friend WithEvents MnuOK As System.Windows.Forms.MenuItem

Friend WithEvents MnuOKCancel As System.Windows.Forms.MenuItem

Friend WithEvents MnuYesNo As System.Windows.Forms.MenuItem

```

Friend WithEvents MnuRC As System.Windows.Forms.MenuItem
Friend WithEvents MnuARI As System.Windows.Forms.MenuItem
Friend WithEvents MnuYNC As System.Windows.Forms.MenuItem
Friend WithEvents mnuDialog As System.Windows.Forms.MenuItem
Friend WithEvents MnuColor As System.Windows.Forms.MenuItem
Friend WithEvents MnuFont As System.Windows.Forms.MenuItem
Friend WithEvents MnuExit As System.Windows.Forms.MenuItem
Friend WithEvents MainMenu1 As System.Windows.Forms.MainMenu
Friend WithEvents FontDialog1 As System.Windows.Forms.FontDialog
Friend WithEvents ColorDialog1 As System.Windows.Forms.ColorDialog
<System.Diagnostics.DebuggerStepThrough()> Private Sub
    InitializeComponent()
Dim resources As System.Resources.ResourceManager = New
    System.Resources.ResourceManager(GetType(frmMenu))
Me.Label1 = New System.Windows.Forms.Label()
Me.MnuMessage = New System.Windows.Forms.MenuItem()
Me.MnuOK = New System.Windows.Forms.MenuItem()
Me.MnuOKCancel = New System.Windows.Forms.MenuItem()
Me.MnuYesNo = New System.Windows.Forms.MenuItem()
Me.MnuRC = New System.Windows.Forms.MenuItem()
Me.MnuARI = New System.Windows.Forms.MenuItem()
Me.MnuYNC = New System.Windows.Forms.MenuItem()
Me.mnuDialog = New System.Windows.Forms.MenuItem()
Me.MnuColor = New System.Windows.Forms.MenuItem()
Me.MnuFont = New System.Windows.Forms.MenuItem()
Me.MnuExit = New System.Windows.Forms.MenuItem()

```

```
Me.MainMenu1 = New System.Windows.Forms.MainMenu()
Me.FontDialog1 = New System.Windows.Forms.FontDialog()
Me.ColorDialog1 = New System.Windows.Forms.ColorDialog()
Me.SuspendLayout()
' Đoạn mã lệnh định nghĩa đối tượng Label 1
Me.Label1.BackColor = System.Drawing.SystemColors.HighlightText
Me.Label1.Font = New System.Drawing.Font("VNI-Times", 12.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
CType(0, Byte))
Me.Label1.Location = New System.Drawing.Point(24, 80)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(336, 72)
Me.Label1.TabIndex = 0
Me.Label1.Text = "Sử dụng hộp thoại Font để đổi font, kích thước và
kiểu chữ nếu muốn"
Me.Label1.TextAlign = System.Drawing.ContentAlignment.TopCenter
' Đoạn mã lệnh định nghĩa mục Message trong menu ngang
Me.MnuMessage.Index = 0
Me.MnuMessage.MenuItems.AddRange(New
System.Windows.Forms.MenuItem() {Me.MnuOK, Me.MnuOKCancel,
Me.MnuYesNo, Me.MnuRC, Me.MnuARI, Me.MnuYNC})
Me.MnuMessage.Text = "Mesage Box"
' Mục tùy chọn OK (MnuOK)
Me.MnuOK.Index = 0
Me.MnuOK.Text = "OK "
' Mục tùy chọn OK - Cancel (MnuOKCancel)
```

```

Me.MnuOKCancel.Index = 1
Me.MnuOKCancel.Text = "OK - Cancel"
' Mục tùy chọn Yes - No (MnuYesNo)
Me.MnuYesNo.Index = 2
Me.MnuYesNo.Text = "Yes - No"
' Mục tùy chọn Retry - Cancel (MnuRC)
Me.MnuRC.Index = 3
Me.MnuRC.Text = "Retry - Cancel"
' Mục tùy chọn Abort - Retry - Ignore (MnuARI)
Me.MnuARI.Index = 4
Me.MnuARI.Text = "Abort - Retry - Ignore"
' Mục tùy chọn Yes - No - Cancel (MnuYNC)
Me.MnuYNC.Index = 5
Me.MnuYNC.Text = "Yes - No - Cancel"
' Mục Dialog trên menu ngang (mnuDialog)
Me.mnuDialog.Index = 1
Me.mnuDialog.MenuItems.AddRange(New
    System.Windows.Forms.MenuItem() {Me.MnuColor, Me.MnuFont})
Me.mnuDialog.Text = "Dialog Box"
' Mục tùy chọn Color (MnuColor)
Me.MnuColor.Index = 0
Me.MnuColor.Text = "Color"
' Mục tùy chọn Font (MnuFont)
Me.MnuFont.Index = 1
Me.MnuFont.Text = "Font"
' Mục Exit trên menu ngang (MnuExit)

```

```
Me.MnuExit.Index = 2
```

```
Me.MnuExit.Text = "Exit"
```

```
' Đối tượng Menu (MainMenu1)
```

```
Me.MainMenu1.MenuItems.AddRange(New
```

```
System.Windows.Forms.MenuItem() {Me.MnuMessage, Me.mnuDialog,  
Me.MnuExit})
```

```
' Đối tượng Form frmMenu
```

```
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
```

```
Me.BackgroundImage =
```

```
CType(resources.GetObject("$this.BackgroundImage"),  
System.Drawing.Bitmap)
```

```
Me.ClientSize = New System.Drawing.Size(376, 197)
```

```
Me.Controls.AddRange(New System.Windows.Forms.Control() {Me.Label1})
```

```
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)
```

```
Me.Menu = Me.MainMenu1
```

```
Me.Name = "frmMenu"
```

```
Me.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen
```

```
Me.Text = "Message - Dialog Boxes"
```

```
Me.ResumeLayout(False)
```

```
End Sub
```

```
#End Region
```

' Kết thúc phần mã lệnh do VB.Net tự phát sinh trong quá trình thiết kế form

' Thủ tục được gọi khi người dùng nhấp mục Exit

```
Private Sub MnuExit_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MnuExit.Click
```

```
Dim nbutton
```



```
nbutton = MsgBox("Are You sure to quit (Yes/No)? ", MsgBoxStyle.YesNo
+ MsgBoxStyle.Question + MsgBoxStyle.SystemModal, "Message Box")
```

```
If nbutton = vbYes Then
```

```
End ' Kết thúc chương trình khi chọn nút Yes
```

```
End If
```

End Sub

' Khi người dùng nhấp mục OK

Private Sub MnuOK_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuOK.Click

```
Dim nbutton
```

```
nbutton = MsgBox("Message Box OK", MsgBoxStyle.OKOnly +
MsgBoxStyle.Information + MsgBoxStyle.SystemModal, "Message Box")
```

End Sub

' Khi người dùng nhấp mục OK - Cancel

Private Sub MnuOKCancel_Click(ByVal sender As System.Object, ByVal e

As System.EventArgs) Handles MnuOKCancel.Click

```
Dim nbutton
```

```
nbutton = MsgBox("Message Box OK - Cancel ", MsgBoxStyle.OKCancel
+ MsgBoxStyle.Exclamation + MsgBoxStyle.SystemModal, "Message Box")
```

End Sub

' Khi người dùng nhấp mục Yes - No

Private Sub MnuYesNo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuYesNo.Click

```
Dim nbutton
```

```
nbutton = MsgBox("Message Box Yes - No ", MsgBoxStyle.YesNo +
MsgBoxStyle.Critical + MsgBoxStyle.SystemModal, "Message Box")
```

End Sub

' Khi người dùng nhấp mục Retry - Cancel

Private Sub MnuRC_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuRC.Click

Dim nbutton

nbutton = MsgBox("Message Box Retry - Cancel ",

MsgBoxStyle.RetryCancel + MsgBoxStyle.Question, "Message Box")

End Sub

' Khi người dùng nhấp mục Abort - Retry - Ignore

Private Sub MnuARI_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuARI.Click

Dim nbutton

nbutton = MsgBox("Message Box Abort - Retry - Ignore ",

MsgBoxStyle.AbortRetryIgnore + MsgBoxStyle.Question, "Message Box")

End Sub

' Khi người dùng nhấp mục Yes - No - Cancel

Private Sub MnuYNC_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuYNC.Click

Dim nbutton

nbutton = MsgBox("Message Box Yes - No - Cancel ",

MsgBoxStyle.YesNoCancel + MsgBoxStyle.Question +

MsgBoxStyle.DefaultButton3, "Message Box")

End Sub

' Khi người dùng nhấp mục Color

Private Sub MnuColor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuColor.Click

```
ColorDialog1.ShowDialog() ' Hiển thị hộp thoại Color
' Đặt màu được chọn làm màu nền Label 1
Label1.BackColor = ColorDialog1.Color
```

End Sub

' Khi người dùng nhấp mục Font

Private Sub MnuFont_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MnuFont.Click

```
FontDialog1.ShowDialog() ' Hiển thị hộp thoại Font
' Gán thuộc tính phông vừa chọn cho văn bản trong Label 1
Label1.Font = FontDialog1.Font
```

End Sub

End Class

LƯU Ý

- ♦ Phần lớn mã lệnh trong lớp **frmMenu** do VB .Net tự động thêm vào trong quá trình thiết kế Form. Bạn chỉ cần nhập vào các dòng lệnh in nghiêng chữ đậm mà thôi.
-

Bước 12. Biên dịch và chạy thử chương trình .

Chọn lệnh File ➡ Save All để lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Xong nhấn F5 hoặc chọn lệnh Debug ➡ Start để chạy thử chương trình.

Kỹ thuật lập trình

Giờ đây, chúng ta sẽ ôn lại một chút về hàm MsgBox và cách sử dụng các hộp Font, Color trong VB .Net.

- Phương thức ShowDialog cho phép hiển thị hộp thoại Font hay Color tương ứng của đối tượng FontDialog1 hay ColorDialog1 đã đặt trên form.
- Thuộc tính Color của một đối tượng ColorDialog: trả lại màu sắc được chọn trong hộp thoại Color.
- Thuộc tính Font của một đối tượng FontDialog: trả lại phông chữ, kích cỡ, kiểu chữ được chọn trong hộp thoại Font.
- Tương tự như VB6, để hiển thị hộp thông báo, bạn có thể sử dụng hàm MsgBox. Thông báo sẽ hiển thị trong hộp thoại, chờ người dùng nhấp lên các nút lệnh trong hộp thông báo và trả về giá trị cho biết nút lệnh được nhấp.

Cú pháp: `MsgBox( message, style, title) as result` .

Trong đó:

- Message: nội dung thông báo.
- Style: giá trị (nguyên dương) xác định kiểu hộp thông báo (số nút, biểu tượng sẽ hiển thị bên trái hộp, nút nhận focus).
- Title: tiêu đề hộp thông báo
- Result: giá trị trả về của hàm MsgBox là một số nguyên dương xác định nút được nhấn.

MÁCH NƯỚC

- ♦ Lập trình viên có thể không sử dụng tham số Title, hộp thông báo sẽ lấy tiêu đề mặc định là tên project đã xây dựng.
 - ♦ Nếu không sử dụng tham số Style, mặc định hộp thông báo có kiểu một nút OK và không có biểu tượng bên trái hộp.
-
- Giá trị tham số Style có thể là tổng giá trị của kiểu nút, kiểu biểu tượng hiển thị, nút được chọn mặc định, ...

- Sau đây là bảng giá trị có thể sử dụng với tham số Style:

Tên hằng	Giá trị	Diễn giải
MsgBoxStyle.OKOnly	0	Hộp thông báo chỉ có 1 nút OK
MsgBoxStyle.OKCancel	1	Hộp thông báo có 2 nút OK, Cancel
MsgBoxStyle.AbortRetryIgnore	2	Hộp thông báo có 3 nút Abort, Retry và Ignore
MsgBoxStyle.YesNoCancel	3	Hộp thông báo có 3 nút Yes, No và Cancel
MsgBoxStyle.YesNo	4	Hộp thông báo có 2 nút Yes và No
MsgBoxStyle.RetryCancel	5	Hộp thông báo có 2 nút Retry, Cancel
MsgBoxStyle.MsgBoxHelp	16384	Hộp thông báo có nút Help
MsgBoxStyle.Critical	16	Biểu tượng X
MsgBoxStyle.Question	32	Biểu tượng ?
MsgBoxStyle.Information	64	Biểu tượng I
MsgBoxStyle.Exclamation	48	Biểu tượng !
MsgBoxStyle.SystemModal	4096	Có thể làm việc khác khi chưa trả lời hộp thông báo
MsgBoxStyle.ApplicationModal	0	Phải trả lời hộp thông báo trước khi làm việc khác
MsgBoxStyle.DefaultButton1	0	Nút đầu tiên trong hộp mặc định được chọn

MsgBoxStyle.DefaultButton2	256	Nút thứ 2 trong hộp mặc định được chọn
MsgBoxStyle.DefaultButton3	512	Nút thứ 3 trong hộp mặc định được chọn
MsgBoxStyle.MsgBoxRight	524288	Canh phải văn bản trong hộp thông báo

- Sau đây là bảng giá trị Result trả về ứng với từng nút được nhấn:

Tên hằng	Giá trị	Diễn giải
VbOK	1	Nút OK được nhấn
VbAbort	3	Nút Abort được nhấn
vbRetry	4	Nút Retry được nhấn
vbIgnore	5	Nút Ignore được nhấn
vbYes	6	Nút Yes được nhấn
VbNo	7	Nút No được nhấn

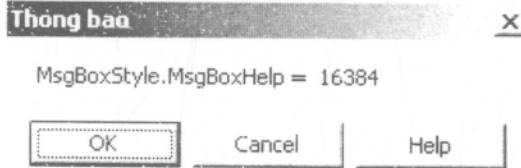
- Hình 3.20 và 3.21 là một số ví dụ minh họa về hộp thông báo sử dụng nhiều giá trị kết hợp trong tham số Style.
- + Câu lệnh sau sẽ cho bạn hộp thông báo như Hình 3.20.

Dim nb

nb = MsgBox("MsgBoxStyle.MsgBoxHelp = " &

Str(MsgBoxStyle.MsgBoxHelp), **MsgBoxStyle.OKCancel +**

MsgBoxStyle.MsgBoxHelp, "Thông báo")

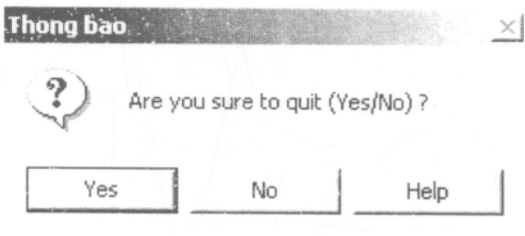


Hình 3.20 Hộp thông báo có thêm nút Help.

- + Câu lệnh sau sẽ cho bạn hộp thông báo như Hình 3.21.

Dim nb

```
nb = MsgBox("Are you sure to quit (Yes/No) ? " , MsgBoxStyle.YesNo
+ MsgBoxStyle.Question + MsgBoxStyle.MsgBoxHelp,
"Thông báo")
```



Hình 3.21 Hộp thông báo có biểu tượng ? và thêm nút Help.

Bài 4.

Chương trình Calendar

Mục đích chương trình

Bài thực hành này giúp bạn nắm cách thiết kế một tờ lịch để bàn và hiển thị nó trên màn hình nền Windows của bạn mà không che khuất cảnh vật phía sau.

Hoạt động chương trình

Giả sử màn hình nền Windows của bạn ban đầu sẽ như Hình 4.1.



Hình 4.1 Màn hình nền Windows trước khi chạy chương trình Calendar.

Sau khi thi hành chương trình Calendar, bạn sẽ được giao diện như Hình 4.2. Để ý rằng, tờ lịch hiển thị ở góc dưới bên phải màn hình, ngay

sát bên trên thanh Status, nhưng không che khuất hoàn toàn cảnh vật của màn hình Windows phía sau nó. Bạn có thể thấy được hàng thông hiện lơ mờ phía sau tờ lịch.



Hình 4.2 Giao diện màn hình Windows sau khi thi hành chương trình Calendar.

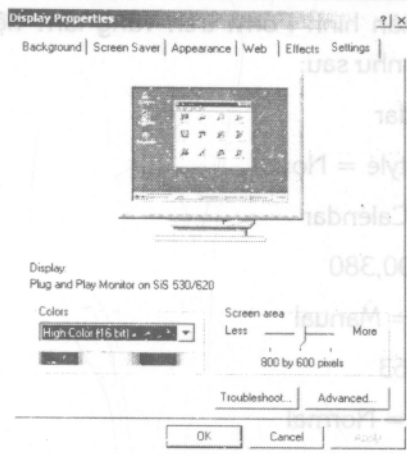
Hướng dẫn thiết kế

Bước 1. Trước tiên bạn cần kiểm tra xem các thiết đặt màn hình nền Windows ban đầu có đúng như quy định hay không.

Nhấp nút phải mouse lên màn hình nền Windows và chọn lệnh Properties trong menu nhanh để gọi cửa sổ Display Properties.

Để được màn hình nền như Hình 4.1, bạn cần chọn tập tin ảnh **Snow Trees.jpg** làm nền trọng mục Background của cửa sổ Display Properties với tùy chọn Picture Display = Center (tập tin **Snow Trees.jpg** sẽ có sẵn trong thư mục Windows khi bạn cài đặt phiên bản Windows 2000 Server).

Tiếp đó kiểm tra xem bạn định độ phân giải màn hình có đúng là 800 x 600 pixels hay không bằng cách nhấp vào mục Settings trong cửa sổ Display Properties (Hình 4.3).



Hình 4.3 Định độ phân giải và chế độ màu cho màn hình Windows của bạn.

Bước 2. Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➤ Programs ➤ Microsoft Visual Studio .Net ➤ Microsoft Visual Studio .Net.

Bước 3. Nhấp nút New Project hay chọn lệnh File ➤ New Project để gọi cửa sổ New Project.

Đặt tên tập tin lưu project: Name = **CALENDAR**

Định thư mục lưu project: Location = **C:\BTBNET**

Nhấp tùy chọn **Close Solution**.

Chọn Project types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.

Xong nhấp OK.

Bước 4. Chọn lệnh View ➤ Solution Explorer để mở cửa sổ Solution Explorer nếu chưa mở. Xong chọn form **Form1.vb** và nhấn F4 hay chọn lệnh View ➤ Properties Windows để mở cửa sổ Properties nếu chưa mở.

Định lại thuộc tính Filename = **Calendar.vb** trong cửa sổ Properties.

Bước 5. Nhấp màn hình Form trên vùng làm việc và định lại các thuộc tính của Form như sau:

- Text = Calendar
- FormBorderStyle = None
- Name = FrmCalendar
- Location = 600,380
- StartPosition = Manual
- Size = 192, 163
- WindowState = Normal
- Localizable = False
- ControlBox = True
- Helpbutton = False
- MaximizeBox: False
- MinimizeBox: False
- ShowInTaskbar: False
- Opacity = 50% (*form có độ trong suốt 50%, có thể thấy cảnh vật phía sau xuyên qua form*)

Bước 6. Nhấp nút phải mouse lên project *Calendar* trong cửa sổ Solution Explorer, xong chọn lệnh Properties trong menu nhanh để gọi hộp thoại *Calendar Property Pages*. Định lại form khởi động:

- Startup Object = *frmCalendar*, xong nhấp Apply
- Nhấp OK để trở lại màn hình thiết kế Form.

Bước 7. Đặt lên form 1 điều khiển MonthCalendar bằng cách nhấp đúp biểu tượng MonthCalendar trên Toolbox.

Sau đây là giá trị mặc định của một số thuộc tính chính mà bạn có thể chấp nhận, không cần chỉnh sửa gì thêm.

- Name = MonthCalendar1 (tên mặc định của điều khiển)
- Backcolor = window (màu nền tờ lịch là màu trắng)
- CalendarDimensions = 1, 1 (tờ lịch chỉ hiển thị 1 tháng - 1 hàng 1 cột)
- ForeColor = HighlightText (các ngày trong tháng có màu đen)
- TitleBackColor = ActiveCaption (nền hiển thị tên tháng và năm có màu xanh đậm)
- TitleForeColor = ActiveCaptionText (tên tháng và năm có màu trắng)
- Location = 0,0
- Size = 192, 155
- Maxdate = 12/31/9998 (có thể xem lịch từ năm 1753 đến 9998)
- MinDate = 1/1/1753
- FirstDateOfWeek = Default

Màn hình thiết kế của bạn bây giờ sẽ như Hình 4.4.



Hình 4.4 Kết quả thiết kế form `frmCalendar`.

Bước 8. Viết mã lệnh chương trình.

Bạn không cần viết thêm bất kỳ dòng lệnh nào cho chương trình này. Tuy nhiên, trình thiết kế form sẽ tự phát sinh đoạn mã lệnh sau trong quá trình bạn thiết kế form:

```
Public Class frmcalendar
```

```
    Inherits System.Windows.Forms.Form
```

```
#Region " Windows Form Designer generated code "
```

```
    Public Sub New()
```

```
        MyBase.New()
```

```
        InitializeComponent()
```

```
    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
```

```
    Private components As System.ComponentModel.IContainer
```

```
    Friend WithEvents MonthCalendar1 As System.Windows.Forms.MonthCalendar
```

```
    Friend WithEvents MonthCalendar2 As System.Windows.Forms.MonthCalendar
```

```
<System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
```

```
    Me.MonthCalendar1 = New System.Windows.Forms.MonthCalendar()
```

```
    Me.MonthCalendar2 = New System.Windows.Forms.MonthCalendar()
```

```
    Me.SuspendLayout()
```

```
' MonthCalendar1
Me.MonthCalendar1.Name = "MonthCalendar1"
Me.MonthCalendar1.TabIndex = 0

' MonthCalendar2
Me.MonthCalendar2.Location = New System.Drawing.Point(168, 152)
Me.MonthCalendar2.Name = "MonthCalendar2"
Me.MonthCalendar2.TabIndex = 1

' frmcalendar
Me.AccessibleRole = System.Windows.Forms.AccessibleRole.None
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(192, 163)
Me.Controls.AddRange(New System.Windows.Forms.Control()
    {Me.MonthCalendar2, Me.MonthCalendar1})
Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
Me.Location = New System.Drawing.Point(600, 380)
Me.MaximizeBox = False
Me.MinimizeBox = False
Me.Name = "frmcalendar"
Me.Opacity = 0.5
Me.ShowInTaskbar = False
Me.StartPosition = System.Windows.Forms.FormStartPosition.Manual
Me.Text = "Calendar"
Me.ResumeLayout(False)

End Sub

#End Region

End Class
```




Hình 4.6 Các tháng được hiển thị khi ngày hệ thống là 19/07/2003.

Thực chất, việc này rất đơn giản. Điều khiển MonthCalendar có hỗ trợ các thuộc tính quy định số tháng hiển thị cùng lúc trên điều khiển mà không cần viết thêm bất cứ dòng lệnh nào.

- Định lại kích cỡ và số tháng hiển thị cùng lúc trên điều khiển MonthCalendar1:
 - CalendarDimensions = 2, 3 (tờ lịch hiển thị 6 tháng - 2 hàng 3 cột)
 - Location = 0,0
 - Size = 388, 440
- Định lại kích cỡ và vị trí của form FrmCalendar:
 - Location = 200,0
 - StartPosition = Manual
 - Size = 388,440
 - Opacity = 69%
- Màn hình thiết kế form frmCalendar giờ đây sẽ như Hình 4.7:



Hình 4.7 Màn hình thiết kế form frmCalendar sau khi điều chỉnh các thuộc tính.

Bước 11. Chạy ứng dụng MonthCalendar từ Windows mà không cần khởi động VB.NET.

Trong quá trình Project Calendar, VB.Net tự động tạo tập tin Calendar.exe đặt trong thư mục C:\btvbn\Calendar\bin\Calendar.exe, cho phép chạy ứng dụng độc lập trong Windows, mà không cần mở VB.Net.

Chọn Start ➡ Run, trong cửa sổ Run, nhập vào lệnh sau:

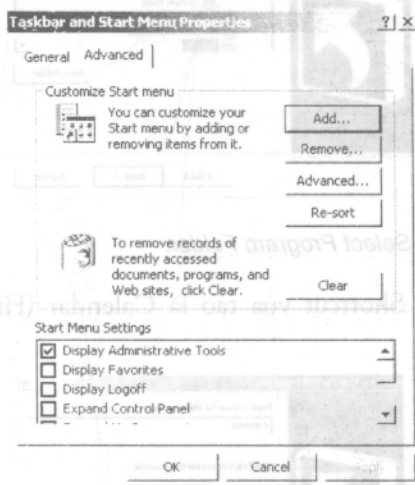
C:\btvbn\Calendar\bin\Calendar.exe

Xong nhấp OK. Ứng dụng sẽ được thi hành.

MÁCH NƯỚC

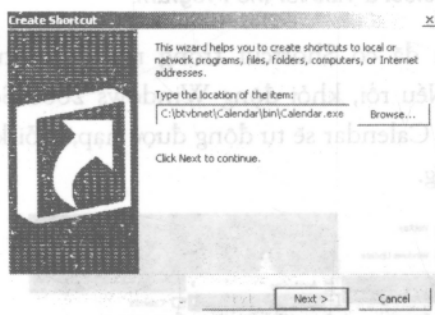
Nếu muốn ứng dụng tự động được nạp mỗi khi bạn bắt đầu Windows, hãy đặt ứng dụng vào nhóm StartUp theo cách sau:

- ♦ Start ➡ Settings ➡ Taskbar and Start menu để mở cửa sổ Properties (Hình 4.8).



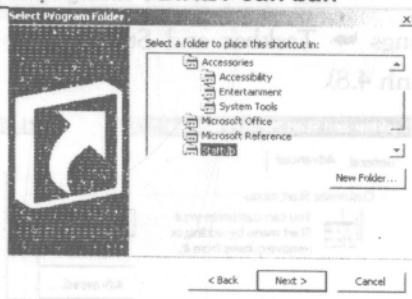
Hình 4.8 Cửa sổ Taskbar and start menu Properties.

- ♦ Nhấp Add mở cửa sổ Create Shortcut và nhập vào đường dẫn gọi tập tin sẽ thi hành (Hình 4.9). Xong nhấp Next.



Hình 4.9. Gọi thi hành ứng dụng Calendar.exe.

- ♦ Trong cửa sổ Select Program Folder, chọn nhóm Startup để bỏ sung Shortcut vừa tạo (Hình 4.10). Xong nhấp Next.



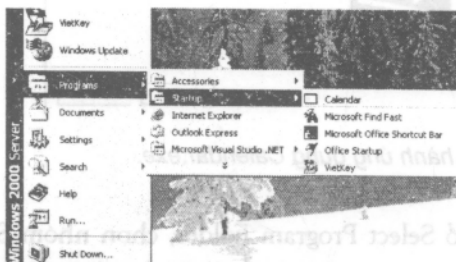
Hình 4.10 Cửa sổ Select Program Folder.

- ◆ Đặt tên cho Shortcut vừa tạo là Calendar (Hình 4.11). Xong nhấp Finish.



Hình 4.11 Cửa sổ Select a Title for the Program.

- ◆ Kiểm tra xem đã có Shortcut trong nhóm Startup hay chưa (Hình 4.12). Nếu rồi, khởi động Windows 2000 Server, bạn sẽ thấy ứng dụng Calendar sẽ tự động được nạp, mỗi khi Windows được khởi động.



Hình 4.12 Shortcut Calendar đã được đặt trong nhóm Startup.

Bài 5.

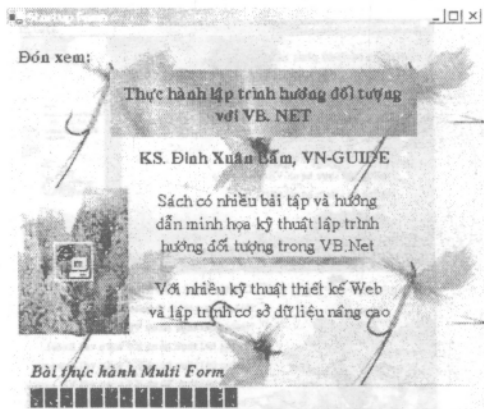
Chương trình Book

Mục đích chương trình

Bài thực hành này giúp bạn nắm cách sử dụng điều khiển Timer, Check box, Picture Box, Progress Bar (*thanh tiến trình*). Cũng như cách thiết kế các ứng dụng có nhiều form.

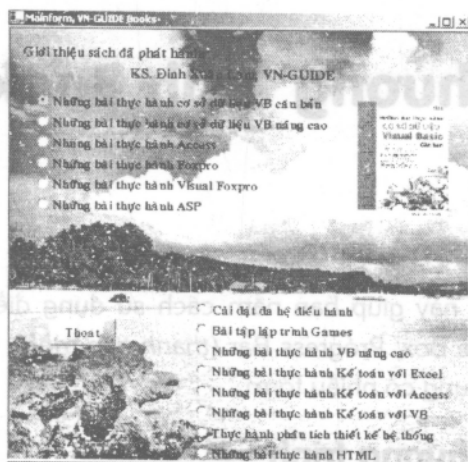
Hoạt động chương trình

Khi thi hành chương trình Book, bạn sẽ được màn hình khởi động có giao diện như Hình 5.1. Để ý rằng, giao diện này chỉ hiển thị trong thời gian 5 giây. Trong khoảng thời gian này mức xanh của thanh tiến trình bên dưới liên tục tăng dần sang phải để xác định phần trăm thời gian đã trôi qua (trong khoảng thời gian 5 giây).



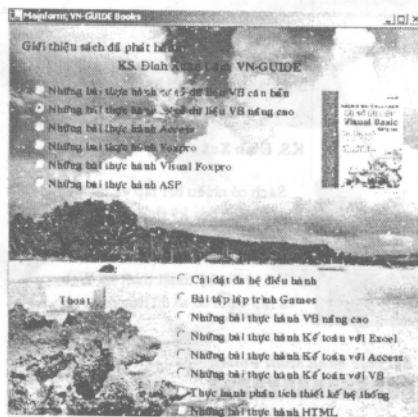
Hình 5.1 Giao diện màn hình khởi động Startup khi thi hành ứng dụng.

Sau 5 giây, màn hình khởi động sẽ biến mất và màn hình chính của ứng dụng sẽ được nạp như Hình 5.2.



Hình 5.2 Giao diện màn hình chính của ứng dụng Book sẽ hiển thị sau 5 giây thi hành ứng dụng.

Mặc định Check box đầu tiên được chọn và hình ảnh cuốn sách “Những bài thực hành cơ sở dữ liệu VB căn bản” sẽ hiển thị bên phải giao diện. Khi người sử dụng chọn một check box, hình ảnh cuốn sách tương ứng sẽ hiển thị (Hình 5.3).



Hình 5.3 Khi người dùng chọn check box thứ 2.

Chương trình sẽ kết thúc khi người dùng nhấp nút Thoát.

Hướng dẫn thiết kế

Bước 1. Do trong ứng dụng xây dựng đã sử dụng khá nhiều hình ảnh, vì thế, trước tiên bạn cần tập hợp các tập tin hình cần sử dụng và chép vào thư mục C:\BTVBNET\HINH.

Sau đây là những tập tin hình sẽ sử dụng:



Các tập tin hình nền: Fly away.jpg, Snow Trees.jpg, Gold Petals.jpg, Paradise.jpg, wuicon1.gif



Các tập tin hình sách: sach1.jpg, sach2.jpg, ... đến sach14.jpg

.....

MÁCH NƯỚC

- ♦ Các hình nền có copy từ thư mục c:\winnt\web\wallpaper hoặc c:\winnt\web (với winnt là thư mục cài đặt Windows 2000). Bạn có thể sử dụng hình tùy ý để thay thế các hình sách đã nêu trong quá trình thực tập bài này.
-

Bước 2. Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➤ Programs ➤ Microsoft Visual Studio .Net ➤ Microsoft Visual Studio .Net.

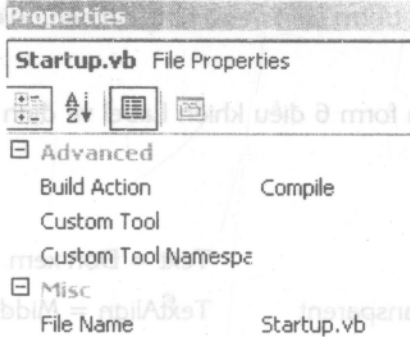
Bước 3. Nhấp nút New Project hay chọn lệnh File ➤ New Project để gọi cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = **BAI5**
- Định thư mục lưu project: Location = **C:\BTVBNET**
- Nhấp tùy chọn **Close Solution**.
- Chọn Project types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.
- Xong nhấp OK.

Bước 4. Thiết kế form Startup.vb.

4a. Chọn lệnh View ➤ Solution Explorer để mở cửa sổ Solution Explorer nếu chưa mở. Xong chọn tên form **Form1.vb** trong cửa sổ Solution Explorer và nhấn F4 hoặc chọn lệnh View ➤ Properties Windows để mở cửa sổ Properties (nếu chưa mở).

Định lại thuộc tính Filename = **Startup.vb** trong cửa sổ Properties (Hình 5.4).



Hình 5.4 Định lại tên tập tin cho form1.

4b. Nhấp màn hình Form trên vùng làm việc và định lại các thuộc tính của Form như sau:

- Text = Startup Form (tiêu đề form)
- FormBorderStyle = Sizable (kiểu viền form)
- Name = Startup (tên form)
- Location = 0,0 (vị trí hiển thị góc trên bên trái màn hình khi thi hành)
- StartPosition = CenterScreen (canh form ở giữa màn hình khi thi hành)
- Size = 456, 392 (kích thước form)
- WindowState = Normal (kích thước form hiển thị như lúc thiết kế)
- Localizable = False (không thể định lại kích thước form khi đang thi hành)
- ControlBox = True
- BackgroundImage: chọn tập tin Fly away.jpg làm ảnh nền của form

- *Opacity = 100% (form hiển thị rõ ràng như khi thiết kế, không có độ trong suốt)*
- 4c. Lần lượt đặt lên form 6 điều khiển Label và định các thuộc tính như sau:
 - Label 1:
Name = Label1 Text = Đón xem
BackColor = Transparent TextAlign = MiddleCenter
AutoSize = True
 - Label 2:
Name = Label2 AutoSize = False
Text = Thực hành lập trình hướng đối tượng với VB.NET
BorderStyle = None TextAlign = MiddleCenter
Size = 288,64 ImageAlign = MiddleCenter
Image = chọn tập tin Gold Pentals.jpg làm ảnh nền của Label
 - Label 3:
Name = Label3
Text = KS. Đinh Xuân Lâm, VN-GUIDE
 - Label 4:
Name = Label4
Text = Sách có nhiều bài tập và hướng dẫn minh họa
 - Label 5:
Name = Label5
Text = Với nhiều kỹ thuật thiết kế Web.....
 - Label 6:
Name = Label6

Text = Bài thực hành Multiform

Các Label đều có chung các thuộc tính sau:

Font = VNI-Times, 12

4d. Tiếp tục đặt lên form 2 điều khiển Picture Box và định các thuộc tính như sau:

- Picture 1:

Name = Picture1

BackgroundImage = *chọn tập tin Snow Trees.jpg làm ảnh nền điều khiển*

BorderStyle = None (*ảnh không có đường viền bao quanh*)

Image = *chọn tập tin wulcon1.jpg*

Sizemode = CenterImage (*canh ảnh giữa điều khiển Picture Box*)

- Picture 2:

Name = Picture2

BackgroundImage = *để trống*

BorderStyle = None (*ảnh không có đường viền bao quanh*)

Image = *chọn tập tin Gold Pentals.jpg*

Sizemode = StretchImage (*Nén ảnh vừa đúng kích cỡ của Picture Box*)

4e. Đặt lên form 1 điều khiển ProgressBar (*thanh tiến trình*) và định các thuộc tính như sau:

Name = Progressbar1

Location = 16,336

Size = 416,23

Value = 0

Step = 10

Maximum = 100

Minimum = 0

4f. Đặt lên form 2 điều khiển Timer và định các thuộc tính như sau:

- Timer 1:

Name = Timer1

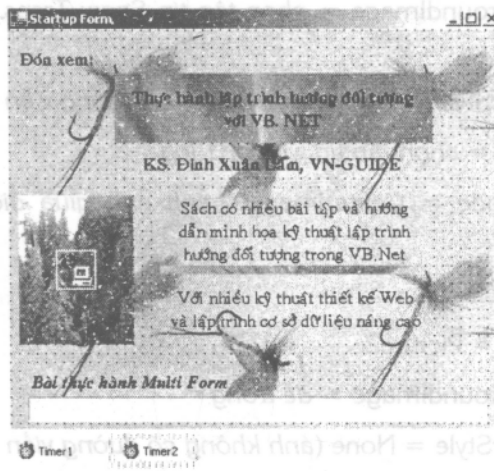
Interval = 5000

- Timer 2:

Name = Timer2

Interval = 250

4g. Định lại vị trí các điều khiển trên form để được kết quả thiết kế form như Hình 5.5.



Hình 5.5 Màn hình thiết kế form Startup.

4h. Viết mã lệnh xử lý cho form Startup (chỉ trình bày phần mã lệnh cần viết thêm, phần mã lệnh do VB.NET tự phát sinh trong quá trình thiết kế thiết tưởng không cần trình bày lại).

Public Class Startup

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

‘ phần mã lệnh do VB.NET tự phát sinh trong quá trình thiết kế

#End Region

‘ Timer1 giúp ẩn form Startup và gọi form chính Mainform sau thời gian 5 giây

Private Sub Timer1_Elapsed(ByVal sender As System.Object, ByVal e As System.Timers.ElapsedEventArgs) Handles Timer1.Elapsed

Dim a As Form

a = New Mainform()

Timer1.Stop()

Timer2.Stop()

Me.Hide()

a.Show()

End Sub

‘ Timer2 giúp điều khiển vệt sáng trên thanh tiến trình sau khoảng thời gian 0.25 giây

Private Sub Timer2_Elapsed(ByVal sender As System.Object, ByVal e As System.Timers.ElapsedEventArgs) Handles Timer2.Elapsed

If ProgressBar1.Value >= 100 Then

ProgressBar1.Value = 95

End If

ProgressBar1.Value = ProgressBar1.Value + 5

End Sub

End Class

MÁCH NƯỚC

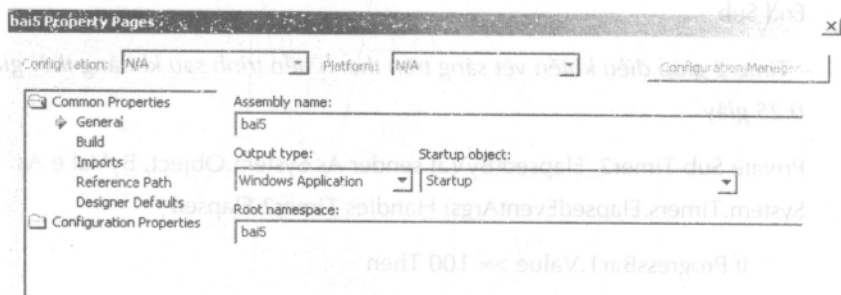
- ◆ Để mở thanh công cụ, chọn lệnh View ➡ Toolbox.
- ◆ Để ảnh Picture1 chỉ còn một đường vệt dài, bạn cần thu nhỏ ảnh và đặt ra sau hai label 5 và 6 bằng cách nhấp nút phải mouse lên Picture1 và chọn lệnh Send to back.

Bước 5. Đặt form Startup làm form khởi động.

Nhấp nút phải mouse lên project *Bai5* trong cửa sổ Solution Explorer, xong chọn lệnh Properties trong menu nhanh để gọi hộp thoại *Bai5 Property Pages* (Hình 5.6).

Định lại form khởi động:

- Startup Object = **Startup**, xong nhấp Apply.
- Nhấp OK để trở lại màn hình thiết kế Form.



Hình 5.6 Màn hình *Bai5 Property Pages*.

Bước 6. Thiết kế form Mainform.vb.

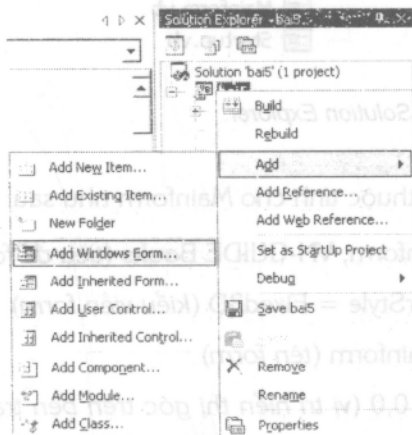
- 6a.** Để thêm một form mới vào Project, nhấp nút phải mouse lên project *Bai5* trong cửa sổ Solution Explorer để gọi menu nhanh, xong chọn lệnh Add ➡ Add Windows Form... (Hình 5.7) để mở cửa sổ Add New Item – bai5 (Hình 5.8).

Chọn Templates = Windows Form

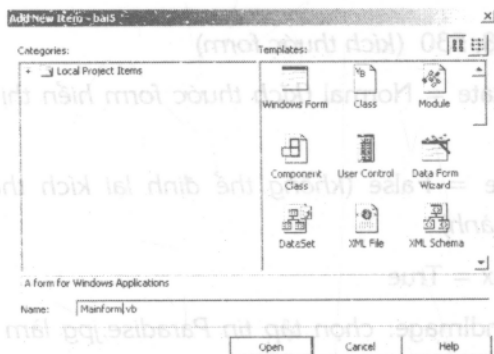
đặt Name = MainForm.vb

Xong nhấp Open.

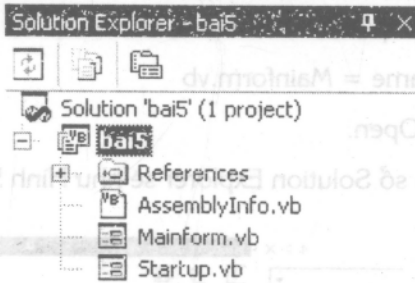
Kết quả của sổ Solution Explorer sẽ như Hình 5.9.



Hình 5.7 Menu nhanh.



Hình 5.8 Cửa sổ Add New Item.



Hình 5.9. Cửa sổ *Solution Explorer*.

6b. Định lại các thuộc tính cho *Mainform* như sau:

- Text = *Mainform*, VN-GUIDE Books (tiêu đề form)
- FormBorderStyle = *Fixed3D* (kiểu viền form)
- Name = *Mainform* (tên form)
- Location = 0,0 (vị trí hiển thị góc trên bên trái màn hình khi thi hành)
- StartPosition = *CenterScreen* (canh form ở giữa màn hình khi thi hành)
- Size = 538, 530 (kích thước form)
- WindowState = *Normal* (kích thước form hiển thị như lúc thiết kế)
- Localizable = *False* (không thể định lại kích thước form khi đang thi hành)
- ControlBox = *True*
- BackgroundImage: chọn tập tin *Paradise.jpg* làm ảnh nền của form
- Opacity = 100% (form hiển thị rõ ràng như khi thiết kế, không có độ trong suốt)

6c. Đặt lên form 2 Label và định lại các thuộc tính như sau:

- Label 1:
Name = Label1 Text = Giới thiệu sách đã phát hành:
- Label 2:
Name = Label2 Text = KS. Đinh Xuân Lâm, VN-GUIDE

GHI CHÚ

- ♦ Hai Label đều có chung các thuộc tính sau:

BackColor = Transparent TextAlign = MiddleCenter

AutoSize = True Font = VNI-Times, 12pt, Style = bold

6d. Đặt lên form 1 Button và định lại các thuộc tính sau:

Name = Button1 Text = &Thoát

TextAlign = MiddleCenter Font = VNI-Times, 12pt, Style = bold

BackgroundImage: *chọn tập tin Snow Trees.jpg làm ảnh nền button*

6e. Đặt lên form 15 Picture box và lần lượt định lại các thuộc tính như sau:

- Picture Box 1 đến Picture Box 14:

Name = PictureBox1 Image = *chọn tập tin ảnh Sach1.jpg*

Name = PictureBox2 Image = *chọn tập tin ảnh Sach2.jpg*

Name = PictureBox3 Image = *chọn tập tin ảnh Sach3.jpg*

.....

Name = PictureBox12 Image = *chọn tập tin ảnh Sach12.jpg*

Name = PictureBox13 Image = *chọn tập tin ảnh Sach13.jpg*

Name = PictureBox14 Image = *chọn tập tin ảnh Sach14.jpg*

- Picture Box 15:

Name = PictureBox Image = *chọn tập tin ảnh Sach1.jpg*
Size = 104,136 SizeMode = StretchImage
Visible = True

GHI CHÚ

- ♦ Các điều khiển PictureBox (trừ điều khiển Picture Box cuối cùng) đều có cùng giá trị các thuộc tính:

Size = 16,24 SizeMode = StretchImage
Visible = False

6f. Đặt lên form 14 Radio Button và định lại các thuộc tính như sau:

- Radio button 1:

Name = Radiobutton1 BackColor = Transparent
Checked = False Font = VNI-Times, 12pt, style = bold
Text = Những bài thực hành cơ sở dữ liệu VB căn bản

- Tương tự Radio button 2 đến Radio button 14:

Name = Radiobutton2
Text = Những bài thực hành cơ sở dữ liệu VB nâng cao
Name = Radiobutton3 Text = Những bài thực hành Access
Name = Radiobutton4 Text = Những bài thực hành Foxpro
Name = Radiobutton5
Text = Những bài thực hành Visual Foxpro
Name = Radiobutton6 Text = Những bài thực hành ASP
Name = Radiobutton7 Text = Cài đặt đa hệ điều hành
Name = Radiobutton8 Text = Bài tập lập trình Games

Name = Radiobutton9

Text = Những bài thực hành VB nâng cao

Name = Radiobutton10

Text = Những bài thực hành kế toán với Excel

Name = Radiobutton11

Text = Những bài thực hành kế toán với Access

Name = Radiobutton12

Text = Những bài thực hành kế toán với VB

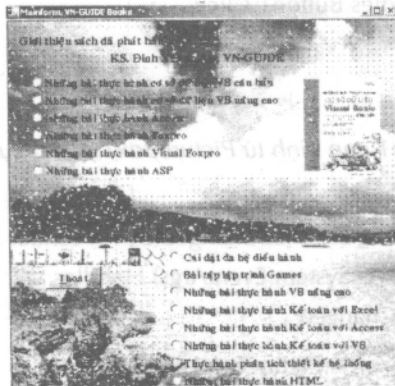
Name = Radiobutton13

Text = Thực hành phân tích thiết kế hệ thống

Name = Radiobutton14

Text = Những bài thực hành HTML

Màn hình thiết kế của bạn bây giờ sẽ như Hình 5.10.



Hình 5.10 Kết quả thiết kế form Mainform.

6g. Viết mã lệnh chương trình: nhấn F7 hay chọn lệnh View ➡ Code để bật cửa sổ Code và nhập vào đoạn mã lệnh sau:

```
Public Class MainForm
```

```
Inherits System.Windows.Forms.Form
```

```
#Region " Windows Form Designer generated code "
```

```
    ' Phần mã lệnh tự phát sinh bởi VB.NET
```

```
    ' trong quá trình thiết kế Form
```

```
#End Region
```

```
    ' Mặc định RadioButton1 được chọn khi form vừa nạp
```

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

```
    RadioButton1.Checked = True
```

```
End Sub
```

```
    ' Khi nhấp nút Thoát thì đóng form, kết thúc chương trình
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click
```

```
    Me.Close()
```

```
End Sub
```

```
    ' Khi nhấp RadioButton1 nạp hình từ PictureBox1 vào điều khiển
```

```
PictureBox
```

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
RadioButton1.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox1.Image
```

```
End Sub
```

```
    ' Khi nhấp RadioButton2 nạp hình từ PictureBox2 vào điều khiển
```

```
PictureBox
```

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton2.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox2.Image
```

```
End Sub
```

' Khi nhấp RadioButton3 nạp hình từ PictureBox3 vào điều khiển PictureBox

```
Private Sub RadioButton3_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton3.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox3.Image
```

```
End Sub
```

' Khi nhấp RadioButton4 nạp hình từ PictureBox4 vào điều khiển PictureBox

```
Private Sub RadioButton4_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton4.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox4.Image
```

```
End Sub
```

' Khi nhấp RadioButton5 nạp hình từ PictureBox5 vào điều khiển PictureBox

```
Private Sub RadioButton5_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton5.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox5.Image
```

```
End Sub
```

' Khi nhấp RadioButton6 nạp hình từ PictureBox6 vào điều khiển PictureBox

```
Private Sub RadioButton6_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
RadioButton6.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox6.Image
```

```
End Sub
```

' Khi nhấp RadioButton7 nạp hình từ PictureBox7 vào điều khiển PictureBox

```
Private Sub RadioButton7_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
RadioButton7.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox7.Image
```

```
End Sub
```

' Khi nhấp RadioButton8 nạp hình từ PictureBox8 vào điều khiển PictureBox

```
Private Sub RadioButton8_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
RadioButton8.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox8.Image
```

```
End Sub
```

' Khi nhấp RadioButton9 nạp hình từ PictureBox9 vào điều khiển PictureBox

```
Private Sub RadioButton9_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
RadioButton9.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox9.Image
```

```
End Sub
```

' Khi nhấp RadioButton10 nạp hình từ PictureBox10 vào điều khiển PictureBox

```
Private Sub RadioButton10_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton10.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox10.Image
```

```
End Sub
```

' Khi nhấp RadioButton11 nạp hình từ PictureBox11 vào điều khiển PictureBox

```
Private Sub RadioButton11_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton11.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox11.Image
```

```
End Sub
```

' Khi nhấp RadioButton12 nạp hình từ PictureBox12 vào điều khiển PictureBox

```
Private Sub RadioButton12_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton12.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox12.Image
```

```
End Sub
```

' Khi nhấp RadioButton13 nạp hình từ PictureBox13 vào điều khiển PictureBox

```
Private Sub RadioButton13_CheckedChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
RadioButton13.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox13.Image
```

```
End Sub
```

' Khi nhấp RadioButton14 nạp hình từ PictureBox14 vào điều khiển PictureBox

```
Private Sub RadioButton14_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
RadioButton14.CheckedChanged
```

```
    Me.PictureBox.Image = Me.PictureBox14.Image
```

```
End Sub
```

```
End Class
```

Bước 7. Biên dịch và chạy thử chương trình.

- Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.
- Chọn lệnh Build ➡ Build Solution để biên dịch project.
- Xong nhấn F5 hoặc chọn lệnh Debug ➡ Start để chạy thử chương trình.

Bước 8. Chạy ứng dụng Bai5 từ Windows mà không cần khởi động VB.NET

Trong quá trình biên dịch Project Bai5 (*Build Solution*), VB.Net tự động tạo tập tin Bai5.exe đặt trong thư mục C:\btvbn\Bai5\bin\Bai5.exe, cho phép chạy ứng dụng độc lập trong Windows, mà không cần mở VB.Net.

Chọn Start ➡ Run, trong cửa sổ Run, nhập vào lệnh sau:

```
C:\btvbn\Bai5\bin\Bai5.exe
```

Xong nhấp OK. Ứng dụng sẽ được thi hành.

Bài 6.

Chương trình TextEditor

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với kỹ thuật lập trình hướng đối tượng trong VB.NET. Một trong những thế mạnh của VB.NET so với Visual Basic 6.0.

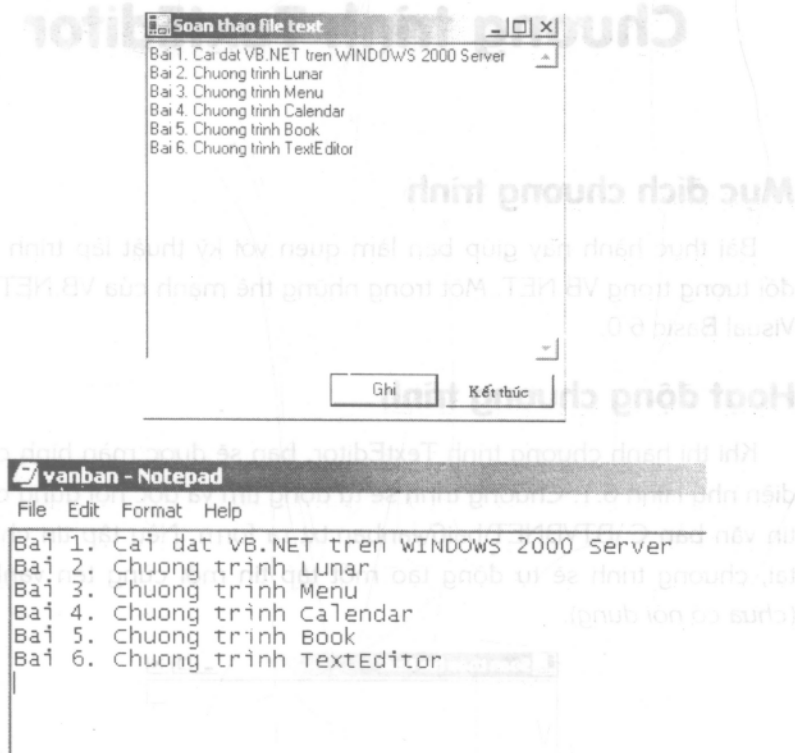
Hoạt động chương trình

Khi thi hành chương trình TextEditor, bạn sẽ được màn hình có giao diện như Hình 6.1. Chương trình sẽ tự động tìm và đọc nội dung của tập tin văn bản C:\BTBNET\bai6\vanban.txt ra form. Nếu tập tin chưa tồn tại, chương trình sẽ tự động tạo một tập tin mới cùng tên vanban.txt (chưa có nội dung).



Hình 6.1 Giao diện màn hình khi form TextEditor được nạp.

Người dùng có thể nhập mới hay điều chỉnh hoặc xóa nội dung văn bản hiển thị trên form. Xong nhấp nút Ghi để lưu nội dung đó vào tập tin C:\BTBNET\bai6\vanban.txt (Hình 6.2). Chương trình sẽ kết thúc khi người dùng nhấp nút Kết thúc.



Hình 6.2 Màn hình form TextEditor và nội dung tập tin vanban.txt sau khi nhập mới.

Hướng dẫn thiết kế

Bước 1. Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➤ Programs ➤ Microsoft Visual Studio .Net ➤ Microsoft Visual Studio .Net.

Bước 2. Nhấp nút New Project hay chọn lệnh File ➡ New Project để gọi cửa sổ New Project

- Đặt tên tập tin lưu project: Name = *TextEditor*
- Định thư mục lưu project: Location = *C:\BTVBNET*
- Nhấp tùy chọn **Close Solution**.
- Chọn Project types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.
- Xong nhấp OK.

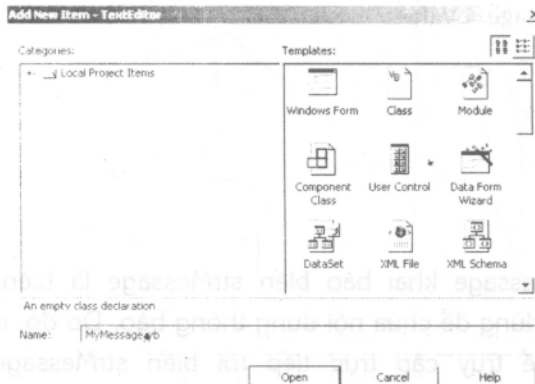
Bước 3. Tạo lớp MyMessage.vb

* Mục đích:

Định nghĩa lớp MyMessage có một thuộc tính là MsgText. Các lớp khác có thể truy xuất hoặc gán giá trị cho thuộc tính này.

Chọn lệnh View ➡ Solution Explorer để mở cửa sổ Solution Explorer nếu chưa thấy.

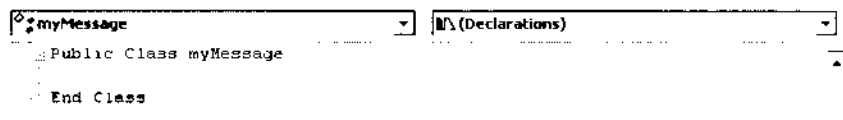
Thêm lớp MyMessage vào cửa sổ Solution Explorer bằng cách chọn lệnh Add ➡ Add Class. Hộp thoại Add New Item sẽ được mở như Hình 6.3.



Hình 6.3 Hộp thoại Add New Item.

Nhập vào tên lớp Name = MyMessage.vb, xong nhấn Open.

Một lớp trống với tên MyMessage sẽ được tạo ra trong project TextEditor như Hình 6.4.



Hình 6.4 Lớp trống MyMessage vừa mới tạo.

Định nghĩa thuộc tính MsgText cho lớp MyMessage bằng cách nhập vào đoạn mã lệnh sau:

```
Public Class myMessage
    Private strMessage As String
    Public Property MsgText() As String
        Get
            MsgText = strMessage
        End Get
        Set(ByVal Value As String)
            strMessage = Value
        End Set
    End Property
End Class
```

*** Giải thích:**

- Lớp MyMessage khai báo biến strMessage là biến thành viên (private), dùng để chứa nội dung thông báo. Do đó, các lớp khác không thể truy cập trực tiếp tới biến strMessage trong lớp MyMessage.

- Do vậy, bạn cần định nghĩa thuộc tính `MsgText` để các lớp khác có thể xác lập (ghi) hoặc truy xuất (đọc) giá trị của biến thành viên `strMessage` trong lớp `MyMessage`.
- Trong tính năng `Set`, tham số `Value` sẽ nhận giá trị từ các lớp khác để gán cho biến thành viên `strMessage`.
- Tham số `Value` trong tính năng `Set` có cùng kiểu dữ liệu với thuộc tính `MsgText` (trong trường hợp này là cùng kiểu `String`).

MÁCH NƯỚC

- ◆ Lớp `MyMessage` không kế thừa từ lớp khác, do đó không có từ khóa `Inherits`.
- ◆ Công dụng của lớp `MyMessage` là lưu văn bản thông báo, cho phép các lớp khác truy xuất và điều chỉnh nội dung thông báo đã lưu trong lớp.
- ◆ Các thuộc tính được định nghĩa trong lớp bắt đầu với từ khóa `Property`, sau đó cung cấp tính năng `Get` (truy xuất giá trị thuộc tính) hoặc `Set` (thay đổi giá trị thuộc tính) hoặc cả hai (thuộc tính đọc ghi).
- ◆ Tham số `Value` trong tính năng `Set` phải cùng kiểu dữ liệu với thuộc tính chứa tính năng `Set`.
- ◆ Các lớp khác có thể truy xuất giá trị biến thành viên `strMessage` trong lớp `MyMessage` qua thuộc tính `MsgText` đã định nghĩa như sau:

```
strThongbao = MyMessage.MsgText
```

```
hoặc strThongbao = MyMessage.MsgText()
```

- ◆ Các lớp khác có thể gán giá trị cho biến thành viên `strMessage` trong lớp `MyMessage` qua thuộc tính `MsgText` đã định nghĩa như sau:

```
MyMessage.MsgText = "Nội dung thông báo"
```

Bước 4. Tạo lớp SmartMessage.vb.*** Mục đích:**

Định nghĩa lớp SmartMessage có một thuộc tính là MsgText. Các lớp khác có thể truy xuất hoặc gán giá trị cho thuộc tính này. Giá trị thuộc tính MsgText sẽ được đọc từ tập tin hoặc lưu vào tập tin

C:\BTVBNET\BAI6\vanban.txt

Điều này sẽ giúp lưu giá trị thuộc tính MsgText mà không bị mất khi chương trình kết thúc hoặc mất điện như lớp MyMessage.

- Chọn lệnh View ➡ Solution Explorer để mở cửa sổ Solution Explorer nếu chưa thấy.

Thêm lớp SmartMessage vào cửa sổ Solution Explorer bằng cách chọn lệnh Add ➡ Add Class. Hộp thoại Add New Item sẽ được mở như Hình 6.3.

Nhập vào tên lớp Name = SmartMessage.vb, xong nhấp Open.

Một lớp trống với tên SmartMessage sẽ được tạo ra trong project TextEditor:

```
Public Class SmartMessage
```

```
End Class
```

- Định nghĩa thuộc tính MsgText cho lớp SmartMessage bằng cách nhập vào đoạn mã lệnh sau:

```
Imports System.IO
```

```
Imports System.Runtime.Serialization
```

```
Imports System.Runtime.Serialization.Formatters.Binary
```

```
Public Class SmartMessage
```

Inherits myMessage

Public Function Write() As Boolean

Dim MsgFile As StreamWriter

Dim MsgFile2 As FileStream

Write = True

Try

MsgFile2 = File.Create("C:\BTVBNET\BAI6\vanban.txt")

MsgFile2.Close()

MsgFile = File.AppendText("C:\BTVBNET\BAI6\vanban.txt")

MsgFile.Write(Me.MsgText)

Catch

Write = False

End Try

MsgFile.Close()

End Function

Public Function Read() As Boolean

Dim MsgFile As StreamReader

Dim MsgFile2 As FileStream

Read = True

Try

If File.Exists("C:\BTVBNET\BAI6\vanban.txt")=True Then

MsgFile = File.OpenText("C:\BTVBNET\BAI6\vanban.txt")

Else

MsgFile2 = File.Create("C:\BTVBNET\BAI6\vanban.txt")

MsgFile2.Close()

```
MsgFile = File.OpenText("C:\BTVBNET\BAI6\vanban.txt")
End If
Me.MsgText = MsgFile.ReadToEnd
Catch
    Read = False
End Try
MsgFile.Close()
End Function
Public Sub New()
    MyBase.New()
    Read()
End Sub
End Class
```

GIẢI THÍCH:

- Có thể thấy lớp SmartMessage tương tự lớp MyMessage, nhưng bổ sung thêm các tính năng Read (*đọc trạng thái thuộc tính MsgText từ tập tin*) và Write (*lưu trạng thái thuộc tính MsgText vào tập tin*). Trong đó, *trạng thái thuộc tính MsgText* là giá trị hiện hành của biến thành viên strMessage.
- Thay vì phải xây dựng lại các tính năng Set, Get và dữ liệu thành viên strMessage cho thuộc tính MsgText trong lớp SmartMessage, bạn có thể kế thừa mọi tính năng đó từ lớp MyMessage, và chỉ cần xây dựng bổ sung các tính năng Read và Write cho thuộc tính MsgText trong lớp SmartMessage mà thôi. Đây chính là một trong những đặc điểm nổi bật của phương pháp lập trình hướng đối tượng.

- Các lệnh:

Imports System.IO

Imports System.Runtime.Serialization

Imports System.Runtime.Serialization.Formatters.Binary

Cho phép lớp SmartMessage sử dụng tập tin I/O và tính năng tạo chuỗi nối tiếp (Serialization)

- Câu lệnh Inherits myMessage trong khai báo lớp, cho phép lớp SmartMessage có mọi tính năng như lớp MyMessage, mà không cần định nghĩa lại.
- Định nghĩa phương thức Write (tính năng mới) cho lớp SmartMessage:

Public Function Write() As Boolean

 ' Các lệnh xử lý

End Function

- Giải thích phương thức Write trong lớp SmartMessage:

 ' Khai báo hai đối tượng tập tin thuộc các lớp StreamWriter và lớp FileStream

 ' Các lớp StreamWriter và lớp FileStream đã được định nghĩa sẵn

 ' trong thư viện lớp .NET FRAMEWORK, cung cấp các tính năng xử lý tập tin

 ' như tạo tập tin, mở tập tin, thêm dữ liệu vào tập tin, đóng tập tin, ...

Dim MsgFile As StreamWriter

Dim MsgFile2 As FileStream

 ' Chưa có lỗi xảy ra khi thực hiện phương thức Write

Write = True

 ' Bắt đầu bắt và xử lý lỗi trong quá trình tạo tập tin, viết dữ liệu tới tập tin

Try

' Tạo tập tin văn bản mới (vanban.txt), nếu tập tin đã tồn tại,

' thì tập tin sẽ bị xóa trước khi tạo mới

MsgFile2 = File.Create("C:\BTVBNET\BA16\vanban.txt")

' đóng tập tin vừa tạo

MsgFile2.Close()

' mở tập tin vừa tạo để thêm dữ liệu vào

MsgFile = File.AppendText("C:\BTVBNET\BA16\vanban.txt")

' ghi dữ liệu từ thuộc tính MsgText của lớp vào tập tin Vanban.txt

MsgFile.Write(Me.MsgText)

' bắt lỗi xảy ra

Catch

Write = False

End Try

' đóng tập tin vanban.txt

MsgFile.Close()

- Định nghĩa phương thức Read (tính năng mới) cho lớp SmartMessage:

Public Function Read() As Boolean

' Các lệnh xử lý

End Function

- Giải thích phương thức Read trong lớp SmartMessage:

' Khai báo hai đối tượng tập tin thuộc các lớp StreamWriter và lớp FileStream

Dim MsgFile As StreamReader

Dim MsgFile2 As FileStream

Read = True

Try

‘ Kiểm tra xem đã tồn tại tập tin vanban.txt trong thư mục qui định hay chưa

‘ Nếu rồi thì mở tập tin, nếu chưa thì tạo tập tin mới cùng tên và mở tập tin

If File.Exists("C:\BTVBNET\BAI6\vanban.txt")=True Then

MsgFile= File.OpenText("C:\BTVBNET\BAI6\vanban.txt")

Else

MsgFile2 = File.Create("C:\BTVBNET\BAI6\vanban.txt")

MsgFile2.Close()

MsgFile= File.OpenText("C:\BTVBNET\BAI6\vanban.txt")

End If

‘ Đọc toàn bộ nội dung tập tin đang mở Vanban.txt và gán cho thuộc

‘ tính MsgText của đối tượng lớp SmartMessage hiện hành

Me.MsgText = MsgFile.ReadToEnd

Catch

Read = False

End Try

‘ Đóng tập tin vanban.txt đang mở

MsgFile.Close()

End Function

- Giải thích phương thức New trong lớp SmartMessage:

Public Sub New()

‘ Khởi động lớp MyMessage là lớp gốc của SmartMessage

MyBase.New()

‘ Gọi thi hành phương thức Read

Read()

End Sub

MÁCH NƯỚC

- ◆ Có thể xây dựng một lớp (lớp SmartMessage) kế thừa các tính năng (dữ liệu thành viên, Get và Set) từ lớp khác (lớp MyMessage).
 - ◆ Tính kế thừa trong lập trình hướng đối tượng là khả năng lớp kế thừa mọi dữ liệu, thuộc tính và phương thức của lớp khác (các tính năng).
 - ◆ Để tự động thực hiện phương thức Read mỗi khi có một đối tượng mới của lớp SmartMessage được khởi tạo (Instance), nhằm tự động gán nội dung tập tin vanban.txt tới thuộc tính MsgText cho đối tượng mới, bạn cần định nghĩa phương thức New trong lớp SmartMessage.
 - ◆ Phương thức New trong lớp sẽ tự thi hành mỗi khi có một đối tượng mới của lớp được khởi tạo.
-

Nhận xét: Đến đây coi như bạn đã hoàn tất việc định nghĩa lớp SmartMessage. Giờ đây, để sử dụng lớp này từ một lớp bất kỳ, bạn chỉ cần khởi tạo một đối tượng thuộc lớp SmartMessage là có thể sử dụng thuộc tính MsgText và hai phương thức Read, Write đã định nghĩa.

Bước 5. Thiết kế form TextEditor.vb.

* **Mục đích:**

Mình họa cách sử dụng lớp SmartMessage từ lớp TextEditor.

- 5a.** Chọn lệnh View ➡ Solution Explorer để mở cửa sổ Solution Explorer nếu chưa mở. Xong chọn tên form **Form1.vb** trong cửa sổ Solution Explorer và nhấn F4 hoặc chọn lệnh View ➡ Properties Windows mở cửa sổ Properties (nếu chưa mở).

Định lại thuộc tính Filename = **TextEditor.vb** trong cửa sổ Properties (Hình 6.4).



Hình 6.4 Định lại tên tập tin cho form1.

5b. Nhấp màn hình Form trên vùng làm việc và định lại các thuộc tính của Form như sau:

- Text = Soạn thảo tập tin text (tiêu đề form)
- FormBorderStyle = Sizable (kiểu viền form)
- Name = TextEditor (tên form)
- StartPosition = WindowsDefaultLocation (vị trí form khi thi hành là mặc định)
- Size = 296, 288 (kích thước form lúc thiết kế)
- WindowState = Normal (kích thước form hiển thị như khi thiết kế)
- Localizable = True (có thể định lại kích thước form khi đang thi hành)
- Opacity = 100% (form hiển thị rõ ràng như khi thiết kế, không có độ trong suốt)

5c. Lần lượt đặt lên form 1 điều khiển Textbox và định các thuộc tính như sau:

Name = Textbox (*tên Textbox*)

Text = rỗng (*nội dung văn bản hiển thị trong Textbox*)

BackColor = màu trắng (*màu nền Textbox*)

ForeColor = màu đen (*màu văn bản hiển thị trên Textbox*)

Scrollbars = Vertical (*có thanh cuộn đứng bên phải Textbox*)

Multiline = True (*cho phép nhập nhiều dòng*)

Wordwrap = True (*cho phép cuộn văn bản dài*)

Size = 288,224 (*kích thước Textbox*)

Font = VNI-Times, 10

Location = 0,0 (*vị trí bắt đầu Textbox ở góc trên bên trái màn hình form*)

MaxLength = 32767 (*chiều dài tối đa của nội dung văn bản nhập trong Textbox*)

TabIndex = 0 (*đặt Textbox là điều khiển sẽ nhận focus đầu tiên*)

5d. Tiếp tục đặt lên form 2 điều khiển Button và định các thuộc tính như sau:

- Button 1:

Name = BtnOK

Text = Ghi

TextAlign = MiddleCenter

TabIndex = 1 (*đặt Button là điều khiển nhận focus từ Textbox khi nhấn Tab*)

Font = VNI-Times, 8, Bold

- Button 2:

Name = btnCancel

Text = &Kết thúc

TextAlign = MiddleCenter

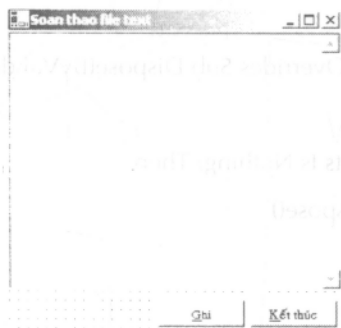
TabIndex = 2 (đặt Button là điều khiển nhận focus từ button 1 khi nhấn Tab)

Font = VNI-Times, 8, Bold

MÁCH NƯỚC

- ♦ Có thể định ảnh nền cho Button bằng thuộc tính BackgroundImage. Khi đó thuộc tính BackGroundImage sẽ được ưu tiên hơn thuộc tính BackColor.
- ♦ Để hiển thị hình ảnh trên Button bên cạnh dòng chữ văn bản, dùng thuộc tính Image.

5e. Định lại vị trí các điều khiển trên form để được kết quả thiết kế form như Hình 6.5.



Hình 6.5 Màn hình thiết kế form TextEditor.

5f. Viết mã lệnh xử lý cho form TextEditor (kể cả phần mã lệnh do VB.NET tự phát sinh trong quá trình thiết kế).

```
Public Class TextEditor
```

```
Inherits System.Windows.Forms.Form
```

```
'Phần mã lệnh do VB.NET tự phát sinh, bạn chỉ cần điều chỉnh lại các
```

```
'lệnh khai báo và thủ tục New cho phù hợp với tên lớp SmartMessage
```

```
' đã định nghĩa
```

```
#Region " Windows Form Designer generated code "
```

```
Private components As System.ComponentModel.Container
```

```
Private WithEvents TextBox As TextBox
```

```
Dim WithEvents TextEditor As System.Windows.Forms.Form
```

```
' Khai báo Msg như một đối tượng thuộc lớp SmartMessage
```

```
Private Msg As SmartMessage
```

```
Public Sub New(ByRef Msg As SmartMessage)
```

```
    MyBase.New()
```

```
    Me.Msg = Msg
```

```
    InitializeComponent()
```

```
    TextBox.Text = Msg.MsgText()
```

```
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
```

```
Friend WithEvents TextBox1 As System.Windows.Forms.TextBox
```

```
Friend WithEvents BtnOK As System.Windows.Forms.Button
Friend WithEvents BtnCancel As System.Windows.Forms.Button
<System.Diagnostics.DebuggerStepThrough> Private Sub
    InitializeComponent()
Dim resources As System.Resources.ResourceManager = New
    System.Resources.ResourceManager(GetType(TextEditor))
Me.TextBox = New System.Windows.Forms.TextBox()
Me.BtnOK = New System.Windows.Forms.Button()
Me.BtnCancel = New System.Windows.Forms.Button()
Me.SuspendLayout()
    TextBox
Me.TextBox.AcceptsReturn = True
Me.TextBox.AccessibleDescription =
    CType(resources.GetObject("TextBox.AccessibleDescription"), String)
Me.TextBox.AccessibleName =
    CType(resources.GetObject("TextBox.AccessibleName"), String)
Me.TextBox.Anchor = CType(resources.GetObject("TextBox.Anchor"),
    System.Windows.Forms.AnchorStyles)
Me.TextBox.AutoSize =
    CType(resources.GetObject("TextBox.AutoSize"), Boolean)
Me.TextBox.BackgroundImage =
    CType(resources.GetObject("TextBox.BackgroundImage"),
    System.Drawing.Image)
Me.TextBox.Dock = CType(resources.GetObject("TextBox.Dock"),
    System.Windows.Forms.DockStyle)
Me.TextBox.Enabled = CType(resources.GetObject("TextBox.Enabled"),
    Boolean)
```



```
Me.TextBox.Font = CType(resources.GetObject("TextBox.Font"),
    System.Drawing.Font)
Me.TextBox.ImeMode = CType(resources.GetObject("TextBox.ImeMode"),
    System.Windows.Forms.ImeMode)
Me.TextBox.Location =
    CType(resources.GetObject("TextBox.Location"), System.Drawing.Point)
Me.TextBox.MaxLength =
    CType(resources.GetObject("TextBox.MaxLength"), Integer)
Me.TextBox.Multiline =
    CType(resources.GetObject("TextBox.Multiline"), Boolean)
Me.TextBox.Name = "TextBox"
Me.TextBox.PasswordChar =
    CType(resources.GetObject("TextBox.PasswordChar"), Char)
Me.TextBox.RightToLeft =
    CType(resources.GetObject("TextBox.RightToLeft"),
    System.Windows.Forms.RightToLeft)
Me.TextBox.ScrollBars =
    CType(resources.GetObject("TextBox.ScrollBars"),
    System.Windows.Forms.ScrollBars)
Me.TextBox.Size = CType(resources.GetObject("TextBox.Size"),
    System.Drawing.Size)
Me.TextBox.TabIndex =
    CType(resources.GetObject("TextBox.TabIndex"), Integer)
Me.TextBox.Text = resources.GetString("TextBox.Text")
Me.TextBox.TextAlign =
    CType(resources.GetObject("TextBox.TextAlign"),
    System.Windows.Forms.HorizontalAlignment)
```

```
Me.TextBox.Visible = CType(resources.GetObject("TextBox.Visible"),
    Boolean)

Me.TextBox.WordWrap =
    CType(resources.GetObject("TextBox.WordWrap"), Boolean)
    1BtnOK

Me.BtnOK.AccessibleDescription =
    CType(resources.GetObject("BtnOK.AccessibleDescription"), String)

Me.BtnOK.AccessibleName =
    CType(resources.GetObject("BtnOK.AccessibleName"), String)

Me.BtnOK.Anchor = CType(resources.GetObject("BtnOK.Anchor"),
    System.Windows.Forms.AnchorStyles)

Me.BtnOK.BackgroundImage =
    CType(resources.GetObject("BtnOK.BackgroundImage"),
    System.Drawing.Image)

Me.BtnOK.Dock = CType(resources.GetObject("BtnOK.Dock"),
    System.Windows.Forms.DockStyle)

Me.BtnOK.Enabled = CType(resources.GetObject("BtnOK.Enabled"),
    Boolean)

Me.BtnOK.FlatStyle = CType(resources.GetObject("BtnOK.FlatStyle"),
    System.Windows.Forms.FlatStyle)

Me.BtnOK.Font = CType(resources.GetObject("BtnOK.Font"),
    System.Drawing.Font)

Me.BtnOK.Image = CType(resources.GetObject("BtnOK.Image"),
    System.Drawing.Image)

Me.BtnOK.ImageAlign =
    CType(resources.GetObject("BtnOK.ImageAlign"),
    System.Drawing.ContentAlignment)
```

```
Me.BtnOK.ImageIndex =  
    CType(resources.GetObject("BtnOK.ImageIndex"), Integer)  
Me.BtnOK.ImeMode = CType(resources.GetObject("BtnOK.ImeMode"),  
    System.Windows.Forms.ImeMode)  
Me.BtnOK.Location = CType(resources.GetObject("BtnOK.Location"),  
    System.Drawing.Point)  
Me.BtnOK.Name = "BtnOK"  
Me.BtnOK.RightToLeft =  
    CType(resources.GetObject("BtnOK.RightToLeft"),  
    System.Windows.Forms.RightToLeft)  
Me.BtnOK.Size = CType(resources.GetObject("BtnOK.Size"),  
    System.Drawing.Size)  
Me.BtnOK.TabIndex = CType(resources.GetObject("BtnOK.TabIndex"),  
    Integer)  
Me.BtnOK.Text = resources.GetString("BtnOK.Text")  
Me.BtnOK.TextAlign = CType(resources.GetObject("BtnOK.TextAlign"),  
    System.Drawing.ContentAlignment)  
Me.BtnOK.Visible = CType(resources.GetObject("BtnOK.Visible"), Boolean)  
'BtnCancel  
Me.BtnCancel.AccessibleDescription =  
    CType(resources.GetObject("BtnCancel.AccessibleDescription"), String)  
Me.BtnCancel.AccessibleName =  
    CType(resources.GetObject("BtnCancel.AccessibleName"), String)  
Me.BtnCancel.Anchor =  
    CType(resources.GetObject("BtnCancel.Anchor"),  
    System.Windows.Forms.AnchorStyles)
```

```
Me.BtnCancel.BackgroundImage =  
    CType(resources.GetObject("BtnCancel.BackgroundImage"),  
        System.Drawing.Image)  
Me.BtnCancel.Dock = CType(resources.GetObject("BtnCancel.Dock"),  
    System.Windows.Forms.DockStyle)  
Me.BtnCancel.Enabled =  
    CType(resources.GetObject("BtnCancel.Enabled"), Boolean)  
Me.BtnCancel.FlatStyle =  
    CType(resources.GetObject("BtnCancel.FlatStyle"),  
        System.Windows.Forms.FlatStyle)  
Me.BtnCancel.Font = CType(resources.GetObject("BtnCancel.Font"),  
    System.Drawing.Font)  
Me.BtnCancel.Image = CType(resources.GetObject("BtnCancel.Image"),  
    System.Drawing.Image)  
Me.BtnCancel.ImageAlign =  
    CType(resources.GetObject("BtnCancel.ImageAlign"),  
        System.Drawing.ContentAlignment)  
Me.BtnCancel.ImageIndex =  
    CType(resources.GetObject("BtnCancel.ImageIndex"), Integer)  
Me.BtnCancel.ImeMode =  
    CType(resources.GetObject("BtnCancel.ImeMode"),  
        System.Windows.Forms.ImeMode)  
Me.BtnCancel.Location =  
    CType(resources.GetObject("BtnCancel.Location"),  
        System.Drawing.Point)  
Me.BtnCancel.Name = "BtnCancel"
```

```
Me.BtnCancel.RightToLeft =  
    CType(resources.GetObject("BtnCancel.RightToLeft"),  
        System.Windows.Forms.RightToLeft)  
Me.BtnCancel.Size = CType(resources.GetObject("BtnCancel.Size"),  
    System.Drawing.Size)  
Me.BtnCancel.TabIndex =  
    CType(resources.GetObject("BtnCancel.TabIndex"), Integer)  
Me.BtnCancel.Text = resources.GetString("BtnCancel.Text")  
Me.BtnCancel.TextAlign =  
    CType(resources.GetObject("BtnCancel.TextAlign"),  
        System.Drawing.ContentAlignment)  
Me.BtnCancel.Visible = CType(resources.GetObject("BtnCancel.Visible"),  
    Boolean)  
'TextEditor  
Me.AccessibleDescription =  
    CType(resources.GetObject("$this.AccessibleDescription"), String)  
Me.AccessibleName =  
    CType(resources.GetObject("$this.AccessibleName"), String)  
Me.Anchor = CType(resources.GetObject("$this.Anchor"),  
    System.Windows.Forms.AnchorStyles)  
Me.AutoScaleBaseSize =  
    CType(resources.GetObject("$this.AutoScaleBaseSize"),  
        System.Drawing.Size)  
Me.AutoScroll = CType(resources.GetObject("$this.AutoScroll"), Boolean)  
Me.AutoScrollMargin =  
    CType(resources.GetObject("$this.AutoScrollMargin"),  
        System.Drawing.Size)
```

```
Me.AutoScrollMinSize =  
    CType(resources.GetObject("$this.AutoScrollMinSize"),  
        System.Drawing.Size)  
Me.BackgroundImage =  
    CType(resources.GetObject("$this.BackgroundImage"),  
        System.Drawing.Image)  
Me.ClientSize = CType(resources.GetObject("$this.ClientSize"),  
    System.Drawing.Size)  
Me.Controls.AddRange(New System.Windows.Forms.Control()  
    {Me.BtnCancel, Me.BtnOK, Me.TextBox})  
Me.Dock = CType(resources.GetObject("$this.Dock"),  
    System.Windows.Forms.DockStyle)  
Me.Enabled = CType(resources.GetObject("$this.Enabled"), Boolean)  
Me.Font = CType(resources.GetObject("$this.Font"), System.Drawing.Font)  
Me.Icon = CType(resources.GetObject("$this.Icon"), System.Drawing.Icon)  
Me.ImeMode = CType(resources.GetObject("$this.ImeMode"),  
    System.Windows.Forms.ImeMode)  
Me.Location = CType(resources.GetObject("$this.Location"),  
    System.Drawing.Point)  
Me.MaximumSize = CType(resources.GetObject("$this.MaximumSize"),  
    System.Drawing.Size)  
Me.MinimumSize = CType(resources.GetObject("$this.MinimumSize"),  
    System.Drawing.Size)  
Me.Name = "TextEditor"  
Me.RightToLeft = CType(resources.GetObject("$this.RightToLeft"),  
    System.Windows.Forms.RightToLeft)  
Me.StartPosition = CType(resources.GetObject("$this.StartPosition"),  
    System.Windows.Forms.FormStartPosition)
```

```
Me.Text = resources.GetString("$this.Text")
Me.Visible = CType(resources.GetObject("$this.Visible"), Boolean)
Me.ResumeLayout(False)
End Sub
#End Region
' Các thủ tục bạn tự viết thêm
' Khi nhấp button Kết thúc sẽ kết thúc chương trình
Private Sub BtnCancel_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BtnCancel.Click
    End
End Sub
' Khi nhấp button Ghi sẽ ghi dữ liệu trong TextBox lên tập tin Vanban.txt
Private Sub BtnOK_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BtnOK.Click
    Msg.MsgText = TextBox.Text()
    Msg.Write()
End Sub
End Class
```

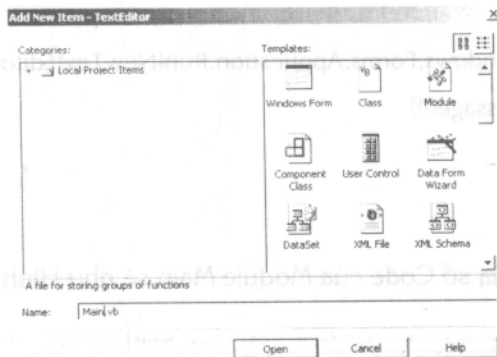
MÁCH NƯỚC

- ◆ Msg là một đối tượng thuộc lớp SmartMessage
 - ◆ Lớp Texteditor kế thừa lớp System.Windows.Forms.Form
-

Bước 6. Bổ sung Sub Main.

Do phương thức New định nghĩa trong form TextEditor (lớp TextEditor) có lấy tham số (Msg), nên bạn cần bắt đầu trình ứng dụng từ module Main, thay vì bắt đầu từ form chính TextEditor, như các trường hợp thông thường.

- Để thêm một module mới vào project, chọn lệnh Project ➔ Add Module để mở cửa sổ Add New Item – TextEditor (Hình 6.6).



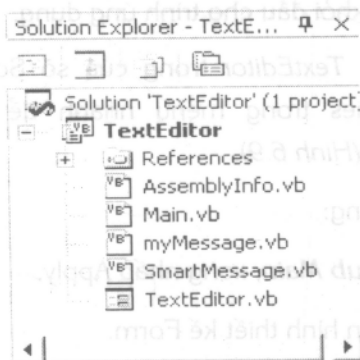
Hình 6.6 Cửa sổ Add New Item – TextEditor.

Chọn Templates = Windows Form

đặt Name = Main.vb

Xong nhấp Open.

Kết quả của sổ Solution Explorer lúc này sẽ như Hình 6.7.



Hình 6.7 Cửa sổ Solution Explorer.

- Định nghĩa đoạn mã sau cho Module Main trong cửa sổ Code:

```
Module Main
```

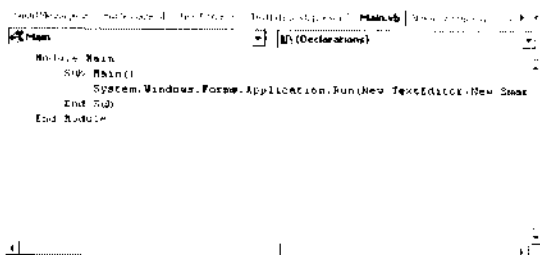
```
Sub Main()
```

```
System.Windows.Forms.Application.Run(New TextEditor(New  
SmartMessage()))
```

```
End Sub
```

```
End Module
```

- Kết quả của cửa sổ Code của Module Main sẽ như Hình 6.8.



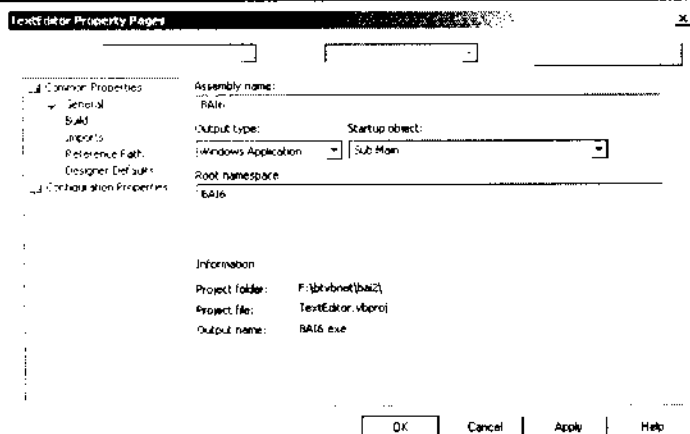
Hình 6.8 Cửa sổ Code của Module Main.

Bước 7. Xác định điểm khởi đầu cho trình ứng dụng.

Nhấp phải tên project *TextEditor* trong cửa sổ Solution Explorer, xong chọn lệnh Properties trong menu nhanh để gọi hộp thoại *TextEditor Property Pages* (Hình 6.9).

Định lại thủ tục khởi động:

- Startup Object = **Sub Main**, xong nhấp Apply.
- Nhấp OK trở lại màn hình thiết kế Form.



Hình 6.9 Màn hình TextEditor Property Pages.

Bước 8. Biên dịch và chạy thử chương trình.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Xong nhấn F5 hoặc chọn lệnh Debug ➡ Start để chạy thử chương trình.

Bài 7.

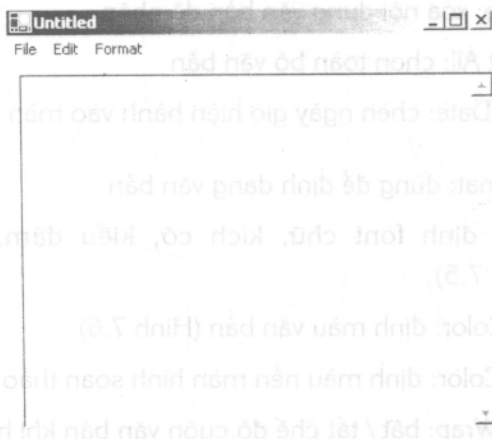
Chương trình Notepad

Mục đích chương trình

Bài thực hành này hướng dẫn bạn cách vận dụng các kỹ thuật đã trình bày trong bài thực hành 3 và 6 để xây dựng một trình soạn thảo văn bản, mô phỏng trình Notepad mà bạn thường gặp trong Windows. Từ đó, hiểu rõ những tiện lợi của kỹ thuật lập trình hướng đối tượng.

Hoạt động chương trình

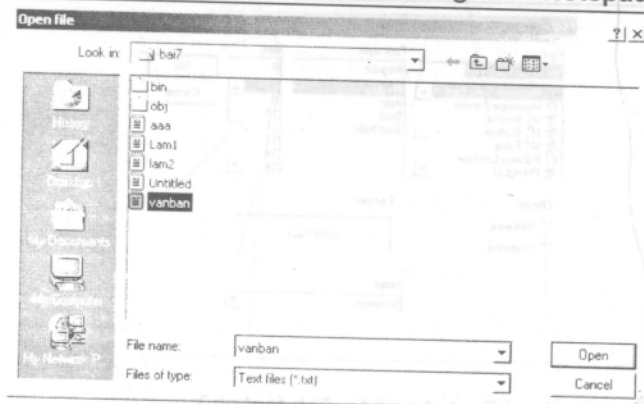
Khi thi hành chương trình, bạn sẽ được màn hình có giao diện như Hình 7.1. Chương trình sẽ tự động mở 1 màn hình trống với tiêu đề mặc định Untitled để bạn soạn thảo văn bản.



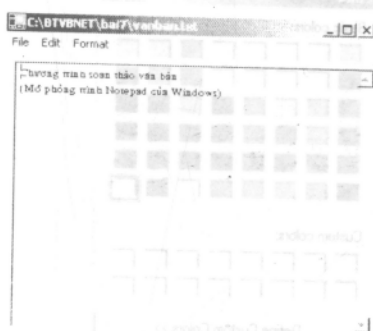
Hình 7.1 Giao diện màn hình khi form Notepad được nạp.

Người dùng có thể nhập mới, nạp tập tin văn bản đã có, điều chỉnh và lưu trữ lại tập tin. Sau đây, là các chức năng xử lý chính của chương trình trên menu:

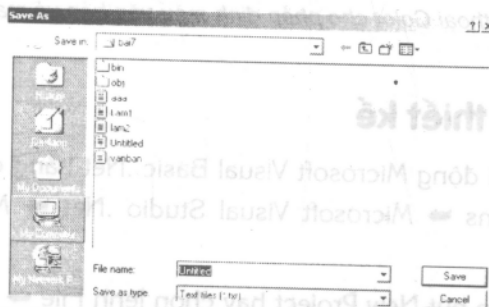
- **Menu File:** dùng để xử lý tập tin văn bản
 - **New:** tạo 1 cửa sổ trống mới để nhập văn bản (Hình 7.1)
 - **Open:** mở 1 tập tin văn bản đang tồn tại (Hình 7.2 và 7.3)
 - **Save:** lưu nội dung hiện hành vào tập tin đang mở
 - **Save As:** lưu nội dung hiện hành tới 1 tập tin mới (Hình 7.4)
 - **Exit:** kết thúc chương trình
- **Menu Edit:** dùng để hiệu chỉnh văn bản
 - **Undo:** hủy bỏ lệnh hiệu chỉnh văn bản vừa thực hiện
 - **Cut:** cắt vùng văn bản đang chọn vào Clipboard
 - **Copy:** sao chép vùng văn bản đang chọn vào Clipboard
 - **Paste:** dán nội dung Clipboard vào màn hình soạn thảo
 - **Delete:** xóa nội dung văn bản đã nhập
 - **Select All:** chọn toàn bộ văn bản
 - **Time/Date:** chèn ngày giờ hiện hành vào màn hình soạn thảo
- **Menu Format:** dùng để định dạng văn bản
 - **Font:** định font chữ, kích cỡ, kiểu đậm, nghiêng, ... (Hình 7.5)
 - **ForeColor:** định màu văn bản (Hình 7.6)
 - **BackColor:** định màu nền màn hình soạn thảo (Hình 7.6)
 - **WordWrap:** bật / tắt chế độ cuộn văn bản khi hết dòng.



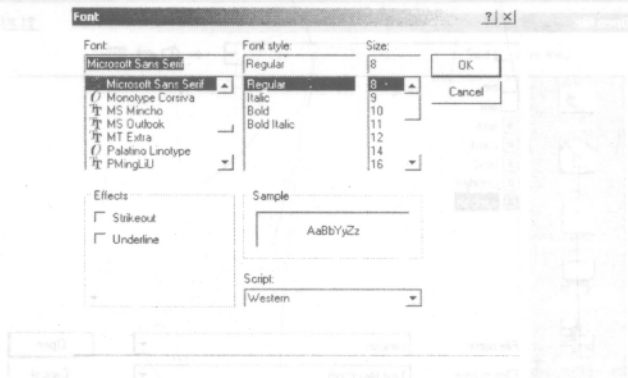
Hình 7.2 Hộp thoại Open file cho phép chọn tập tin văn bản cần mở.



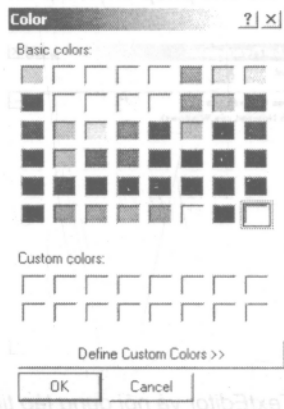
Hình 7.3 Màn hình form TextEditor và nội dung tập tin vanban.txt sau khi mở.



Hình 7.4 Hộp thoại Save As cho phép lưu văn bản hiện hành vào một tập tin mới.



Hình 7.5 Hộp thoại Font cho phép định font chữ.



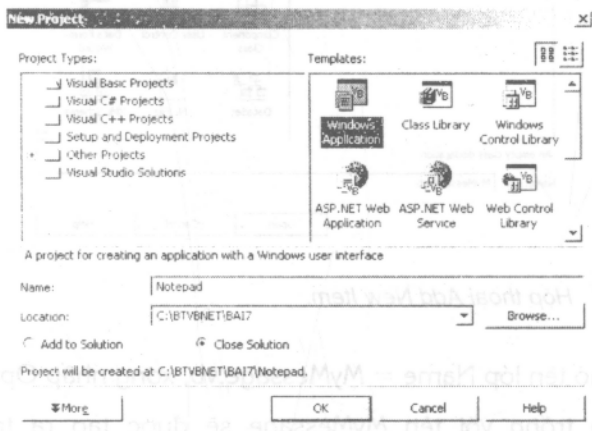
Hình 7.6 Hộp thoại Color cho phép định màu văn bản và màu nền.

Hướng dẫn thiết kế

Bước 1. Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net.

Bước 2. Nhấp nút New Project hay chọn lệnh File ➡ New Project để gọi cửa sổ New Project (Hình 7.7).

- Đặt tên tập tin lưu project: Name = *Notepad*
- Định thư mục lưu project: Location = *C:\BTVBNET\BAI7*
- Nhấp tùy chọn **Close Solution**
- Chọn Project types = *Visual Basic Projects*, xong nhấp biểu tượng *Windows Application* trong mục *Templates*
- Xong nhấp OK.



Hình 7.7 Cửa sổ New Project

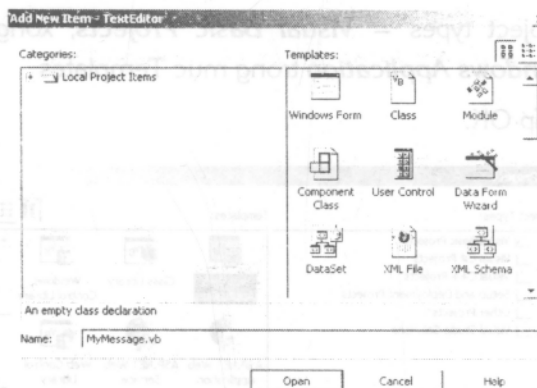
Bước 3. Tạo lớp MyMessage.vb.

* **Mục đích:**

Tương tự như bài thực hành 6, định nghĩa lớp MyMessage có 1 thuộc tính là MsgText. Các lớp khác có thể truy xuất hoặc gán giá trị cho thuộc tính này.

Chọn lệnh View ➡ Solution Explorer để bật cửa sổ Solution Explorer nếu chưa thấy.

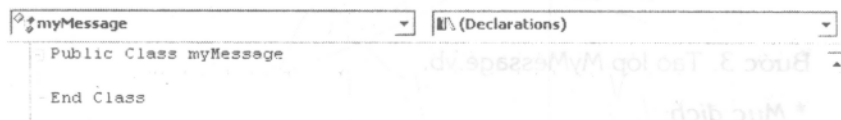
Thêm lớp MyMessage vào cửa sổ Solution Explorer bằng cách chọn lệnh Add ➡ Add Class. Hộp thoại Add New Item sẽ được mở ra như Hình 7.8.



Hình 7.8 Hộp thoại Add New Item.

Nhập vào tên lớp Name = MyMessage.vb, xong nhấn Open.

Một lớp trống với tên MyMessage sẽ được tạo ra trong project TextEditor như Hình 7.9.



Hình 7.9 Lớp trống MyMessage vừa mới tạo.

Định nghĩa thuộc tính MsgText cho lớp MyMessage bằng cách nhập vào đoạn mã lệnh sau:

```
Public Class myMessage
    Private strMessage As String

    Public Property MsgText() As String
```



```

Get
    MsgText = strMessage
End Get
Set(ByVal Value As String)
    strMessage = Value
End Set
End Property
End Class

```

Bước 4. Tạo lớp SmartMessage.vb.

* **Mục đích:**

Định nghĩa lớp SmartMessage có một thuộc tính là MsgText. Các lớp khác có thể truy xuất hoặc gán giá trị cho thuộc tính này. Giá trị thuộc tính MsgText sẽ được đọc từ tập tin hoặc lưu vào tập tin có tên được truyền kèm trong lời gọi phương thức Read hoặc Write.

- Chọn lệnh View ➡ Solution Explorer kích hoạt cửa sổ Solution Explorer nếu chưa thấy.
- Thêm lớp SmartMessage vào cửa sổ Solution Explorer bằng cách chọn lệnh Add ➡ Add Class. Hộp thoại Add New Item sẽ mở ra như Hình 7.8.

Nhập vào tên lớp Name = SmartMessage.vb, xong nhấp Open.

Một lớp trống với tên SmartMessage sẽ được tạo ra trong project. TextEditor:

```

Public Class SmartMessage

End Class

```

- Định nghĩa thuộc tính MsgText cho lớp SmartMessage bằng cách nhập vào đoạn mã lệnh sau:

```
Imports System.IO
```

```
Imports System.Runtime.Serialization
```

```
Imports System.Runtime.Serialization.Formatters.Binary
```

```
Public Class SmartMessage
```

```
    Inherits myMessage
```

```
    Public Function Write(ByVal sfilename As String) As Boolean
```

```
        Dim MsgFile As StreamWriter
```

```
        Dim MsgFile2 As FileStream
```

```
        Write = True
```

```
        Try
```

```
            MsgFile2 = File.Create(sfilename)
```

```
            MsgFile2.Close()
```

```
            MsgFile = File.AppendText(sfilename)
```

```
            MsgFile.Write(Me.MsgText)
```

```
        Catch
```

```
            Write = False
```

```
        End Try
```

```
        MsgFile.Close()
```

```
    End Function
```

```
    Public Function Read(ByVal sfilename As String) As Boolean
```

```
        Dim MsgFile As StreamReader
```

```
        Dim MsgFile2 As FileStream
```

```
        Read = True
```

```
        Try
```

```

If File.Exists(sfilename) = True Then
    MsgFile = File.OpenText(sfilename)
Else
    MsgFile2 = File.Create(sfilename)
    MsgFile2.Close()
    MsgFile = File.OpenText(sfilename)
End If
Me.MsgText = MsgFile.ReadToEnd
Catch
    Read = False
End Try
MsgFile.Close()
End Function
Public Sub New()
    MyBase.New()
End Sub
End Class

```

Giải thích:

- Có thể thấy lớp SmartMessage vừa xây dựng tương tự như lớp SmartMessage đã trình bày trong bài thực hành 6. Tuy nhiên, lớp đã có sự điều chỉnh nhỏ như sau: các tính năng Read (đọc trạng thái thuộc tính MsgText từ tập tin) và Write (lưu trạng thái thuộc tính MsgText vào tập tin) có 1 đối số, đó là tên tập tin để đọc hoặc ghi dữ liệu.

- Định nghĩa phương thức Write cho lớp SmartMessage:

```

Public Function Write(ByVal sfilename As String) As Boolean
    ' Các lệnh xử lý
End Function

```

- Định nghĩa phương thức Read cho lớp SmartMessage:
Public Function Read(ByVal sfilename As String) As Boolean
 ' Các lệnh xử lý
End Function

Bước 5. Thiết kế form Notepad.vb.

* **Mục đích:**

Mình họa cách sử dụng lớp SmartMessage từ lớp Notepad.

5a. Chọn lệnh View ➡ Solution Explorer để mở cửa sổ Solution Explorer nếu chưa mở. Xong chọn tên form **Form1.vb** trong cửa sổ Solution Explorer và nhấn F4 hoặc chọn lệnh View ➡ Properties Windows kích hoạt cửa sổ Properties (nếu chưa kích hoạt).

Định lại thuộc tính Filename = **Notepad.vb** trong cửa sổ Properties.

5b. Nhấp màn hình Form trên vùng làm việc và định lại các thuộc tính của form như sau:

- Text = Untitled (tiêu đề form)
- FormBorderStyle = Sizable (kiểu viền form)
- Name = Notepad (tên form)
- StartPosition = WindowsDefaultLocation (vị trí form khi thi hành là mặc định)
- Size = 360, 312 (kích thước form lúc thiết kế)
- WindowState = Normal (kích thước form hiển thị như khi thiết kế)
- Localizable = True (có thể định lại kích thước form khi đang thi hành)

- Opacity = 100% (form hiển thị rõ ràng như khi thiết kế, không có độ trong suốt)

5c. Lần lượt đặt lên form 1 điều khiển TextBox và định các thuộc tính như sau:

Name = Textbox (tên Text box)

Text = rỗng (nội dung văn bản hiển thị trong Text box)

BackColor = màu trắng (màu nền Text box)

ForeColor = màu đen (màu văn bản hiển thị trên Text box)

Scrollbars = Vertical (có thanh cuộn đứng bên phải Text box)

Multiply = True (cho phép nhập nhiều dòng)

Wordwrap = True (cho phép cuộn văn bản dài)

Size = 344, 256 (kích thước Text box)

Font = VNI-Times, 10

Location = 8,8 (vị trí bắt đầu Text box ở góc trên bên trái màn hình form)

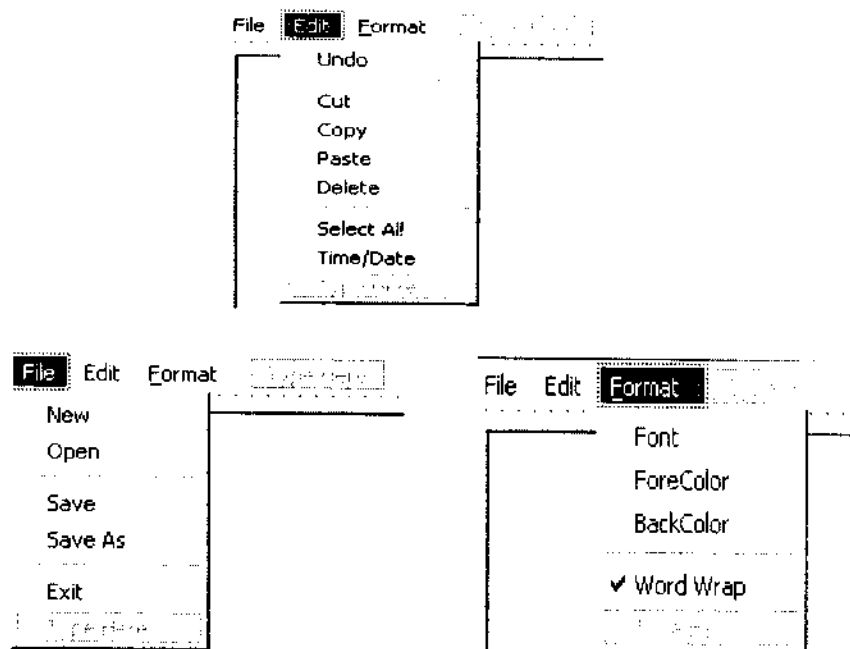
MaxLength = 32767 (chiều dài tối đa của nội dung văn bản nhập trong Textbox)

TabIndex = 0 (đặt Textbox là điều khiển sẽ nhận focus đầu tiên)

5d. Thiết kế Menu cho form Notepad.

- Nhấp biểu tượng MainMenu trên Toolbox, một đối tượng Menu mới sẽ xuất hiện dưới cửa sổ thiết kế form. Xong nhấp vào vùng làm việc form Notepad.
- Nhấp vào mục Type Here và nhập từ "File". Vb.Net tự động thêm các mục Type Here bên phải và dưới mục menu bạn vừa nhập để chờ bạn định nghĩa các lệnh trên menu ngang và các tùy chọn trong menu xổ xuống ứng với từng lệnh.

- Tiếp tục thực hiện việc định nghĩa menu để được kết quả như Hình 7.10.



Hình 7.10 Menu form Notepad sau khi hoàn tất.

- Định lại tên cho các tùy chọn Menu.

Chọn mục File, trong cửa sổ Properties, đặt thuộc tính *Name* là *MenuItem1*. Tương tự, bạn định lại tên cho các mục chọn trên menu như sau:

Mục chọn menu (thuộc tính Text)	Tên mục chọn menu (thuộc tính Name)
File	MenuItem1
New	MNew
Open	MOpen

-	MenuItem2
Save	Msave
Save As	Msaveas
-	MenuItem3
Exit	Mexit
Edit	MenuItem4
Undo	Mundo
-	MenuItem5
Cut	Mcut
Copy	Mcopy
Paste	Mpaste
Delete	Mdelete
-	Mnultem6
Select All	Mselect
Time / Date	Mdate
&Format	MenuItem7
Font	Mfont
ForeColor	Mfore
BackColor	Mback
-	MenuItem8
Word Wrap	Mwordwrap

Riêng tùy chọn Word wrap có thuộc tính Checked = True.

Định lại thuộc tính Menu cho form Notepad trong cửa sổ Properties như sau:

Menu = MainMenu1

- 5e.** Bổ sung 4 điều khiển FontDialog, ColorDialog, OpenFileDialog và SaveFileDialog từ ToolBox vào form frmMenu bằng cách lần lượt nhấp đúp lên biểu tượng của 4 điều khiển đó. Khi đó, vùng bên dưới màn hình thiết kế form sẽ xuất hiện thêm 4 đối tượng với tên FontDialog1, ColorDialog1, OpenFileDialog1 và SaveFileDialog1.

5f. Tiếp tục đặt lên form 1 điều khiển DateTimePicker1 và định các thuộc tính như sau:

Name = DateTimePicker1

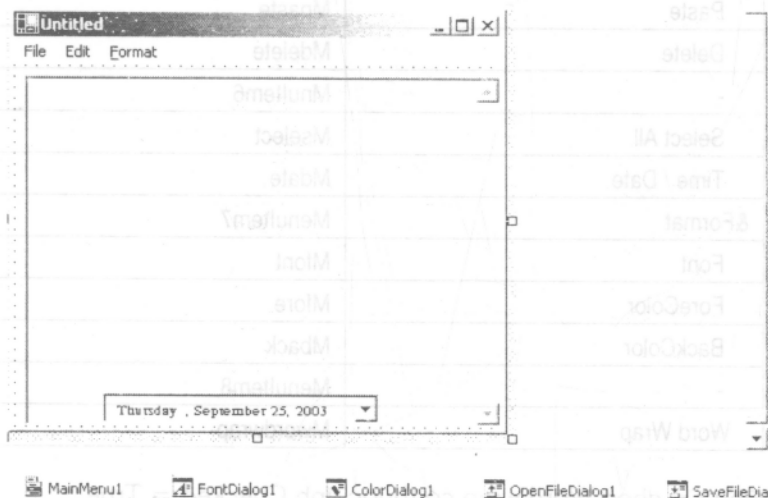
MinDate = 1/1/1753

MaxDate = 31/12/9998

Visible = false

Font = VNI-Times, 8, Bold

5g. Định lại vị trí các điều khiển trên form để được kết quả thiết kế form như Hình 7.11.



Hình 7.11 Màn hình thiết kế form Notepad.

5h. Viết mã lệnh xử lý cho form Notepad (không tính phần mã lệnh do VB.NET tự phát sinh trong quá trình thiết kế).

Public Class Notepad

Inherits System.Windows.Forms.Form

Dim sfilename

#Region

Private components As System.ComponentModel.Container

Private WithEvents TextBox As TextBox

Private Msg As SmartMessage

Public Sub New(ByRef Msg As SmartMessage)

MyBase.New()

Me.Msg = Msg

InitializeComponent()

TextBox.Text = Msg.MsgText()

End Sub

' Phần mã lệnh do VB.NET tự phát sinh trong quá trình thiết kế.

...

#End Region

' Khi chọn lệnh Format ➡ Font, hộp thoại Font sẽ hiển thị cho phép

' người dùng chọn font cho văn bản hiển thị trong Text box

Private Sub MFont_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles mfont.Click

FontDialog1.ShowDialog() *' Hiển thị hộp thoại Font*

' Lấy được tính font vừa chọn cho văn bản trong TextBox

TextBox.Font = FontDialog1.Font

End Sub

' Khi chọn lệnh File ➡ Exit, hiển thị hộp thông báo (Hình 7.12)

' Chương trình sẽ kết thúc khi người dùng chọn Yes

Private Sub MExit_Click(ByVal sender As System.Object, ByVal e As

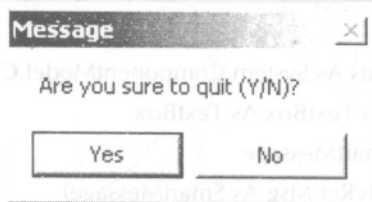
System.EventArgs) Handles MExit.Click

If MsgBox("Are you sure to quit (Y/N)?", MsgBoxStyle.YesNo,
"Message") = MsgBoxResult.Yes Then

End

End If

End Sub



Hình 7.12 Hộp thông báo hỏi ý người dùng.

- ' Khi chọn lệnh **Format** ➔ **ForeColor**, hộp thoại **Color** sẽ hiển thị cho phép người dùng chọn màu cho văn bản hiển thị trong **Text box**

```
Private Sub MFore_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MFore.Click
    ColorDialog1.ShowDialog()      ' Hiển thị hộp thoại Color
    ' Đặt màu được chọn làm màu chữ trong TextBox
    TextBox.ForeColor = ColorDialog1.Color
```

End Sub

- ' Khi chọn lệnh **Format** ➔ **BackColor**, hộp thoại **Color** sẽ hiển thị cho phép người dùng chọn màu nền **Text box**

```
Private Sub Mback_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Mback.Click
    ColorDialog1.ShowDialog()      ' Hiển thị hộp thoại Color
    ' Đặt màu được chọn làm màu nền TextBox
    TextBox.BackColor = ColorDialog1.Color
```

End Sub

- ' Khi chọn lệnh **File** ➔ **Open**, hộp thoại **Open file** sẽ hiển thị cho phép người dùng chọn tập tin văn bản cần mở

```
Private Sub MOpen_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MOpen.Click
    OpenFileDialog1.ShowDialog()    ' mở hộp thoại Open file
    sfilename = OpenFileDialog1.FileName() ' xác định tập tin sẽ mở
```

```
Notepad.ActiveForm.Text = sfilename      ' Hiển thị tên tập tin lên  
                                           tiêu đề form  
Msg.Read(sfilename)                     ' mở và đọc nội dung tập tin  
TextBox.Text = Msg.MsgText               ' gán nội dung tập tin ra Textbox  
TextBox.Refresh()                       ' Làm mới nội dung Textbox
```

End Sub

' Khi chọn lệnh File ➡ Save As, hộp thoại Save As sẽ hiển thị cho phép

' người dùng lưu nội dung tập tin văn bản đang mở với một tên mới

```
Private Sub MSaveAs_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MSaveAs.Click
```

```
SaveFileDialog1.ShowDialog()             ' mở hộp thoại Open file
```

```
sfilename = SaveFileDialog1.FileName()    ' xác định tên tập tin sẽ lưu
```

```
Msg.MsgText = TextBox.Text()             ' xác định nội dung sẽ lưu
```

```
Msg.Write(sfilename)                     ' Tạo / mở và lưu nội dung tập tin
```

```
Notepad.ActiveForm.Text = sfilename      ' Hiển thị tên tập tin lên  
                                           tiêu đề form
```

End Sub

' Khi chọn lệnh File ➡ New, trở về chế độ soạn tập tin mới

```
Private Sub MNew_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MNew.Click
```

```
TextBox.Text() = ""                      ' Xóa trống Textbox
```

```
sfilename = ""                           ' Tập tin mới chưa xác định tên
```

```
Notepad.ActiveForm.Text = "Untitled"
```

End Sub

' Khi chọn lệnh File ➡ Save, lưu nội dung hiện hành vào tập tin

```
Private Sub MSave_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MSave.Click
```

```
' Trong trường hợp soạn tập tin mới
```

```
If sfilename = "" Then
```

```

SaveFileDialog1.ShowDialog()           ' gọi hộp thoại Save As
sfilename = SaveFileDialog1.FileName() ' xác định tên tập tin mới
                                         cần tạo

Msg.MsgText = TextBox.Text()

Msg.Write(sfilename)                   ' tạo và lưu văn bản vào tập tin

Notepad.ActiveForm.Text = sfilename   ' hiển thị tên tập tin mới
                                         ra tiêu đề form

Else ' Trường hợp mở tập tin để điều chỉnh
    Msg.MsgText = TextBox.Text()
    Msg.Write(sfilename) ' Ghi nội dung mới vào tập tin
End If

End Sub

' Khi chọn lệnh Format ➡ Wordwrap, đổi chế độ cuộn thành không cuộn
' văn bản trong Textbox hoặc ngược lại
Private Sub Mwordwrap_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Mwordwrap.Click
    Mwordwrap.Checked = Not Mwordwrap.Checked
    TextBox.WordWrap = Mwordwrap.Checked
    ' Nếu ở chế độ cuộn, thì ẩn thanh cuộn ngang
    If TextBox.WordWrap = True Then
        TextBox.ScrollBars = ScrollBars.Vertical
    Else ' Nếu ở chế độ không cuộn, thì hiển thị cả thanh cuộn ngang
        TextBox.ScrollBars = ScrollBars.Both
    End If
End Sub

End Sub

' Khi chọn lệnh Edit ➡ Copy, sao chép phần văn bản đang đánh khối
' trong Textbox vào Clipboard
Private Sub MCopy_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MCopy.Click

```

```
TextBox.Copy()
```

```
End Sub
```

** Khi chọn lệnh Edit ➡ Paste, dán nội dung Clipboard vào Textbox*

```
Private Sub MPaste_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MPaste.Click
```

```
    TextBox.Paste()
```

```
End Sub
```

** Khi chọn lệnh Edit ➡ Cut, cắt phần văn bản đang đánh khối*

** trong Textbox vào Clipboard*

```
Private Sub MCut_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MCut.Click
```

```
    TextBox.Cut()
```

```
End Sub
```

** Khi chọn lệnh Edit ➡ Undo, hủy bỏ thao tác lệnh chỉnh sửa vừa mới*

** thực hiện*

```
Private Sub MUndo_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MUndo.Click
```

```
    TextBox.Undo()
```

```
End Sub
```

** Khi chọn lệnh Edit ➡ Delete, xóa toàn bộ nội dung Textbox*

```
Private Sub MDelete_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MDelete.Click
```

```
    TextBox.Clear()
```

```
End Sub
```

** Khi chọn lệnh Edit ➡ Select All, chọn toàn bộ nội dung Textbox*

```
Private Sub MSelect_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles Mselect.Click
```

```
    TextBox.SelectAll()
```

```
End Sub
```

‘ Khi chọn lệnh Edit ➡ Time/Date, dán ngày giờ hệ thống hiện hành tới Textbox

```
Private Sub MDate_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles MDate.Click
```

```
    TextBox.Text = TextBox.Text + DateTimePicker1.Value.ToString
```

```
End Sub
```

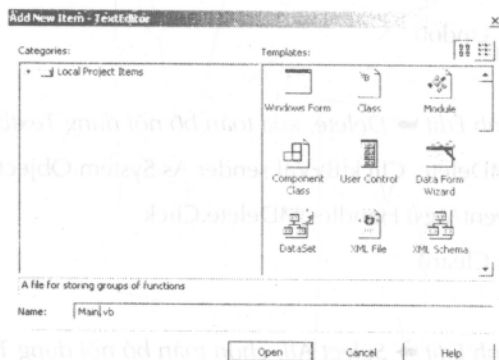
MÁCH NƯỚC

- ◆ Msg là một đối tượng thuộc lớp SmartMessage.
- ◆ Lớp Notepad kế thừa lớp System.Windows.Forms.Form.

Bước 6. Bổ sung Sub Main.

Do phương thức New định nghĩa trong form Notepad (lớp Notepad) có lấy tham số (Msg), nên bạn cần bắt đầu trình ứng dụng từ module Main, thay vì bắt đầu từ form chính Notepad, như các trường hợp thông thường.

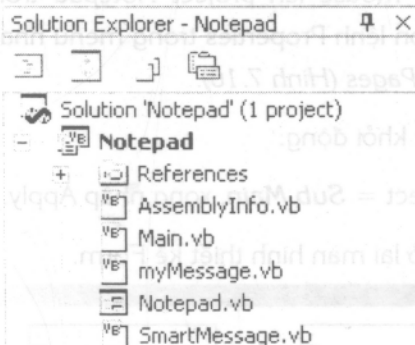
- Để thêm một module mới vào project, chọn lệnh Project ➡ Add Module kích hoạt của sổ Add New Item – Notepad (Hình 7.13).



Hình 7.13 Cửa sổ Add New Item – Notepad.

Chọn Templates = Windows Form
đặt Name = Main.vb
Xong nhấp Open.

Kết quả của sổ Solution Explorer lúc này sẽ như Hình 7.14.



Hình 7.14 Cửa sổ Solution Explorer.

- Định nghĩa đoạn mã sau cho module Main trong cửa sổ Code:

Module Main

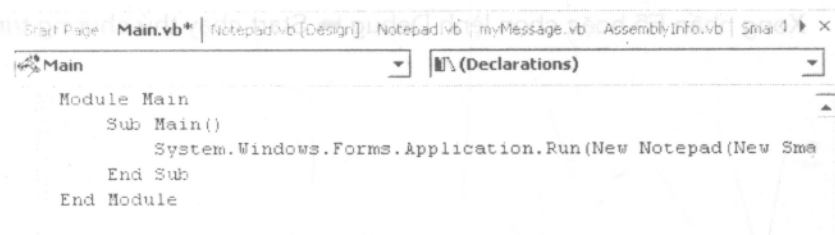
Sub Main()

System.Windows.Forms.Application.Run(New Notepad(New
SmartMessage()))

End Sub

End Module

- Kết quả của sổ Code của module Main sẽ như Hình 7.15.



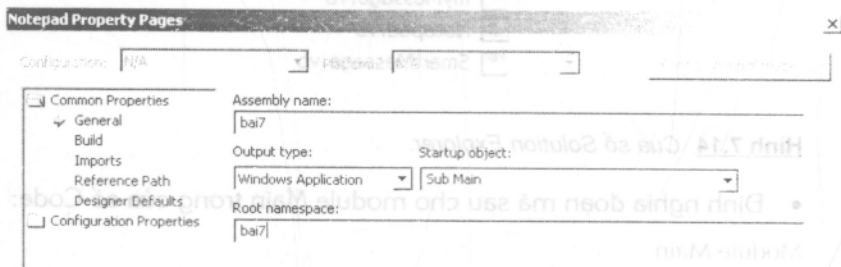
Hình 7.15 Cửa sổ Code của module Main.

BƯỚC 7. Xác định điểm khởi đầu cho trình ứng dụng.

Nhấp nút phải mouse lên project *Notepad* trong cửa sổ Solution Explorer, xong chọn lệnh Properties trong menu nhanh để gọi hộp thoại *Notepad Property Pages* (Hình 7.16).

Định lại thủ tục khởi động:

- Startup Object = **Sub Main**, xong nhấp Apply.
- Nhấp OK trở lại màn hình thiết kế Form.



Hình 7.16 Màn hình *Notepad Property Pages*.

BƯỚC 8. Biên dịch và chạy thử chương trình.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Xong nhấn F5 hoặc chọn lệnh Debug ➡ Start chạy thử chương trình.



Bài 8.

Chương trình cập nhật phòng ban

Mục đích chương trình

Bài thực hành này làm quen với việc thiết kế form cập nhật, điều chỉnh nội dung một table trong cơ sở dữ liệu Access với VB.NET. Từ đó, rút ra được những điểm tiện lợi so với kỹ thuật lập trình cơ sở dữ liệu đã sử dụng trong VB6.

Hoạt động chương trình

Khi thi hành chương trình, bạn sẽ được màn hình có giao diện như Hình 8.1. Chương trình sẽ tự động mở màn hình xem và cập nhật danh mục phòng ban.

FPhongban [Minimize] [Maximize] [Close]

Mã phòng ban: G0001 Update

Tên phòng ban: Phòng Nhân Sự Cancel/All

<< < 1 of 8 > >>

Load Add Delete Cancel

Hình 8.1 Giao diện màn hình khi form Fphongban được nạp.

Người dùng có thể tiến hành điều chỉnh mã phòng và tên phòng ban của record hiện hành. Cũng như có thể sử dụng các button <<, <, >, >> để xem nội dung các record trong table Phongban.

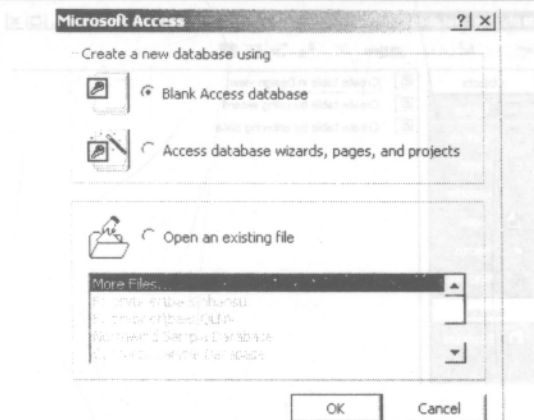
Sau đây là các nút chức năng xử lý chính của form:

- <<: chuyển đến record đầu tiên
- < : chuyển đến record trước đó
- > : chuyển đến record kế tiếp
- >>: chuyển đến record cuối cùng
- Load: Nạp nội dung table Phongban cho điều khiển dữ liệu DataSet.
- Add: Thêm record mới.
- Delete: Xóa record hiện hành.
- Cancel: Hủy bỏ thao tác chỉnh sửa mới nhất, khi chưa thực hiện lệnh cập nhật vào table Phongban.
- Update: Cập nhật các chỉnh sửa, thêm, xóa vào table Phongban.
- Cancel All: Hủy bỏ tất cả các lệnh thêm, xóa, chỉnh sửa nội dung các record đã thực hiện, nhưng chưa được cập nhật vào table Phongban với lệnh Update.

Hướng dẫn thiết kế

Bước 1. Thiết kế cơ sở dữ liệu nhansu.mdb với Microsoft Access 2000.

Khởi động Access 2000, chọn tùy chọn Blank Access Database (Hình 8.2), xong nhấp OK.



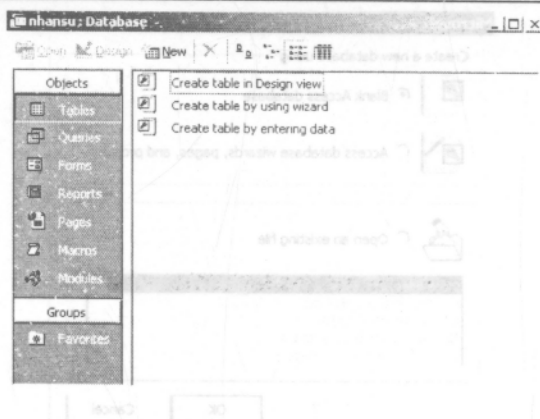
Hình 8.2 Cửa sổ Microsoft Access khi vừa khởi động.

Trong cửa sổ File New Database, nhập tên cơ sở dữ liệu mới cần tạo là Nhansu.mdb trong mục **File name**, chọn thư mục lưu trữ là C:\BTBNET trong menu popup **Save in** (Hình 8.3), xong nhấn Create.



Hình 8.3

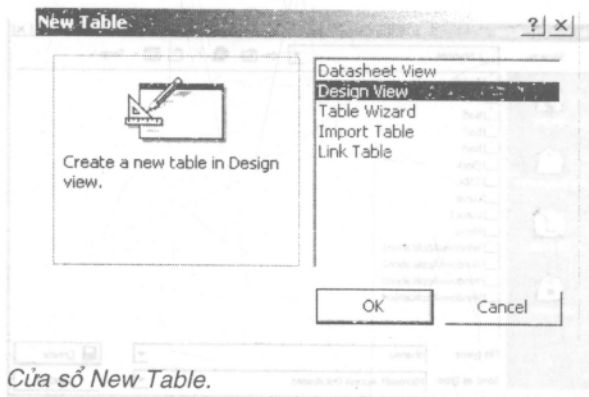
Một cơ sở dữ liệu Access trống được tạo như Hình 8.4.



Hình 8.4 Cửa sổ cơ sở dữ liệu Nhansu.mdb mới tạo.

Định nghĩa cấu trúc table DMPhongban:

- Chọn Tab Tables, xong nhấn nút New. Trong cửa sổ New Table (Hình 8.5), chọn Design View, xong nhấn OK.

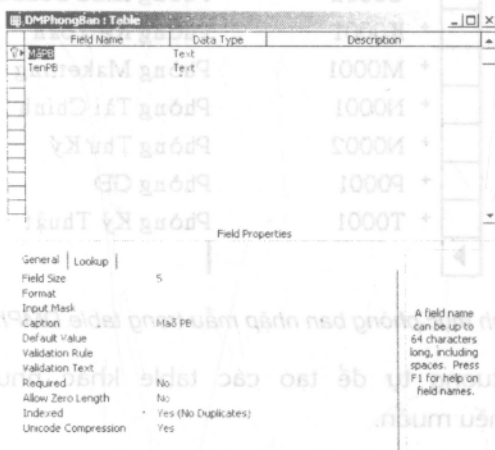


Hình 8.5 Cửa sổ New Table.

- Khai báo tên, kiểu dữ liệu và kích thước các field trong table DMPhongban như sau:

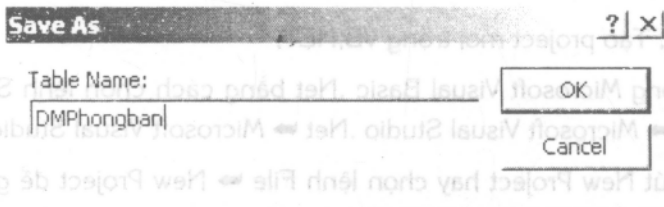
Fieldname	FieldType	FieldSize
Mapb	Text	5
Tenpb	Text	50

Đặt khóa chính cho field **Mapb** bằng cách chọn field Mapb, xong nhấp biểu tượng khóa (Primary key) trên Toolbar Standard (Hình 8.6).



Hình 8.6 Định nghĩa các field cho table mới.

Nhấp nút Close ở góc phải của sổ hoặc nhấn Ctrl+W để đóng table mới, lưu và đặt tên cho table mới trong cửa sổ Save As là DMPhongban (Hình 8.7).



Hình 8.7 Đặt tên table mới định nghĩa cấu trúc.

Nhấp đúp mouse vào table DMPhongban để mở và nhập tay vào dữ liệu mẫu như Hình 8.8.

Xong đóng cơ sở dữ liệu Nhansu.mdb và thoát Access.

DMPhongBan : Table		
	Mã PB	Tên PB
+ G0001		Phòng Nhân Sự
+ G0002		Phòng Kinh Doanh
+ K0001		Phòng Kế Toán
+ M0001		Phòng Marketing
+ N0001		Phòng Tài Chính
+ N0002		Phòng Thư Ký
+ P0001		Phòng GD
+ T0001		Phòng Kỹ Thuật

Hình 8.8 Danh mục phòng ban nhập mẫu trong table DMPhongban.

Tiến hành tương tự để tạo các table khác như DMNhanvien, DMGiadinh, ... nếu muốn.

GHI CHÚ

- ♦ Bạn cũng có thể sử dụng Microsoft Access 97 để thiết kế cơ sở dữ liệu nhansu.mdb cho bài thực hành này, mà không phải thay đổi các bước tiến hành nào.

Bước 2. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net

Nhấp nút New Project hay chọn lệnh File ➡ New Project để gọi cửa sổ New Project

- Đặt tên tập tin lưu project: Name = *Bai8*
- Định thư mục lưu project: Location = *C:\BTVBNET*
- Nhấp tùy chọn **Close Solution**.
- Chọn Project types = *Visual Basic Projects*, xong nhấp biểu tượng *Windows Application* trong mục *Templates*.
- Xong nhấp OK.

VB.NET sẽ tự động khởi tạo 1 project mới với 1 form trống mặc định là Form1.vb chứa trong thư mục BA18, là thư mục con của C:\BTVBNET (Hình 8.9).



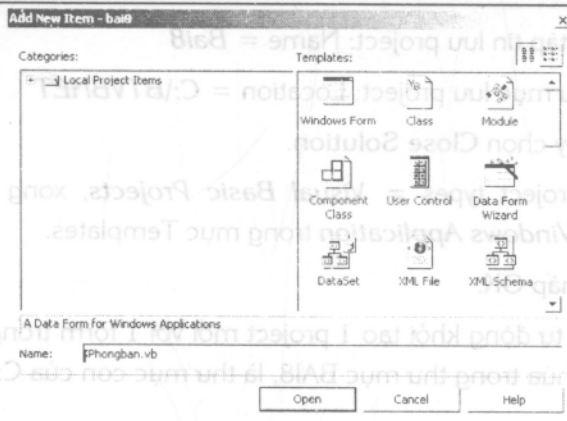
Hình 8.9 Cửa sổ Project Bai8 mới tạo.

Để tiện việc quản lý, giờ đây bạn hãy copy cơ sở dữ liệu Nhansu.mdb từ thư mục C:\BTVBNET vào C:\BTVBNET\BA18.

Bước 3. Tạo form FPhongban.vb dùng cập nhật dữ liệu table DMPhongban.

Chọn lệnh View ➡ Solution Explorer để bật cửa sổ Solution Explorer nếu chưa thấy.

Chọn lệnh Project ➡ Add Windows form Trong cửa sổ Add New Item (Hình 8.10), nhấp biểu tượng Data Form Wizard.



Hình 8.10 Hộp thoại Add New Item.

Nhập vào tên form Name = FPhongban.vb, xong nhấn Open.

Nhấp Next trong cửa sổ Data Form Wizard.

Nhập tên điều khiển Dataset mới là phongban với tùy chọn Create a new dataset named, xong nhấn Next (Hình 8.11).



Hình 8.11 Cửa sổ Data Form Wizard.

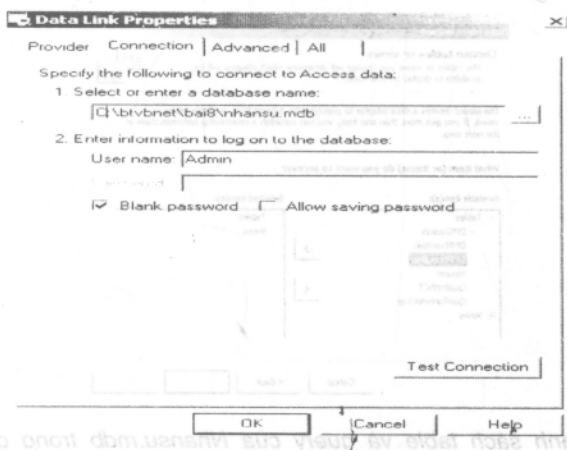
Chọn cơ sở dữ liệu Access cần kết nối dữ liệu bằng cách nhấp nút New Connection..., để gọi cửa sổ Data Link Properties.

Chọn tab Provider, xong chọn Microsoft Jet 4.0 OLE DB Provider (Hình 8.12).



Hình 8.12 Cửa sổ Data Link Properties – trang Provider.

Chọn tab Connection, nhấp nút ... ở phía sau text box Select or enter a database name để chọn cơ sở dữ liệu kết nối (Hình 8.13). Xong nhấp Test Connection, nếu thấy thông báo Test connection succeeded, xem như bạn đã thành công. Nhấp OK để trở lại cửa sổ Data Form Wizard (Hình 8.14), xong nhấp Next.



Hình 8.13 Cửa sổ Data Link Properties – trang Connection.

MÁCH NƯỚC

- ◆ Nếu bạn sử dụng cơ sở dữ liệu Access 97, thì có thể chọn Microsoft Jet 3.51 OLE DB Provider cũng được.
- ◆ Kết nối không thành công thường do lỗi tên cơ sở dữ liệu kết nối không được tìm thấy, hoặc phiên bản cơ sở dữ liệu không phù hợp với Provider bạn đã chọn.



Hình 8.14 Cửa sổ Data Form Wizard sau khi chọn cơ sở dữ liệu kết nối.

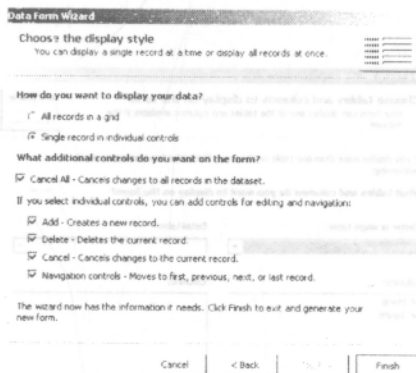
Danh sách các table và query trong cơ sở dữ liệu Nhansu.mdb sẽ được liệt kê trong cửa sổ Data Form Wizard (Hình 8.15).



Hình 8.15 Danh sách table và query của Nhansu.mdb trong cửa sổ Data Form Wizard.

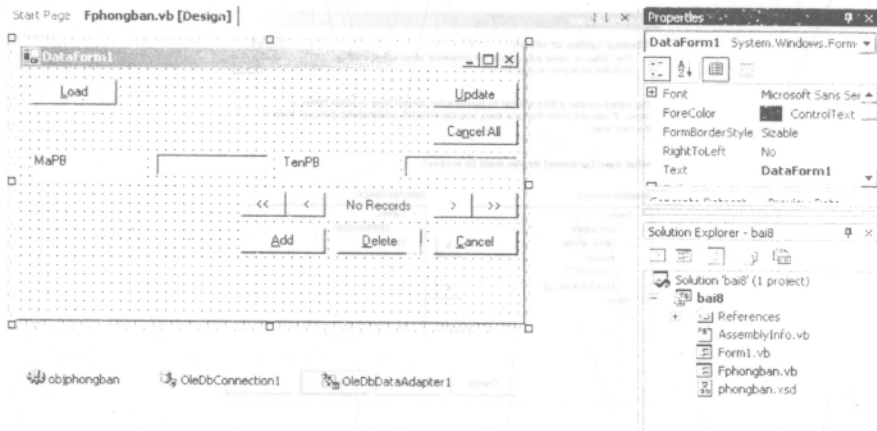
- Single record in individual controls (Hình 8.18). Xong nhấn Finish.

MPHongban



Hình 8.18 Chọn kiểu trình bày form

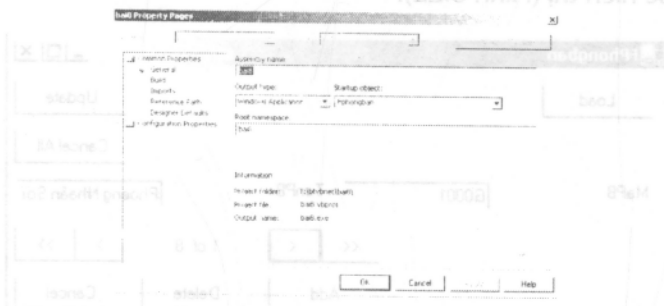
Kết quả của sổ thiết kế form Fphongban sẽ như Hình 8.19.



Hình 8.19 Cửa sổ thiết kế form Fphongban.

Loại bỏ form trống (Form1.vb) khỏi project bằng cách nhấp nút phải mouse lên form trong cửa sổ Solution Explorer, xong chọn lệnh Delete.

Đặt form Fphongban làm form khởi động bằng cách nhấp phải tên project bai8 trong cửa sổ Solution Explorer, chọn lệnh Properties, định Startup Object = Fphongban (Hình 8.20).



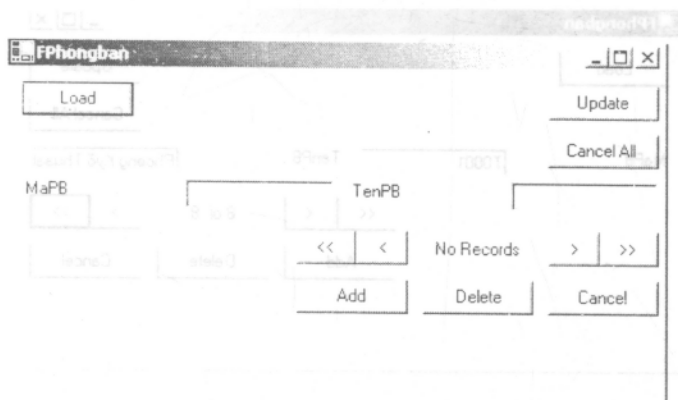
Hình 8.20 Cửa sổ Bai8 Property Pages.

Bước 4. Chạy thử chương trình để xem kết quả.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start để chạy thử chương trình. Bạn sẽ được màn hình form như Hình 8.21.



Hình 8.21 Màn hình form Fphongban khi vừa thi hành.

Điều ngạc nhiên đầu tiên là bạn đã nhập danh sách 8 phòng ban vào table DMPhongban rồi, mà nay chương trình báo không có record nào là sao? Để giải đáp thắc mắc, bạn nhấp nút Load, lập tức phòng ban đầu tiên sẽ hiển thị (Hình 8.22).

The screenshot shows a Windows form titled "FPhongban". It has a "Load" button on the left and "Update", "Cancel All", "Add", "Delete", and "Cancel" buttons on the right. The form displays the first record with "MaPB" as "G0001", "TenPB" as "Phong Nhân Sự", and "1 of 8" records shown. Navigation buttons include "<<", "<", ">", ">>", "Add", "Delete", and "Cancel".

Hình 8.22 Màn hình form FPhongban sau khi nhấp nút Load.

Nhấp nút >>, record cuối cùng sẽ hiển thị như Hình 8.23. Lần lượt thử các nút còn lại, tất cả đều hoạt động tuyệt vời như dự kiến.

The screenshot shows the same "FPhongban" form, but now displaying the last record with "MaPB" as "T0001", "TenPB" as "Phong Kỹ Thuật", and "8 of 8" records shown. The navigation buttons remain the same.

Hình 8.23 Màn hình form FPhongban sau khi nhấp nút >>.

Một điểm thú vị khác của form FPhongban vừa tạo là sau khi nhấp Delete để xóa liên tục 4 record, sau đó bạn đổi ý nhấp Cancel All, cả 4 record bị xóa sẽ được phục hồi. Thực chất, VB.Net đã giúp bạn thực hiện những công việc tuyệt vời đó với những mã lệnh tự phát sinh trong quá trình thiết kế form của bạn.

Bước 5. Ý nghĩa đoạn mã lệnh trong form Fphongban do VB.Net tự phát sinh trong quá trình thiết kế.

Lưu ý: Bạn nên nắm ý nghĩa các mã lệnh xử lý ứng với các button trên form. Điều này sẽ giúp bạn thuận lợi hơn cho việc điều chỉnh form về sau.

Nhấp đúp lên button Load để mở cửa sổ Code của form FPhongban.

```
Public Class Fphongban
```

```
    Inherits System.Windows.Forms.Form
```

' Đoạn mã lệnh định nghĩa các điều khiển sử dụng trên form

```
#Region " Windows Form Designer generated code "
```

```
    Public Sub New()
```

```
        MyBase.New()
```

```
        InitializeComponent()
```

```
    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
```

```
Private components As System.ComponentModel.IContainer
Friend WithEvents OleDbSelectCommand1 As
    System.Data.OleDb.OleDbCommand
Friend WithEvents OleDbInsertCommand1 As
    System.Data.OleDb.OleDbCommand
Friend WithEvents OleDbUpdateCommand1 As
    System.Data.OleDb.OleDbCommand
Friend WithEvents OleDbDeleteCommand1 As
    System.Data.OleDb.OleDbCommand
Friend WithEvents objphongban As bai8.phongban
Friend WithEvents OleDbConnection1 As
    System.Data.OleDb.OleDbConnection
Friend WithEvents OleDbDataAdapter1 As
    System.Data.OleDb.OleDbDataAdapter
Friend WithEvents btnLoad As System.Windows.Forms.Button
Friend WithEvents btnUpdate As System.Windows.Forms.Button
Friend WithEvents btnCancelAll As System.Windows.Forms.Button
Friend WithEvents lblMaPB As System.Windows.Forms.Label
Friend WithEvents editMaPB As System.Windows.Forms.TextBox
Friend WithEvents lblTenPB As System.Windows.Forms.Label
Friend WithEvents editTenPB As System.Windows.Forms.TextBox
Friend WithEvents btnNavFirst As System.Windows.Forms.Button
Friend WithEvents btnNavPrev As System.Windows.Forms.Button
Friend WithEvents lblNavLocation As System.Windows.Forms.Label
Friend WithEvents btnNavNext As System.Windows.Forms.Button
Friend WithEvents btnLast As System.Windows.Forms.Button
```



```
Friend WithEvents btnAdd As System.Windows.Forms.Button
Friend WithEvents btnDelete As System.Windows.Forms.Button
Friend WithEvents btnCancel As System.Windows.Forms.Button
<System.Diagnostics.DebuggerStepThrough> Private Sub
    InitializeComponent()
        Me.OleDbSelectCommand1 = New
            System.Data.OleDb.OleDbCommand()
        Me.OleDbConnection1 = New
            System.Data.OleDb.OleDbConnection()
        Me.OleDbInsertCommand1 = New
            System.Data.OleDb.OleDbCommand()
        Me.OleDbUpdateCommand1 = New
            System.Data.OleDb.OleDbCommand()
        Me.OleDbDeleteCommand1 = New
            System.Data.OleDb.OleDbCommand()
        Me.objphongban = New bai8.phongban()
        Me.OleDbDataAdapter1 = New
            System.Data.OleDb.OleDbDataAdapter()
        Me.btnLoad = New System.Windows.Forms.Button()
        Me.btnUpdate = New System.Windows.Forms.Button()
        Me.btnCancelAll = New System.Windows.Forms.Button()
        Me.lblMaPB = New System.Windows.Forms.Label()
        Me.editMaPB = New System.Windows.Forms.TextBox()
        Me.lblTenPB = New System.Windows.Forms.Label()
        Me.editTenPB = New System.Windows.Forms.TextBox()
        Me.btnNavFirst = New System.Windows.Forms.Button()
```

```

Me.btnNavPrev = New System.Windows.Forms.Button()
Me.lblNavLocation = New System.Windows.Forms.Label()
Me.btnNavNext = New System.Windows.Forms.Button()
Me.btnLast = New System.Windows.Forms.Button()
Me.btnAdd = New System.Windows.Forms.Button()
Me.btnDelete = New System.Windows.Forms.Button()
Me.btnCancel = New System.Windows.Forms.Button()
' đoạn mã lệnh định nghĩa các điều khiển kết nối dữ liệu
CType(Me.objphongban,
    System.ComponentModel.ISupportInitialize).BeginInit()
Me.SuspendLayout()
Me.OleDbSelectCommand1.CommandText = "SELECT
    MaPB, TenPB FROM DMPhongBan"
Me.OleDbSelectCommand1.Connection = Me.OleDbConnection1
Me.OleDbConnection1.ConnectionString =
    "Provider=Microsoft.Jet.OLEDB.4.0;Password='''''';
    User ID=Admin;Data Source=C:\btvbnet" & _
    "\bai8\nhansu.mdb;Mode=Share Deny None;Extended
    Properties='''''';Jet OLEDB:System da" & _
    "tabase='''''';Jet OLEDB:Registry Path='''''';Jet OLEDB:Database
    Password='''''';Jet OLEDB:En" & _
    "gine Type=5;Jet OLEDB:Database Locking Mode=1;
    Jet OLEDB:Global Partial Bulk Ops=" & _
    "2;Jet OLEDB:Global Bulk Transactions=1;Jet OLEDB:New Database
    Password='''''';Jet OL" & _
    "EDB>Create System Database=False;Jet OLEDB:Encrypt
    Database=False;Jet OLEDB:Don" & _

```

```
"t Copy Locale on Compact=False;Jet OLEDB:Compact Without  
Replica Repair=False;Je" & _  
"t OLEDB:SFP=False"
```

```
Me.OleDbInsertCommand1.CommandText = "INSERT INTO  
DMPhongBan(MaPB, TenPB) VALUES (?, ?)"
```

```
Me.OleDbInsertCommand1.Connection = Me.OleDbConnection1
```

```
Me.OleDbInsertCommand1.Parameters.Add(New  
System.Data.OleDb.OleDbParameter("MaPB",  
System.Data.OleDb.OleDbType.VarWChar, 5, "MaPB"))
```

```
Me.OleDbInsertCommand1.Parameters.Add(New  
System.Data.OleDb.OleDbParameter("TenPB",  
System.Data.OleDb.OleDbType.VarWChar, 50, "TenPB"))
```

```
Me.OleDbUpdateCommand1.CommandText = "UPDATE  
DMPhongBan SET MaPB = ?, TenPB = ? WHERE (MaPB = ?)  
AND (TenPB = ? OR ? IS" & _  
" NULL AND TenPB IS NULL)"
```

```
Me.OleDbUpdateCommand1.Connection = Me.OleDbConnection1
```

```
Me.OleDbUpdateCommand1.Parameters.Add(New  
System.Data.OleDb.OleDbParameter("MaPB",  
System.Data.OleDb.OleDbType.VarWChar, 5, "MaPB"))
```

```
Me.OleDbUpdateCommand1.Parameters.Add(New  
System.Data.OleDb.OleDbParameter("TenPB",  
System.Data.OleDb.OleDbType.VarWChar, 50, "TenPB"))
```

```
Me.OleDbUpdateCommand1.Parameters.Add(New  
System.Data.OleDb.OleDbParameter("Original_MaPB",  
System.Data.OleDb.OleDbType.VarWChar, 5,  
System.Data.ParameterDirection.Input, False, CType(0, Byte),  
CType(0, Byte), "MaPB", System.Data.DataRowVersion.Original,  
Nothing))
```

```
Me.OleDbUpdateCommand1.Parameters.Add(New
    System.Data.OleDb.OleDbParameter("Original_TenPB",
    System.Data.OleDb.OleDbType.VarWChar, 50,
    System.Data.ParameterDirection.Input, False, CType(0, Byte),
    CType(0, Byte), "TenPB", System.Data.DataRowVersion.Original,
    Nothing))

Me.OleDbUpdateCommand1.Parameters.Add(New
    System.Data.OleDb.OleDbParameter("Original_TenPB1",
    System.Data.OleDb.OleDbType.VarWChar, 50,
    System.Data.ParameterDirection.Input, False, CType(0, Byte),
    CType(0, Byte), "TenPB", System.Data.DataRowVersion.Original,
    Nothing))

Me.OleDbDeleteCommand1.CommandText = "DELETE FROM
    DMPhongBan WHERE (MaPB = ?) AND (TenPB = ? OR ? IS NULL
    AND TenPB IS " & _"NULL)"

Me.OleDbDeleteCommand1.Connection = Me.OleDbConnection1

Me.OleDbDeleteCommand1.Parameters.Add(New
    System.Data.OleDb.OleDbParameter("Original_MaPB",
    System.Data.OleDb.OleDbType.VarWChar, 5,
    System.Data.ParameterDirection.Input, False, CType(0, Byte),
    CType(0, Byte), "MaPB", System.Data.DataRowVersion.Original,
    Nothing))

Me.OleDbDeleteCommand1.Parameters.Add(New
    System.Data.OleDb.OleDbParameter("Original_TenPB",
    System.Data.OleDb.OleDbType.VarWChar, 50,
    System.Data.ParameterDirection.Input, False, CType(0, Byte),
    CType(0, Byte), "TenPB", System.Data.DataRowVersion.Original,
    Nothing))

Me.OleDbDeleteCommand1.Parameters.Add(New
```

```
System.Data.OleDb.OleDbParameter("Original_TenPB1",  
System.Data.OleDb.OleDbType.VarWChar, 50,  
System.Data.ParameterDirection.Input, False, CType(0, Byte),  
CType(0, Byte), "TenPB", System.Data.DataRowVersion.Original,  
Nothing))
```

```
Me.objphongban.DataSetName = "phongban"  
Me.objphongban.Locale = New System.Globalization.CultureInfo("en-US")  
Me.objphongban.Namespace = "http://www.tempuri.org/phongban.xsd"  
Me.OleDbDataAdapter1.DeleteCommand = Me.OleDbDeleteCommand1  
Me.OleDbDataAdapter1.InsertCommand = Me.OleDbInsertCommand1  
Me.OleDbDataAdapter1.SelectCommand = Me.OleDbSelectCommand1  
Me.OleDbDataAdapter1.TableMappings.AddRange(New  
System.Data.Common.DataTableMapping() {New  
System.Data.Common.DataTableMapping("Table", "DMPhongBan",  
New System.Data.Common.DataColumnMapping() {New  
System.Data.Common.DataColumnMapping("MaPB", "MaPB"), New  
System.Data.Common.DataColumnMapping("TenPB", "TenPB")}}})  
Me.OleDbDataAdapter1.UpdateCommand =  
Me.OleDbUpdateCommand1
```

' đoạn mã lệnh định nghĩa các button

```
Me.btnLoad.Location = New System.Drawing.Point(10, 10)  
Me.btnLoad.Name = "btnLoad"  
Me.btnLoad.TabIndex = 0  
Me.btnLoad.Text = "&Load"  
Me.btnUpdate.Location = New System.Drawing.Point(365, 10)  
Me.btnUpdate.Name = "btnUpdate"
```

```
Me.btnUpdate.TabIndex = 1
Me.btnUpdate.Text = "&Update"
Me.btnCancelAll.Location = New System.Drawing.Point(365, 43)
Me.btnCancelAll.Name = "btnCancelAll"
Me.btnCancelAll.TabIndex = 2
Me.btnCancelAll.Text = "Ca&ncel All"
Me.lblMaPB.Location = New System.Drawing.Point(10, 76)
Me.lblMaPB.Name = "lblMaPB"
Me.lblMaPB.TabIndex = 3
Me.lblMaPB.Text = "MaPB"
' đoạn mã lệnh định nghĩa các Textbox và Label
Me.editMaPB.DataBindings.Add(New
    System.Windows.Forms.Binding("Text", Me.objphongban,
        "DMPhongBan.MaPB"))
Me.editMaPB.Location = New System.Drawing.Point(120, 76)
Me.editMaPB.Name = "editMaPB"
Me.editMaPB.TabIndex = 4
Me.editMaPB.Text = ""
Me.lblTenPB.Location = New System.Drawing.Point(230, 76)
Me.lblTenPB.Name = "lblTenPB"
Me.lblTenPB.TabIndex = 5
Me.lblTenPB.Text = "TenPB"
Me.editTenPB.DataBindings.Add(New
    System.Windows.Forms.Binding("Text", Me.objphongban,
        "DMPhongBan.TenPB"))
Me.editTenPB.Location = New System.Drawing.Point(340, 76)
```

```
Me.editTenPB.Name = "editTenPB"
```

```
Me.editTenPB.TabIndex = 6
```

```
Me.editTenPB.Text = ""
```

' đoạn mã lệnh định nghĩa các button

```
Me.btnNavFirst.Location = New System.Drawing.Point(195, 109)
```

```
Me.btnNavFirst.Name = "btnNavFirst"
```

```
Me.btnNavFirst.Size = New System.Drawing.Size(40, 23)
```

```
Me.btnNavFirst.TabIndex = 7
```

```
Me.btnNavFirst.Text = "<<"
```

```
Me.btnNavPrev.Location = New System.Drawing.Point(235, 109)
```

```
Me.btnNavPrev.Name = "btnNavPrev"
```

```
Me.btnNavPrev.Size = New System.Drawing.Size(35, 23)
```

```
Me.btnNavPrev.TabIndex = 8
```

```
Me.btnNavPrev.Text = "<"
```

```
Me.lblNavLocation.BackColor = System.Drawing.Color.White
```

```
Me.lblNavLocation.Location = New System.Drawing.Point(270, 109)
```

```
Me.lblNavLocation.Name = "lblNavLocation"
```

```
Me.lblNavLocation.Size = New System.Drawing.Size(95, 23)
```

```
Me.lblNavLocation.TabIndex = 9
```

```
Me.lblNavLocation.Text = "No Records"
```

```
Me.lblNavLocation.TextAlign =
```

```
System.Drawing.ContentAlignment.MiddleCenter
```

```
Me.btnNavNext.Location = New System.Drawing.Point(365, 109)
```

```
Me.btnNavNext.Name = "btnNavNext"
```

```
Me.btnNavNext.Size = New System.Drawing.Size(35, 23)
```

```
Me.btnNavNext.TabIndex = 10
```

```
Me.btnNavNext.Text = ">"
Me.btnLast.Location = New System.Drawing.Point(400, 109)
Me.btnLast.Name = "btnLast"
Me.btnLast.Size = New System.Drawing.Size(40, 23)
Me.btnLast.TabIndex = 11
Me.btnLast.Text = ">>"
Me.btnAdd.Location = New System.Drawing.Point(195, 142)
Me.btnAdd.Name = "btnAdd"
Me.btnAdd.TabIndex = 12
Me.btnAdd.Text = "&Add"
Me.btnDelete.Location = New System.Drawing.Point(280, 142)
Me.btnDelete.Name = "btnDelete"
Me.btnDelete.TabIndex = 13
Me.btnDelete.Text = "&Delete"
Me.btnCancel.Location = New System.Drawing.Point(365, 142)
Me.btnCancel.Name = "btnCancel"
Me.btnCancel.TabIndex = 14
Me.btnCancel.Text = "&Cancel"
' đoạn mã lệnh định nghĩa form FPhongban
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(442, 223)
Me.Controls.AddRange(New System.Windows.Forms.Control()
    {Me.btnLoad, Me.btnUpdate, Me.btnCancelAll, Me.lblMaPB,
    Me.editMaPB, Me.lblTenPB, Me.editTenPB, Me.btnNavFirst,
    Me.btnNavPrev, Me.lblNavLocation, Me.btnNavNext, Me.btnLast,
    Me.btnAdd, Me.btnDelete, Me.btnCancel})
```



```

Me.Name = "Fphongban"
Me.Text = "FPhongban"
CType(Me.objphongban,
System.ComponentModel.ISupportInitialize).EndInit()
Me.ResumeLayout(False)

End Sub

#End Region

' Thủ tục xử lý nút Cancel
Private Sub btnCancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnCancel.Click
    Me.BindingContext(objphongban, "DMPhongBan").CancelCurrentEdit()
    Me.objphongban_PositionChanged()
End Sub

' Thủ tục xử lý nút Delete
Private Sub btnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnDelete.Click
    If (Me.BindingContext(objphongban, "DMPhongBan").Count > 0) Then
        Me.BindingContext(objphongban,
"DMPhongBan").RemoveAt(Me.BindingContext(objphongban,
"DMPhongBan").Position)
        Me.objphongban_PositionChanged()
    End If
End Sub

' Thủ tục xử lý nút Add
Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
    Try

```

```
Me.BindingContext(objphongban, "DMPhongBan").EndCurrentEdit()
Me.BindingContext(objphongban, "DMPhongBan").AddNew()
Catch eEndEdit As System.Exception
    System.Windows.Forms.MessageBox.Show(eEndEdit.Message)
End Try
Me.objphongban_PositionChanged()
End Sub

' Thủ tục xử lý nút Update
Private Sub btnUpdate_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnUpdate.Click
    Try
        Me.UpdateDataSet()
    Catch eUpdate As System.Exception
        System.Windows.Forms.MessageBox.Show(eUpdate.Message)
    End Try
    Me.objphongban_PositionChanged()
End Sub

' Thủ tục xử lý nút Load
Private Sub btnLoad_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles btnLoad.Click
    Try
        Me.LoadDataSet()
    Catch eLoad As System.Exception
        System.Windows.Forms.MessageBox.Show(eLoad.Message)
    End Try
    Me.objphongban_PositionChanged()
```

End Sub

' Thủ tục xử lý nút <<

Private Sub btnNavFirst_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnNavFirst.Click

Me.BindingContext(objphongban, "DMPhongBan").Position = 0

Me.objphongban_PositionChanged()

End Sub

' Thủ tục xử lý nút >>

Private Sub btnLast_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnLast.Click

Me.BindingContext(objphongban, "DMPhongBan").Position =

(Me.objphongban.Tables("DMPhongBan").Rows.Count - 1)

Me.objphongban_PositionChanged()

End Sub

' Thủ tục xử lý nút <

Private Sub btnNavPrev_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnNavPrev.Click

Me.BindingContext(objphongban, "DMPhongBan").Position =

(Me.BindingContext(objphongban, "DMPhongBan").Position - 1)

Me.objphongban_PositionChanged()

End Sub

' Thủ tục xử lý nút >

Private Sub btnNavNext_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnNavNext.Click

Me.BindingContext(objphongban, "DMPhongBan").Position =

(Me.BindingContext(objphongban, "DMPhongBan").Position + 1)

```
Me.objphongban_PositionChanged()
```

```
End Sub
```

```
' Thủ tục xác định record hiện hành đang được xử lý
```

```
Private Sub objphongban_PositionChanged()
```

```
Me.lblNavLocation.Text = (((Me.BindingContext(objphongban,
```

```
"DMPhongBan").Position + 1).ToString + " of ") _
```

```
+ Me.BindingContext(objphongban,
```

```
"DMPhongBan").Count.ToString)
```

```
End Sub
```

```
' Thủ tục xử lý nút Cancel All
```

```
Private Sub btnCancelAll_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles btnCancelAll.Click
```

```
Me.objphongban.RejectChanges()
```

```
End Sub
```

```
' Thủ tục cập nhật dữ liệu vào DataSet
```

```
Public Sub UpdateDataSet()
```

```
Dim objDataSetChanges As bai8.phongban = New bai8.phongban()
```

```
Me.BindingContext(objphongban, "DMPhongBan").EndCurrentEdit()
```

```
objDataSetChanges = CType(objphongban.GetChanges, bai8.phongban)
```

```
If (Not (objDataSetChanges) Is Nothing) Then
```

```
Try
```

```
Me.UpdateDataSource(objDataSetChanges)
```

```
objphongban.Merge(objDataSetChanges)
```

```
objphongban.AcceptChanges()
```

```
Catch eUpdate As System.Exception
```

```
Throw eUpdate
```

End Try

End If

End Sub

' Thủ tục nạp dữ liệu từ table DMPhongban lên form qua DataSet

Public Sub LoadDataSet()

Dim objDataSetTemp As bai8.phongban

objDataSetTemp = New bai8.phongban()

Try

Me.FillDataSet(objDataSetTemp)

Catch eFillDataSet As System.Exception

Throw eFillDataSet

End Try

Try

objphongban.Clear()

objphongban.Merge(objDataSetTemp)

Catch eLoadMerge As System.Exception

Throw eLoadMerge

End Try

End Sub

' Thủ tục cập nhật dữ liệu vào table DMPhongban từ DataSet

Public Sub UpdateDataSource(ByVal ChangedRows As bai8.phongban)

Try

If (Not (ChangedRows) Is Nothing) Then

Me.OleDbConnection1.Open()

OleDbDataAdapter1.Update(ChangedRows)

End If

```
Catch updateException As System.Exception
```

```
    Throw updateException
```

```
Finally
```

```
    Me.OleDbConnection1.Close()
```

```
End Try
```

```
End Sub
```

' Thủ tục kiểm tra dữ liệu trước khi cập nhật

```
Public Sub FillDataSet(ByVal dataSet As bai8.phongban)
```

```
    dataSet.EnforceConstraints = False
```

```
Try
```

```
    Me.OleDbConnection1.Open()
```

```
    Me.OleDbDataAdapter1.Fill(dataSet)
```

```
Catch fillException As System.Exception
```

```
    Throw fillException
```

```
Finally
```

```
    dataSet.EnforceConstraints = True
```

```
    Me.OleDbConnection1.Close()
```

```
End Try
```

```
End Sub
```

```
End Class
```

Bước 6. Điều chỉnh form FPhongban.vb vừa tạo.

Form Fphongban vừa tạo còn một số điểm bạn chưa vừa ý, chẳng hạn:

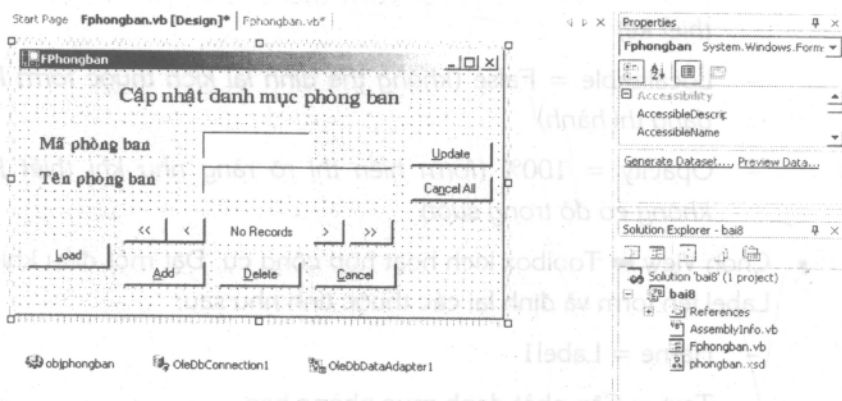
- Dữ liệu chưa hiển thị được tiếng Việt.
- Bạn muốn điều chỉnh lại nội dung, vị trí của các Label hiển thị trên form.

- Bạn muốn dữ liệu trong table DMPhongban sẽ được nạp vào DataSet ngay khi form vừa khởi động, mà không cần đợi nhấp button Load.
- Thực tế, những điều chỉnh này sẽ được thực hiện một cách hết sức dễ dàng.

6a. Chỉnh sửa dữ liệu hiển thị trên form.

- Nhấp đúp lên form Fphongban.vb trong cửa sổ Solution Explorer để mở lại giao diện thiết kế form.
- Định lại các thuộc tính của form như sau:
 - Text = Fphongban (*tiêu đề form*)
 - FormBorderStyle = Sizable (*kiểu viền form*)
 - BackColor = 192, 192, 255 (*màu nền của form*)
 - Name = Fphongban (*tên form*)
 - StartPosition = WindowsDefaultLocation (*vị trí form khi thi hành là mặc định*)
 - Size = 450, 250 (*kích thước form lúc thiết kế*)
 - WindowState = Normal (*kích thước form hiển thị như khi thiết kế*)
 - Localizable = False (*không thể định lại kích thước form khi đang thi hành*)
 - Opacity = 100% (*form hiển thị rõ ràng như khi thiết kế, không có độ trong suốt*)
- Chọn View ➡ Toolbox kích hoạt hộp công cụ. Đặt một điều khiển Label lên form và định lại các thuộc tính như sau:
 - Name = Label1
 - Text = Cập nhật danh mục phòng ban
 - BackColor = 192, 192, 255

- Font = VNI-Times, 14pt, Style = Bold
- Size = 272, 32
- Location = 88, 8
- Định lại thuộc tính cho 2 Label đang có sẵn trên form:
 - * Label Mapb:
 - Name = LblMapb
 - Text = Mã phòng ban
 - BackColor = 192, 192, 255
 - Font = VNI-Times, 12pt, Style = Bold
 - * Label Mapb:
 - Name = LblTenpb
 - Text = Tên phòng ban
 - BackColor = 192, 192, 255
 - Font = VNI-Times, 12pt, Style = Bold
- Định lại thuộc tính Font cho 2 Textbox trên form: VNI-Times, Regular, 11.
- Định lại vị trí cho các button trên form như Hình 8.24.



Hình 8.24 Form Fphongban sau khi chỉnh sửa.

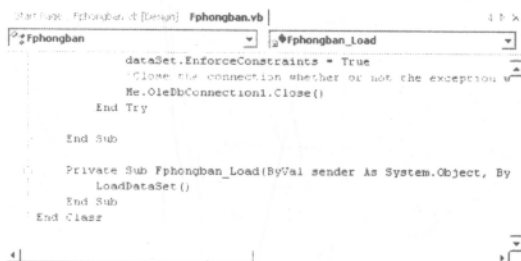
6b. Để nạp dữ liệu vào DataSet ngay khi form vừa khởi động.

Nhấp đúp lên form để mở cửa sổ Code, xong viết thêm đoạn lệnh sau vào cuối lớp (trước từ khóa End Class – Hình 8.25).

Thủ tục được gọi khi form vừa nạp

```
Private Sub Fphongban_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles MyBase.Load
    LoadDataSet()
End Sub
```

Xong biên dịch lại chương trình và chạy thử, form của bạn giờ đây sẽ trông khác hẳn như Hình 8.26.



Hình 8.25 Thêm thủ tục biến cố Load form Fphongban.

Fphongban

Cập nhật danh mục phòng ban

Mã phòng ban	G0001	Update
Tên phòng ban	Phòng Nhân Sự	Cancel All

Load << < 1 of 8 > >>

Add Delete Cancel

Hình 8.26 Kết quả form Fphongban khi vừa thi hành.

Bài 9.

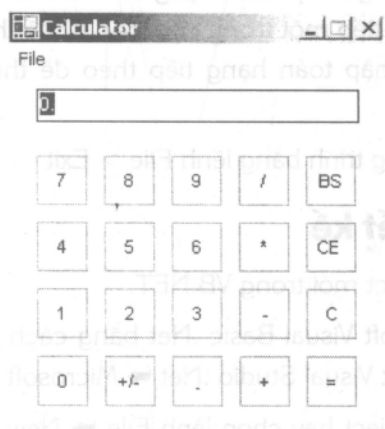
Chương trình Calculator

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với việc sử dụng các điều khiển và các hàm xử lý dữ liệu được xây dựng sẵn trong VB.NET. Từ đó, so sánh sự khác biệt với VB6, nếu bạn đã có dịp làm quen.

Hoạt động chương trình

Khi thi hành, chương trình sẽ hiển thị 1 máy tính tay có giao diện như Hình 9.1.



Hình 9.1 Máy tính tay khi thi hành chương trình Calculator.

Người dùng có thể nhập các phím số (0->9) hay gõ số trực tiếp vào.

Máy tính cung cấp các toán tử sau:

+/-: đổi số dương thành số âm và ngược lại

+: phép cộng

-: phép trừ

*: phép nhân

/: phép chia

=: lệnh gán, thực hiện phép toán

C: hủy bỏ toán tử vừa thực hiện, reset chương trình trở lại trạng thái ban đầu

CE: hủy bỏ toán hạng vừa nhập

BS: xóa bớt 1 ký số trong toán hạng đang nhập

.: đánh dấu vị trí bắt đầu phần thập phân trong số

Thao tác: Người dùng sẽ gõ vào toán hạng 1, xong nhấp một trong các toán tử, trước khi gõ tiếp toán hạng 2. Nhấn nút = để hoàn thành phép toán hoặc nhấp tiếp một trong các toán tử để thực hiện phép toán cũ và tiếp tục chờ nhập toán hạng tiếp theo để thực hiện phép toán mới.

- Kết thúc chương trình bằng lệnh File > Exit

Hướng dẫn thiết kế

Bước 1. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File ➡ New Project để gọi cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = *Calc*
- Định thư mục lưu project: Location = *C:\BTVBNET*

- Nhấp tùy chọn **Close Solution**.
- Chọn **Project types = Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục **Templates**
- Xong nhấp **OK**

VB.NET sẽ tự động khởi tạo một project mới với một form trống mặc định là **Form1.vb** chứa trong thư mục **CALC**, là thư mục con của **C:\BTVBNET**.

Bước 2. Định lại thuộc tính Form1.

Chọn **Form1** và định lại các thuộc tính sau trong cửa sổ **Properties**:

Name = Calculator

Text = Calculator

StartPosition = CenterScreen

Size = 232, 258

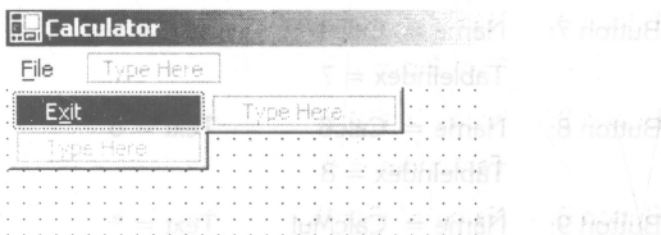
MaximizeBox = False

Bước 3. Định nghĩa menu cho ứng dụng.

Nhấp đúp lên công cụ **MainMenu** trên **Toolbox** để bổ sung điều khiển này vào form. Định lại tên cho điều khiển như sau:

Name: CalcMenu

Tạo menu **File** bằng cách nhập từ **&File** vào trong mục **Type Here**, sau đó nhập lệnh **E&xit** vào mục **Type Here** ở bên dưới (Hình 9.2).



Hình 9.2 Máy tính tay khi thi hành chương trình Calculator.

Bước 4. Bổ sung 1 Text box và 20 button vào form.

Xong định lại thuộc tính cho các điều khiển như sau:

Text box: Name = CalcField

Wordwrap = True

TabIndex = 0

TextAlign = Right

Text = 0.

RightToLeft = No

Button 1: Name = Calc7 Text = 7

TabIndex = 1

Button 2: Name = Calc8 Text = 8

TabIndex = 2

Button 3: Name = Calc9 Text = 9

TabIndex = 3

Button 4: Name = CalcDiv Text = /

TabIndex = 4

Button 5: Name = CalcBS Text = BS

TabIndex = 5

Button 6: Name = Calc4 Text = 4

TabIndex = 6

Button 7: Name = Calc5 Text = 5

TabIndex = 7

Button 8: Name = Calc6 Text = 6

TabIndex = 8

Button 9: Name = CalcMul Text = *

TabIndex = 9

Button 10:	Name = CalcCE TableIndex = 10	Text = CE
Button 11:	Name = Calc1 TableIndex = 11	Text = 1
Button 12:	Name = Calc2 TableIndex = 12	Text = 2
Button 13:	Name = Calc3 TableIndex = 13	Text = 3
Button 14:	Name = CalcSub TableIndex = 14	Text = -
Button 15:	Name = CalcCan TableIndex = 15	Text = C
Button 16:	Name = Calc0 TableIndex = 16	Text = 0
Button 17:	Name = CalcSign TableIndex = 17	Text = +/-
Button 18:	Name = CalcDec TableIndex = 18	Text = .
Button 19:	Name = CalcPlus TableIndex = 19	Text = +
Button 20:	Name = CalcRes TableIndex = 20	Text = =

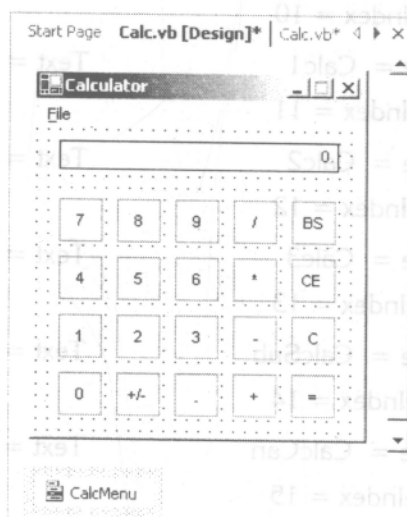
Các Button đều có cùng thuộc tính:

FlatStyle = Popup

TextAlign = MiddleCenter

Size = 30, 30

Kết quả thiết kế form sẽ như Hình 9.3.



Hình 9.3 Màn hình form Calculator sau khi thiết kế.

Bước 5. Viết mã lệnh xử lý và ý nghĩa mã lệnh.

Option Strict Off

Imports System.ComponentModel

Imports System.Drawing

Imports System.Windows.Forms

Imports Microsoft.VisualBasic

Public Class Calculator

Inherits System.Windows.Forms.Form

' Các biến dữ liệu

Private mOp1, mOp2 As Double ' Khai báo biến lưu 2 toán hạng sẽ nhập vào.

Private mNumOps As Integer ' Số các toán hạng

```
Private mLastInput As Operation ' Chỉ loại biến cố keypress cuối cùng.
Private mOpFlag As String       ' Cho biết toán hạng đang nhập.
Private mOpPrev As String       ' Toán hạng trước đó
Private mMinus As String        ' Định nghĩa cho dấu âm #define "-"
Private Enum Operation
    None = 0
    Operand
    Operator
    CE
    Cancel
End Enum
Public Sub New()
    MyBase.New()
    Calculator = Me
    InitializeComponent()
    Reset()
    mMinus = "-"
End Sub

Protected 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
```


' Vùng mã lệnh được phát sinh trong quá trình thiết kế Form Calculator

' Bạn không cần phải điều chỉnh gì trong phần này

#Region " Windows Form Designer generated code "

```
Private components As System.ComponentModel.Container  
Friend WithEvents ExitMenu As System.Windows.Forms.MenuItem  
Friend WithEvents FileMenu As System.Windows.Forms.MenuItem  
Friend WithEvents CalcMenu As System.Windows.Forms.MainMenu  
Friend WithEvents CalcRes As System.Windows.Forms.Button  
Friend WithEvents CalcPlus As System.Windows.Forms.Button  
Friend WithEvents CalcDec As System.Windows.Forms.Button  
Friend WithEvents CalcSign As System.Windows.Forms.Button  
Friend WithEvents Calc0 As System.Windows.Forms.Button  
Friend WithEvents CalcCan As System.Windows.Forms.Button  
Friend WithEvents CalcSub As System.Windows.Forms.Button  
Friend WithEvents Calc3 As System.Windows.Forms.Button  
Friend WithEvents Calc2 As System.Windows.Forms.Button  
Friend WithEvents Calc1 As System.Windows.Forms.Button  
Friend WithEvents CalcCE As System.Windows.Forms.Button  
Friend WithEvents CalcMul As System.Windows.Forms.Button  
Friend WithEvents Calc6 As System.Windows.Forms.Button  
Friend WithEvents Calc5 As System.Windows.Forms.Button  
Friend WithEvents Calc4 As System.Windows.Forms.Button  
Friend WithEvents CalcBS As System.Windows.Forms.Button  
Friend WithEvents CalcDiv As System.Windows.Forms.Button  
Friend WithEvents Calc9 As System.Windows.Forms.Button  
Friend WithEvents Calc8 As System.Windows.Forms.Button
```

```
Friend WithEvents Calc7 As System.Windows.Forms.Button
Friend WithEvents CalcField As System.Windows.Forms.TextBox
Dim WithEvents Calculator As System.Windows.Forms.Form
Private Sub InitializeComponent()
    ' Mã lệnh định nghĩa các thành phần trên form
    Me.CalcSub = New System.Windows.Forms.Button()
    Me.CalcField = New System.Windows.Forms.TextBox()
    Me.CalcDiv = New System.Windows.Forms.Button()
    Me.CalcPlus = New System.Windows.Forms.Button()
    Me.CalcRes = New System.Windows.Forms.Button()
    Me.CalcCan = New System.Windows.Forms.Button()
    Me.Calc7 = New System.Windows.Forms.Button()
    Me.Calc6 = New System.Windows.Forms.Button()
    Me.CalcBS = New System.Windows.Forms.Button()
    Me.Calc4 = New System.Windows.Forms.Button()
    Me.Calc3 = New System.Windows.Forms.Button()
    Me.Calc2 = New System.Windows.Forms.Button()
    Me.CalcSign = New System.Windows.Forms.Button()
    Me.Calc() = New System.Windows.Forms.Button()
    Me.Calc9 = New System.Windows.Forms.Button()
    Me.Calc8 = New System.Windows.Forms.Button()
    Me.CalcCE = New System.Windows.Forms.Button()
    Me.FileMenu = New System.Windows.Forms.MenuItem()
    Me.ExitMenu = New System.Windows.Forms.MenuItem()
    Me.Calc5 = New System.Windows.Forms.Button()
    Me.CalcDec = New System.Windows.Forms.Button()
```

```

Me.Calc1 = New System.Windows.Forms.Button()
Me.CalcMul = New System.Windows.Forms.Button()
Me.CalcMenu = New System.Windows.Forms.MainMenu()
Me.SuspendLayout()
' CalcSub
Me.CalcSub.FlatStyle = System.Windows.Forms.FlatStyle.Popup
Me.CalcSub.Location = New System.Drawing.Point(136, 128)
Me.CalcSub.Name = "CalcSub"
Me.CalcSub.Size = New System.Drawing.Size(30, 30)
Me.CalcSub.TabIndex = 14
Me.CalcSub.Text = "."

```

```

.....
End Sub

```

```

#End Region

```

' Phần mã lệnh bạn phải nhập thêm sau khi thiết kế form để xử lý các biến cố

' Biến cố nhấp button B5 - xóa bớt 1 ký số trong toán hạng đang nhập

```

Protected Sub CalcBS_Click(ByVal sender As System.Object, ByVal e As

```

```

    System.EventArgs) Handles CalcBS.Click

```

```

    Dim Value As String

```

```

    Value = CalcField.Text

```

```

    If ((Value.CompareTo("0.") <> 0) And (Value.Length <> 0)) Then

```

```

        Value = Value.Remove(Value.Length - 1, 1)

```

```

    End If

```

```

    If ((Value.Length = 0) Or (Value = "-")) Then

```

```

        Value = "0."

```

End If

CalcField.Text = Value

End Sub

' Biến cố nhấp button C (Cancel) - hủy bỏ toán tử vừa thực hiện,

' Reset chương trình trở lại trạng thái ban đầu

Protected Sub CalcCan_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles CalcCan.Click

Reset()

End Sub

' Biến cố nhấp button CE (Cancel Entry) - hủy bỏ toán hạng vừa nhập

Protected Sub CalcCE_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles CalcCE.Click

If (mLastInput.Equals(Operation.Operand)) Then

CalcField.Text = "0."

ElseIf (mLastInput.Equals(Operation.Operator)) Then

mOpFlag = mOpPrev

End If

mLastInput = Operation.CE

End Sub

' Biến cố click button / - thực hiện phép chia

Protected Sub CalcDiv_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles CalcDiv.Click

CalcRes_Click(sender, e) *' gọi thủ tục xử lý phép toán*

End Sub

' Biến cố click button -, thực hiện phép trừ

Protected Sub CalcSub_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles CalcSub.Click

```
CalcRes_Click(sender, e)
```

```
End Sub
```

*' Biến cố nhấp button *, thực hiện phép nhân*

```
Protected Sub CalcMul_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles CalcMul.Click
```

```
CalcRes_Click(sender, e)
```

```
End Sub
```

' Biến cố nhấp button +, thực hiện phép cộng

```
Protected Sub CalcPlus_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles CalcPlus.Click
```

```
CalcRes_Click(sender, e)
```

```
End Sub
```

*' Thủ tục xử lý các phép toán +, -, *, / và khi người dùng nhấp button =*

```
Protected Sub CalcRes_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles CalcRes.Click
```

```
If (CalcField.Text.Length = 0) Then
```

```
Return
```

```
End If
```

```
If (mLastInput.Equals(Operation.Operand)) Then
```

```
mNumOps = mNumOps + 1
```

```
End If
```

```
Select Case mNumOps
```

```
Case 1
```

```
mOp1 = Double.Parse(CalcField.Text)
```

```
Case 2
```

```
mOp2 = Double.Parse(CalcField.Text)
```

```

Select Case mOpFlag
    Case "+"
        mOp1 = mOp1 + mOp2
    Case "-"
        mOp1 = mOp1 - mOp2
    Case "*"
        mOp1 = mOp1 * mOp2
    Case "/"
        If (mOp2 = 0) Then
            MessageBox.Show("Can't divide by zero!",
                "Calculator", MessageBoxButtons.OK)
        Else
            mOp1 = mOp1 / mOp2
        End If
    Case "="
        mOp1 = mOp2
End Select
CalcField.Text = mOp1.ToString()
mNumOps = 1
End Select
mLastInput = Operation.Operator
mOpPrev = mOpFlag
Dim SelButton As Button
SelButton = sender
mOpFlag = SelButton.Text
End Sub

```

‘ Biến cố nhấp button +/-, thực hiện phép đảo dấu toán hạng hiện hành

```
Protected Sub CalcSign_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles CalcSign.Click  
    If (CalcField().Text.Substring(0, 1) = "-") Then  
        CalcField().Text = CalcField().Text.TrimStart(mMinus.ToCharArray())  
    Else  
        CalcField().Text = mMinus + CalcField().Text  
    End If  
    mLastInput = Operation.Operand
```

End Sub

‘ Biến cố nhấp button 0

```
Protected Sub Calc0_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc0.Click  
    CalcNum_Click(sender, e)
```

End Sub

‘ Biến cố nhấp button 1

```
Protected Sub Calc1_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc1.Click  
    CalcNum_Click(sender, e)
```

End Sub

‘ Biến cố nhấp button 2

```
Protected Sub Calc2_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc2.Click  
    CalcNum_Click(sender, e)
```

End Sub

‘ Biến cố nhấp button 3

```
Protected Sub Calc3_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc3.Click  
    CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Biến cố nhấp button 4
```

```
Protected Sub Calc4_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc4.Click  
    CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Biến cố nhấp button 5
```

```
Protected Sub Calc5_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc5.Click  
    CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Biến cố nhấp button 6
```

```
Protected Sub Calc6_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc6.Click  
    CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Biến cố nhấp button 7
```

```
Protected Sub Calc7_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc7.Click  
    CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Biến cố nhấp button 8
```

```
Protected Sub Calc8_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles Calc8.Click
```



```
CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Biến cố nhấp button 9
```

```
Protected Sub Calc9_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Calc9.Click
```

```
CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Biến cố nhấp button (dấu chấm thập phân)
```

```
Protected Sub CalcDec_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles CalcDec.Click
```

```
CalcNum_Click(sender, e)
```

```
End Sub
```

```
' Thủ tục được gọi khi nhấp 1 trong 10 phím số (0 đến 9)
```

```
Protected Sub CalcNum_Click(ByVal sender As Object, ByVal e As  
System.EventArgs)
```

```
If (mLastInput.Equals(Operation.Operand) = False) Then
```

```
CalcField.Text = "0."
```

```
End If
```

```
Dim SelButton As Button
```

```
SelButton = sender
```

```
FormatEditField(SelButton.Text)
```

```
mLastInput = Operation.Operand
```

```
End Sub
```

```
' Thủ tục biến cố nhấp menu Exit
```

```
Protected Sub ExitMenu_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles ExitMenu.Click
```

Application.Exit()

End Sub

' Định dạng giá trị số hiển thị trong Text box

Private Sub FormatEditField(ByVal NewChar As String)

Dim Value As String

Value = CalcField.Text

If (Value.CompareTo("0.") = 0) Then

Value = NewChar

Else

' Xác định liệu có nhiều hơn 1 ký số

If (NewChar = ".") Then

' nếu button vừa được nhấn là dấu chấm thập phân

If (Value.IndexOf(NewChar) <> -1) Then

Return

End If

End If

Value = Value + NewChar

End If

CalcField.Text = Value

End Sub

' Thủ tục khởi tạo các giá trị ban đầu

Private Sub Reset()

mOp1 = 0

mOp2 = 0

mNumOps = 0

mLastInput = Operation.None

```
mOpFlag = ""  
mOpPrev = ""  
CalcField.Text = "0."  
  
End Sub  
  
End Class
```

Bước 6. Chạy thử chương trình để xem kết quả.

- Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.
- Chọn lệnh Build ➡ Build Solution để biên dịch project.
- Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start chạy thử chương trình.

Bài 10.

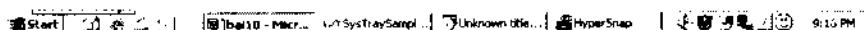
Chương trình SystemTray

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với kỹ thuật xây dựng 1 chương trình chạy thường trú trên thanh Taskbar, cách kích hoạt và kết thúc chương trình. Ngoài ra, bạn cũng làm quen với đối tượng DateTime, giúp bạn dễ dàng lấy ngày giờ hệ thống hiện hành và hiển thị nó ra màn hình.

Hoạt động chương trình

Khi thi hành, chương trình sẽ được nạp và chạy thường trú với biểu tượng khuôn mặt cười hiển thị trên thanh Taskbar như Hình 10.1.



Hình 10.1 Biểu tượng khuôn mặt cười trên thanh Taskbar sau khi chạy chương trình.

Chương trình cho phép người dùng xem ngày giờ hiện hành của máy tính.

Để kích hoạt chương trình, người dùng có thể nhấp nút phải mouse lên biểu tượng khuôn mặt cười trên thanh Taskbar, để gọi menu nhanh như Hình 10.2.



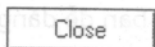
Hình 10.2 Menu nhanh của chương trình SystemTray.

- Chọn lệnh Show Form... để gọi cửa sổ Clock, hiển thị ngày và giờ hệ thống hiện hành (Hình 10.3). Nhấp button Close để đóng cửa sổ Clock.



10/10/2003

09:30:25



Hình 10.3 Cửa sổ Clock.

- Chọn lệnh Toggle Image đổi icon khuôn mặt cười của chương trình trên thanh Taskbar thành icon khuôn mặt buồn hoặc ngược lại (Hình 10.4).



Hình 10.4 Biểu tượng khuôn mặt buồn trên thanh Taskbar sau khi chạy chương trình.

- Chọn lệnh Exit kết thúc chương trình.

Hướng dẫn thiết kế

Bước 1. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File ➡ New Project kích hoạt của sổ New Project.

- Đặt tên tập tin lưu project: Name = *SysTraySample.vbproj*
- Định thư mục lưu project: Location = *C:\BTVBNET*
- Nhấp tùy chọn Close Solution.
- Chọn Project types = *Visual Basic Projects*, xong nhấp biểu tượng *Windows Application* trong mục Templates.
- Xong nhấp OK.

VB.NET sẽ tự động khởi tạo 1 project mới với 1 form trống mặc định là Form1.vb chứa trong thư mục *SysTraySample*, là thư mục con của *C:\BTVBNET*.

Bước 2. Bổ sung hai tập tin icon vào project.

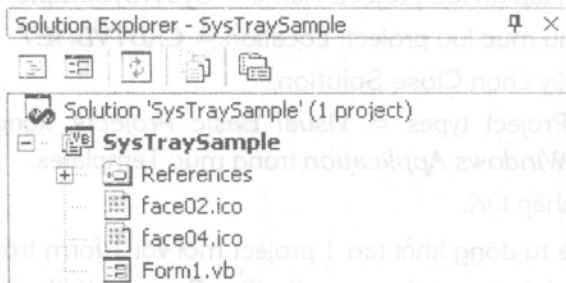
Sao chép hai tập tin icon face02.ico và face04.ico vào thư mục chứa project *C:\BTVBNET\SysTraySample*.

Chọn lệnh Project ➡ Add Existing Item... Chọn thư mục *SysTraySample*, đặt File of Type = All Files, nhấp tập tin face02, xong nhấp Open (Hình 10.4).



Hình 10.4 Chọn tập tin icon bổ sung vào project.

Thực hiện tương tự để bổ sung tập tin face04 vào project. Kết quả của sổ Solution Explorer sẽ như Hình 10.5.



Hình 10.5 Cửa sổ Solution Explorer sau khi bổ sung hai tập tin icon.

Định lại thuộc tính cho hai tập tin icon: Nhấp nút phải mouse lên biểu tượng face02.ico trong cửa sổ Solution Explorer, xong chọn lệnh Properties, đổi thuộc tính Build Action = Embedded Resource (Hình 10.6). Thực hiện tương tự với face04.ico.

Chú ý: Việc định lại thuộc tính Build Action cho các tập tin ảnh là rất quan trọng. Nếu bạn bỏ qua bước này, chương trình sẽ không chạy và báo lỗi.



Hình 10.6 Định lại thuộc tính Build Action cho hai tập tin icon.

MẠCH NƯỚC

- ♦ Hai tập tin này chứa 2 biểu tượng khuôn mặt cười và khuôn mặt buồn. Bạn dễ dàng tìm thấy trong Windows. Nếu không tìm thấy, có thể thay thế bằng 2 icon khác nhưng vẫn đặt tên tập tin là face02.ico và face04.ico để tiếp tục bài thực hành.

Bước 3. Định lại thuộc tính Form1.

Chọn Form1 và định lại các thuộc tính sau trong cửa sổ Properties:

Name = Form1

Text = (Rỗng)

AccessibleRole = None

FormBorderStyle = None

Enabled = False

StartPosition = Manual

Size = 1, 7

ControlBox = False

WindowState = Normal

MaximizeBox = False

MinimizeBox = False

Opacity = 0%

ShowInTaskBar = False

SizeGripStyle = Hide

Bước 4. Đặt điều khiển NotifyIcon vào form1.

Nhấp đúp lên công cụ NotifyIcon trên Toolbox để bổ sung điều khiển này vào form. Định lại tên cho điều khiển như sau:

Name: NotifyIcon

Visible: True

Bước 5. Viết mã lệnh xử lý cho Form1.

Nhấp lại Form1, chọn lệnh View ➡ Code, nhập thêm và điều chỉnh đoạn mã lệnh do VB.Net tự phát sinh cho Form1 trong quá trình thiết kế, để được kết quả như sau:

```
Imports System.ComponentModel
```

```
Imports System.Drawing
```

```
Imports System.Windows.Forms
```

```
Imports System.Resources
```

```
' Định nghĩa lớp FORM1
```

Public Class Form1

```
Inherits System.Windows.Forms.Form ' Kế thừa lớp Form
```

```
' Định nghĩa các biến chỉ các icon notifyicon
```

```
' Trong đó mSmileIcon - Biến tham chiếu tới icon khuôn mặt cười
```

```
' mFrownIcon - Biến tham chiếu tới icon khuôn mặt buồn
```

```
' mSmileDisplayed - Biến xác định icon đang hiển thị
```

```
' Lưu ý: SysTraySample là tên thư mục đang chứa project và các tập tin icon
```

```
Private mSmileIcon As New Icon(GetType(Form1).Assembly,
```

```
    GetManifestResourceStream("SysTraySample.face02.ico"))
```

```
Private mFrownIcon As New Icon(GetType(Form1).Assembly,
```

```
    GetManifestResourceStream("SysTraySample.face04.ico"))
```

```
Private mSmileDisplayed As Boolean
```

```
' Thủ tục New() được gọi khi nạp lớp Form1
```

Public Sub New()

```
MyBase.New
```

```
Form1 = Me
```

```
' Thủ tục này được yêu cầu bởi trình thiết kế Win Form
```

```
InitializeComponent
```

' Do form1 không được sử dụng trực tiếp, nên cần ẩn nó ngay lập tức
Me.Hide()

' Gọi thủ tục khởi tạo icon trên thanh Taskbar
InitializeNotifyIcon()

End Sub

' Thủ tục khởi tạo tray icon

Private Sub InitializeNotifyIcon()

' Khởi tạo icon mặc định là khuôn mặt cười

notifyIcon = New System.Windows.Forms.NotifyIcon()

NotifyIcon.Icon = mSmileIcon

NotifyIcon.Text = "Right Click for the menu"

NotifyIcon.Visible = True

mSmileDisplayed = True *' Icon đang hiển thị là khuôn mặt cười*

' Chèn các lệnh (mục chọn) vào 1 mảng và thêm nó tới menu ngữ cảnh

Dim mnulmts(3) As MenuItem *' Mảng gồm 4 phần tử*

' Mục chọn đầu tiên là lệnh Show Form..., đây cũng là tùy chọn mặc định

mnulmts(0) = New MenuItem("Show Form...", New

EventHandler(AddressOf Me.ShowFormSelect))

mnulmts(0).DefaultItem = True

' Mục chọn thứ 2 là lệnh Toggle Image

mnulmts(1) = New MenuItem("Toggle Image", New

EventHandler(AddressOf Me.ToggleImageSelect))

' Mục chọn thứ 3 là thanh phân cách

mnulmts(2) = New MenuItem("-")

' Mục chọn thứ 4 là lệnh Exit

mnulmts(3) = New MenuItem("Exit", New EventHandler(AddressOf
Me.ExitSelect))

' Thêm mảng vào menu ngữ cảnh của điều khiển notifyicon

Dim notifyiconMnu As ContextMenu = New ContextMenu(mnultms)

notifyicon.ContextMenu = notifyiconMnu

End Sub

' Thủ tục được gọi khi người dùng chọn lệnh Show Form...

Public Sub ShowFormSelect(ByVal sender As Object, ByVal e As

System.EventArgs)

' Hiển thị hộp thoại Settings (nạp form Settings)

Dim SettingsForm As New SettingsForm()

SettingsForm.ShowDialog()

End Sub

' Thủ tục được gọi khi người dùng chọn lệnh Toggle Image

Public Sub ToggleImageSelect(ByVal sender As Object, ByVal e As

System.EventArgs)

' Xác định icon nào đang hiển thị và chuyển thành icon khác

' Nếu Icon đang hiển thị là khuôn mặt cười

If mSmileDisplayed Then

' Hiển thị icon khuôn mặt buồn lên Taskbar

NotifyIcon.Icon = mFrownIcon

NotifyIcon.Text = "Sad"

mSmileDisplayed = False

Else

' Ngược lại nếu Icon đang hiển thị là khuôn mặt buồn

' Hiển thị icon khuôn mặt cười lên Taskbar

NotifyIcon.Icon = mSmileIcon

NotifyIcon.Text = "Happy"

```
mSmileDisplayed = True
```

```
End If
```

End Sub

' Thủ tục được gọi khi người dùng chọn lệnh Exit

```
Public Sub ExitSelect(ByVal sender As Object, ByVal e As System.EventArgs)
```

' ẩn icon trên Taskbar

```
NotifyIcon.Visible = False
```

' đóng form1, kết thúc chương trình

```
Me.Close()
```

End Sub

' Thủ tục được gọi khi Form1 được đóng, nhằm giải phóng các thành phần

' đã khởi tạo

```
Protected 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

' Phần mã lệnh được VB.Net tự động khởi tạo trong quá trình thiết kế Form

#Region " Windows Form Designer generated code "

```
Private components As System.ComponentModel.Container
```

```
Private WithEvents notifyicon As System.Windows.Forms.NotifyIcon
```

```
Dim WithEvents Form1 As System.Windows.Forms.Form
```

' Thủ tục khởi tạo các thành phần trên Form1

' Nội dung thủ tục này có thể được sửa đổi trong quá trình bạn thiết kế Form

Private Sub InitializeComponent()

```
Me.components = New System.ComponentModel.Container()
Me.notifyicon = New System.Windows.Forms.NotifyIcon()
'@design Me.TrayHeight = 90
'@design Me.TrayLargeIcon = False
'@design Me.SnapToGrid = False
'@design Me.TrayAutoArrange = True
'@design notifyicon.SetLocation(New System.Drawing.Point(7, 7))
notifyicon.Text = ""
notifyicon.Visible = True
Me.MaximizeBox = False
Me.StartPosition = System.Windows.Forms.FormStartPosition.Manual
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.None
Me.Enabled = False
Me.AccessibleRole = System.Windows.Forms.AccessibleRole.None
Me.ShowInTaskbar = False
Me.SizeGripStyle = System.Windows.Forms.SizeGripStyle.Hide
Me.ControlBox = False
Me.MinimizeBox = False
Me.ClientSize = New System.Drawing.Size(1, 7)
Me.Opacity = 0.0#
```

End Sub

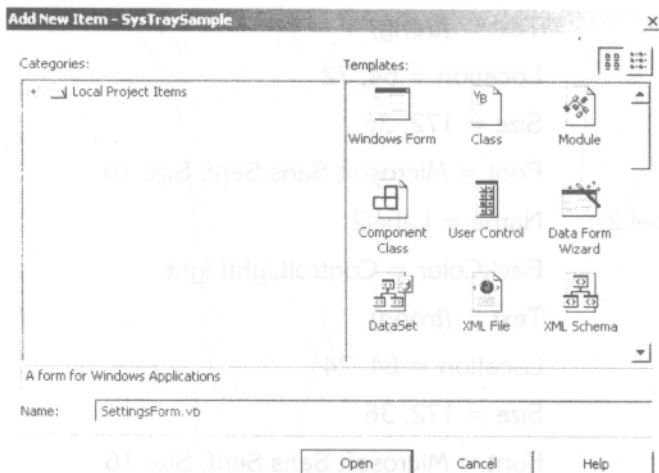
#End Region

End Class

Bước 6. Thiết kế form SettingsForm.

Giờ đây, bạn cần thiết kế form SettingsForm nhằm hiển thị ngày giờ hiện hành của hệ thống khi Form được gọi.

- Chọn lệnh Project ➔ Add Windows Form. Trong mục Templates, nhấp biểu tượng Windows Form, nhập tên form mới là SettingsForm trong mục Name, xong nhấp Open (Hình 10.7).



Hình 10.7 Thêm form mới vào project.

- Định lại các thuộc tính cho form SettingsForm như sau:

Name = SettingsForm

Text = Clock

BackColor = ControlLightLight

StartPosition = WindowsDefaultLocation

Size = 300,200

ControlBox = True

Opacity = 100%

ShowInTaskBar = True

- Đặt lên form 2 điều khiển Label và định lại các thuộc tính như sau:

Label 1: Name = Label1

BackColor = ControlLightLight

Text = (trống)

Location = 64, 72

Size = 172, 36

Font = Microsoft Sans Serif, Size 16

Label 2: Name = Label2

BackColor = ControlLightLight

Text = (trống)

Location = 64, 24

Size = 172, 36

Font = Microsoft Sans Serif, Size 16

- Đặt lên form 1 điều khiển Button và định lại các thuộc tính như sau:

Button 1: Name = Button1

Text = Close

FlatStyle = Standard

TextAlign = MiddleCenter

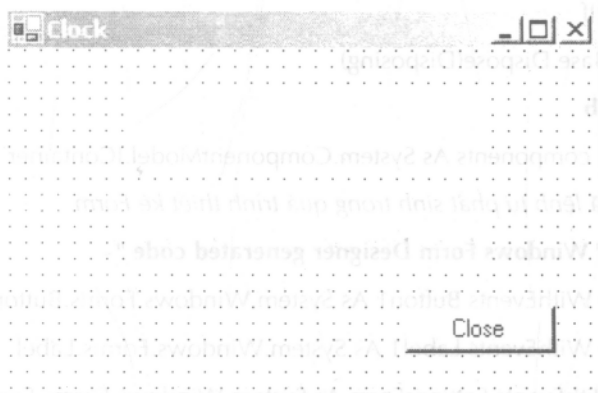
- Đặt lên form 1 điều khiển Timer và định lại các thuộc tính như sau:

Name = Timer1

Interval = 1000

Enabled = True

Kết quả thiết kế form sẽ như Hình 10.8.



Hình 10.8 Màn hình form *SettingsForm* sau khi thiết kế.

Bước 7. Viết mã lệnh xử lý form *SettingsForm*.

```
Imports System.Drawing
```

```
Imports System.Windows.Forms
```

```
Imports System.ComponentModel
```

Public Class SettingsForm

```
Inherits System.Windows.Forms.Form
```

```
Public Sub New()
```

```
MyBase.New
```

```
SettingsForm = Me
```

```
InitializeComponent()
```

```
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

Private components As System.ComponentModel.IContainer

' Phần mã lệnh tự phát sinh trong quá trình thiết kế Form

#Region " Windows Form Designer generated code "

Private WithEvents Button1 As System.Windows.Forms.Button

Private WithEvents Label1 As System.Windows.Forms.Label

Dim WithEvents SettingsForm As System.Windows.Forms.Form

Friend WithEvents Timer1 As System.Windows.Forms.Timer

Friend WithEvents Label2 As System.Windows.Forms.Label

Private Sub InitializeComponent()

Me.components = New System.ComponentModel.Container()

Me.Button1 = New System.Windows.Forms.Button()

Me.Label1 = New System.Windows.Forms.Label()

Me.Timer1 = New System.Windows.Forms.Timer(Me.components)

Me.Label2 = New System.Windows.Forms.Label()

Me.SuspendLayout()

Me.Button1.Location = New System.Drawing.Point(196, 128)

Me.Button1.Name = "Button1"

Me.Button1.TabIndex = 1

Me.Button1.Text = "Close"

Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 16.0!)

Me.Label1.Location = New System.Drawing.Point(64, 72)

Me.Label1.Name = "Label1"

```
Me.Label1.Size = New System.Drawing.Size(172, 36)
Me.Label1.TabIndex = 0
Me.Timer1.Enabled = True
Me.Timer1.Interval = 1000
Me.Label2.Font = New System.Drawing.Font("Microsoft Sans Serif",
    15.75!, System.Drawing.FontStyle.Regular,
    System.Drawing.GraphicsUnit.Point, CType(0, Byte))
Me.Label2.Location = New System.Drawing.Point(64, 24)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(172, 36)
Me.Label2.TabIndex = 2
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.BackColor = System.Drawing.SystemColors.ControlLightLight
Me.ClientSize = New System.Drawing.Size(292, 173)
Me.Controls.AddRange(New System.Windows.Forms.Control()
    {Me.Label2, Me.Button1, Me.Label1})
Me.Name = "SettingsForm"
Me.Text = "Clock"
Me.ResumeLayout(False)
```

End Sub

#End Region

' Thủ tục biến cố được gọi khi người dùng nhấp nút Close

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles Button1.Click
    Me.Close() ' đóng form SettingsForm
```

End Sub

' Thủ tục biến cố được gọi sau từng khoảng thời gian 1 giây

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles Timer1.Tick

' Định nghĩa biến đối tượng dt xác định ngày giờ hệ thống hiện hành

Dim dt As DateTime = DateTime.Now

' Gán ngày tháng cho Label2 với dạng "mm/dd/yyyy"

Label2.Text = dt.ToString("d")

' Gán thời gian cho Label1 với dạng "hh:mm:ss"

Label1.Text = dt.ToString("hh:mm:ss")

End Sub

End Class

Bước 8. Chạy thử chương trình để xem kết quả.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

- Chọn lệnh Build ➡ Build Solution để biên dịch project.
- Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start để chạy thử chương trình.

Bước 9. Thi hành chương trình dưới dạng tập tin EXE.

Để chạy chương trình ngoài Windows mà không cần khởi động VB.Net, chọn tập tin SystemTray.exe trong thư mục C:\BTVBNET\SystemTray\Bin.

Bài 11.

Chương trình cơ sở dữ liệu – Thiết kế form đôi

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với kỹ thuật thiết kế form đôi, có thể tiến hành cập nhật dữ liệu cho 2 table có cùng khóa quan hệ trong cơ sở dữ liệu Access với VB.NET.

Hoạt động chương trình

Khi thi hành chương trình, bạn sẽ được màn hình có giao diện như Hình 11.1. Chương trình sẽ tự động mở màn hình xem và cập nhật danh mục phòng ban.

Hình 11.1 Giao diện màn hình khi form Fnhanvien được nạp.

Người dùng có thể nhấp nút Load để tiến hành nạp dữ liệu trong cơ sở dữ liệu Access **Nhansu.mdb** ra điều khiển DataSet và hiển thị thông tin các nhân viên của phòng ban đầu tiên lên form (Hình 11.2). Người dùng có thể điều chỉnh thông tin nhân viên trong phòng ban vừa hiển thị, cũng như có thể điều chỉnh mã phòng và tên phòng ban hiện hành. Người dùng cũng có thể sử dụng các button <<, <, >, >> để xem danh sách nhân viên của các phòng ban khác.

MaPB	MaNV	Hovaten	Phai	DCti	Dienthoai	Ngaysinh	Chucvu	Doc	Tiegiao
N0001	N006	Lương Bửu Thủ	☒	92 Lê Tuấn	8940549	10/24/1980	Nhau viên	Kinh	Thừa Kế G
N0001	N007	Lâm Sơn Hoa	☒	74 Kỳ Coa Q1	8740374	7/15/1979	Nhau viên	Kinh	Phân Giao
N0001	N111	Nguyễn Hoàn	☐	2 Nguyễn Cốc	8111111	9/20/1973	Nhau viên	Kinh	Phân Giao

Hình 11.2 Giao diện màn hình form Fnhanvien sau khi nhấp nút Load.

Sau đây là các nút chức năng xử lý chính của form:

- <<: chuyển đến phòng ban đầu tiên
- <: chuyển đến phòng ban trước đó
- >: chuyển đến phòng ban kế tiếp
- >>: chuyển đến phòng ban cuối cùng
- Load: nạp nội dung table DMPhongban và DMNhanvien cho điều khiển dữ liệu DataSet.
- Add: thêm phòng ban mới.
- Delete: xóa phòng ban hiện hành.

- **Cancel:** hủy bỏ thao tác chỉnh sửa phòng ban mới nhất, khi chưa thực hiện lệnh cập nhật vào table DMPhongban.
- **Update:** cập nhật các chỉnh sửa, thêm, xóa thông tin phòng ban, nhân viên vào table DMPhongban và DMNhanvien.
- **Cancel All:** hủy bỏ tất cả lệnh thêm, xóa, chỉnh sửa nội dung các record đã thực hiện, nhưng chưa được cập nhật vào table DMPhongban và DMNhanvien với lệnh Update.

CHÚ Ý

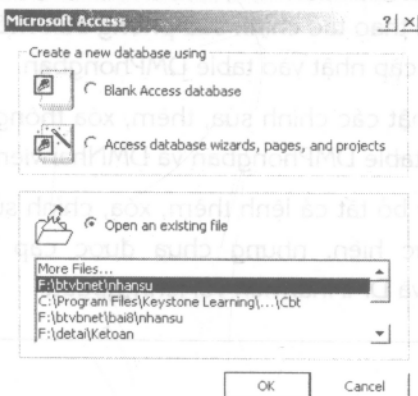
Form phải bảo đảm đáp ứng các yêu cầu sau:

- ◆ Khi 1 phòng ban bị xóa, các nhân viên đăng ký trong phòng ban đó cũng bị xóa theo.
 - ◆ Chỉ khi phòng ban mới được cập nhật vào table DMPhongban thì người dùng mới được thêm nhân viên mới vào phòng ban này.
 - ◆ Số lượng nhân viên thêm mới vào mỗi phòng ban là không hạn chế.
 - ◆ Có thể chuyển đổi nhân viên từ phòng ban này sang phòng ban khác một cách dễ dàng.
-

Hướng dẫn thiết kế

Bước 1. Thiết kế cơ sở dữ liệu nhansu.mdb với Microsoft Access 2000.

- Sử dụng lại cơ sở dữ liệu Access nhansu.mdb mà bạn đã tạo từ bài thực hành 8 bằng cách copy vào thư mục C:\BTVBNET.
- Khởi động Access 2000, chọn tùy chọn Open an Existing File (Hình 11.3), chọn cơ sở dữ liệu c:\BTVBNet\Nhansu.mdb, xong nhấp OK.



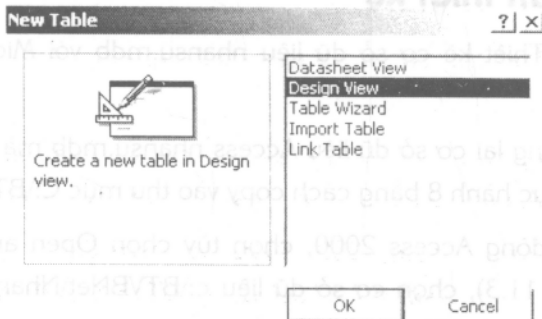
Hình 11.3 Cửa sổ Microsoft Access khi vừa khởi động.

MÁCH NƯỚC

- ◆ Nếu chưa thấy tên cơ sở dữ liệu cần mở trong danh sách thì nhấp More Files để tìm.

Trong cơ sở dữ liệu Nhansu.mdb đã có sẵn table DMPhongban và dữ liệu tương ứng mà bạn đã tạo từ bài thực hành 8. Giờ đây, bạn cần định nghĩa cấu trúc table DMNhanvien.

- Chọn tab Tables, xong nhấp nút New. Trong cửa sổ New Table (Hình 11.4), chọn Design View, xong nhấp OK.

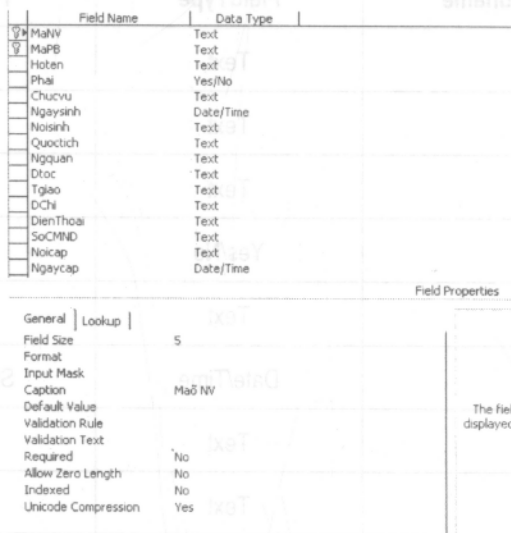


Hình 11.4 Cửa sổ New Table.

Khai báo tên, kiểu dữ liệu và kích thước các field trong table DMNhanvien như sau:

Fieldname	FieldType	FieldSize
Mapb	Text	5
Manv	Text	5
Hoten	Text	50
Phai	Yes/No	
Chucvu	Text	30
Ngaysinh	Date/Time	Short Date
Noisinh	Text	50
Quoctich	Text	20
Ngquan	Text	50
Dtoc	Text	20
Tgiao	Text	20
Dchi	Text	60
Dienthoai	Text	30
SoCMND	Text	12
Noicap	Text	20
Ngaycap	Date/Time	Short Date

- Đặt khóa chính cho field **MaPB** và **Manv** bằng cách chọn cả 2 field (giữ Shift và nhấp trên vùng xám trước tên field), xong nhấp biểu tượng khóa (Primary key) trên Toolbar Standard (Hình 11.5).



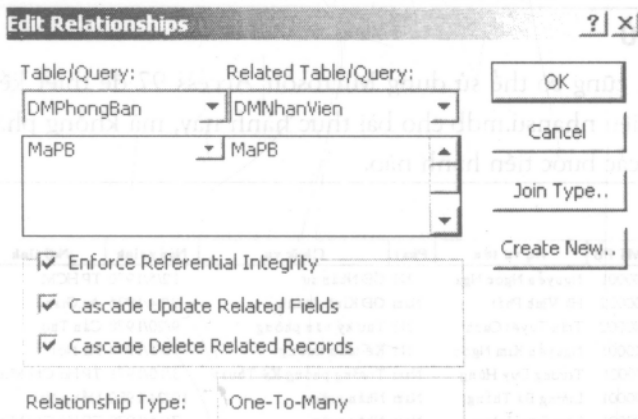
Hình 11.5 Định nghĩa các field cho table mới – DMNhanvien.

Nhấp nút Close ở góc phải của sổ hoặc nhấn Ctrl+W đóng table mới, lưu và đặt tên cho table mới trong cửa sổ Save As là DMNhanvien.

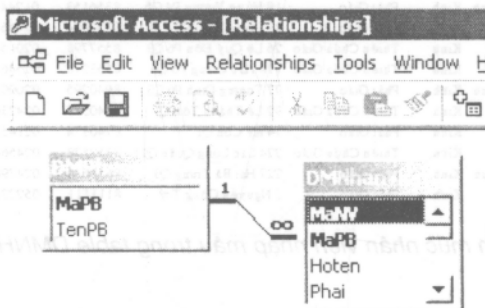
- Đặt mối quan hệ giữa 2 Table DMNhanvien với DMPhongban:

Chọn Tools ➡ Relationship, bổ sung 2 table DMNhanvien và DMPhongban vào cửa sổ Relationship, nhấn-giữ mouse trên file MaPB trong table DMPhongBan và kéo tới field MaPB trong table Nhanvien, cửa sổ Edit Relationships sẽ mở ra. Đánh dấu chọn các checkbox như Hình 11.6.

Kết quả của sổ Relationships sau khi đặt quan hệ sẽ như Hình 11.7.



Hình 11.6 Cửa sổ Edit Relationships.



Hình 11.7 Cửa sổ Relationships.

- Nhấp đúp lên table DMNhanvien để mở và nhập tay vào dữ liệu mẫu như Hình 11.8.

Xong đóng cơ sở dữ liệu Nhansu.mdb và thoát Access.

GHỊ CHÚ

- ♦ Bạn cũng có thể sử dụng Microsoft Access 97 để thiết kế cơ sở dữ liệu nhansu.mdb cho bài thực hành này, mà không phải thay đổi các bước tiến hành nào.

Mã NV	Mã PB	Họ và tên	Phái	Chức vụ	Ngày sinh	Nơi sinh	Quốc tịch
N001	G0001	Nguyễn Ngọc Nga	Nữ	GB Nhân sự	12/5/1970	TP HCM	Việt Nam
N002	G0002	Hà Vinh Phát	Nam	GB Kinh doanh	2/14/1975	An Giang	Việt Nam
N003	N0002	Trần Tuyết Canh	Nữ	Thư ký văn phòng	9/20/1978	Cần Thơ	Việt Nam
N004	K0001	Nguyễn Kim Ngọc	Nữ	Kế toán trưởng	6/1/1974	Hà Nội	Việt Nam
N005	T0001	Trương Duy Hùng	Nam	Trưởng phòng Kỹ Thuật	3/10/1971	TP Hồ Chí Minh	Việt Nam
N006	N0001	Lương Bá Thắng	Nam	Nhân viên	10/24/1980	Đông Nam	Việt Nam
N007	N0001	Lâm Sơn Hoàng	Nam	Nhân viên	7/15/1979	TP Hồ Chí Minh	Việt Nam
N008	P0001	Nguyễn Minh Hồng	Nam	Phó Tổng GB	11/30/1965	Hà Nội	Việt Nam
N009	M0001	Vương Ngọc Lan	Nữ	Trưởng phòng Marketing	1/31/1973	TP Hồ Chí Minh	Việt Nam
N111	N0001	Nguyễn Hoàng Anh	Nữ	Nhân viên	9/20/1975	Long An	Việt Nam

Mã NV	Ngay nhap	Dân tộc	Tôn giáo	Địa chỉ	Điện thoại	Số CMND	Nơi cấp	Ngày cấp
N001	TP Hồ Chí Minh	Kinh	Phật Giáo	19 Hùng Vương P4 Q5	8356153	012444564	CA Q5	4/3/1987
N002	An Giang	Kinh	Phật Giáo	89 Đồng Khởi Q1	8352074	024567899	CA Q1	4/25/1981
N003	Cần Thơ	Kinh	Thiên Chúa Giáo	26 Lê Quý Đôn P6 Q3	8557798	020154678	CA Q3	12/30/1983
N004	Hà Nội	Kinh	Thiên Chúa Giáo	178 Hồ Gươm P6 Q5	8553278	075467023	CA HN	9/10/1980
N005	TP Hồ Chí Minh	Kinh	Phật Giáo	77 Trương Định P6 Q3	8940795	020001340	CA Q3	2/15/1987
N006	Cần Thơ	Kinh	Thiên Chúa Giáo	92 Lê Thánh Tôn Q1	8940549	024336656	CA Q1	11/11/1985
N007	Đồng Tháp	Kinh	Phật Giáo	74 Kỳ Co Q1	8740374	021455124	CA Q1	12/30/1994
N008	Hà Nội	Kinh	Thiên Chúa Giáo	224 Lạc Long Quân Q1	8352578	024568702	CA HN	1/10/1987
N009	TP Hồ Chí Minh	Kinh	Phật Giáo	227 Hai Bà Trưng Q5	8651646	024798265	CA Q5	10/14/1982
N111	Long An	Kinh	Phật Giáo	2 Nguyễn Công Trứ	8111111	052221111	CA Q5	9/20/1989

Hình 11.8 Danh mục nhân viên nhập mẫu trong table DMNHANVIEN.

Bước 2. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➤ Programs ➤ Microsoft Visual Studio .Net ➤ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File > New Project để gọi cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = *Bai11*
- Định thư mục lưu project: Location = *C:\BTVBNET*
- Nhấp tùy chọn Close Solution.

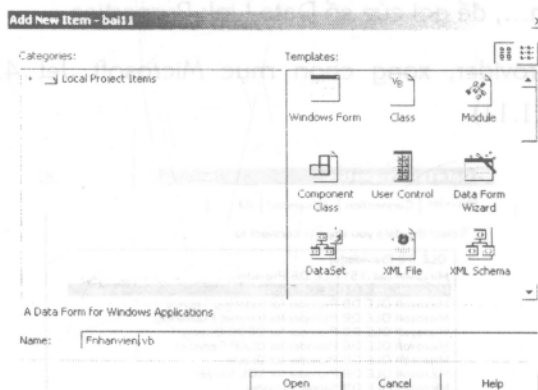
- Chọn Project types = *Visual Basic Projects*, xong nhấp biểu tượng *Windows Application* trong mục Templates.
- Xong nhấp OK.

VB.NET sẽ tự động khởi tạo 1 project mới với 1 form trống mặc định là Form1.vb chứa trong thư mục BAI11, là thư mục con của C:\BTVBNET.

Bước 3. Tạo form FNhanvien.vb dùng cập nhật dữ liệu table DMNhanvien.

Chọn lệnh View ➡ Solution Explorer kích hoạt của sổ Solution Explorer nếu chưa thấy.

Chọn lệnh Project ➡ Add Windows form..... Trong cửa sổ Add New Item (Hình 11.9), nhấp biểu tượng Data Form Wizard.



Hình 11.9 Hộp thoại Add New Item.

Nhập vào tên form Name = FNhanvien.vb, xong nhấp Open.

Nhấp Next trong cửa sổ Data Form Wizard.

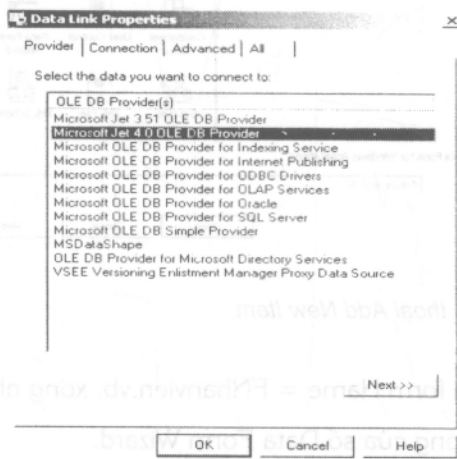
Nhập tên điều khiển DataSet mới là Dnhanvien với tùy chọn Create a new dataset named, xong nhấp Next (Hình 11.10).



Hình 11.10. Cửa sổ Data Form Wizard.

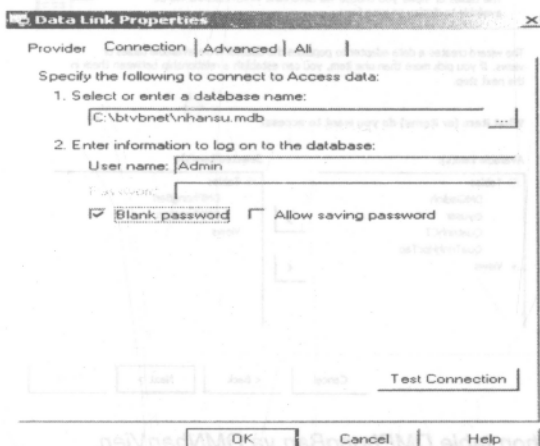
Chọn cơ sở dữ liệu Access cần kết nối dữ liệu bằng cách nhấp nút New Connection..., để gọi cửa sổ Data Link Properties.

Chọn tab Provider, xong chọn mục Microsoft Jet 4.0 OLE DB Provider (Hình 11.11).



Hình 11.11 Cửa sổ Data Link Properties – trang Provider.

Chọn tab Connection, nhấn nút ... ở phía sau text box Select or enter a database name để chọn cơ sở dữ liệu kết nối (Hình 11.12). Xong nhấn Test Connection, nếu thấy thông báo Test connection succeeded, xem như bạn đã thành công. Nhấp OK để trở lại cửa sổ Data Form Wizard, xong nhấn Next.



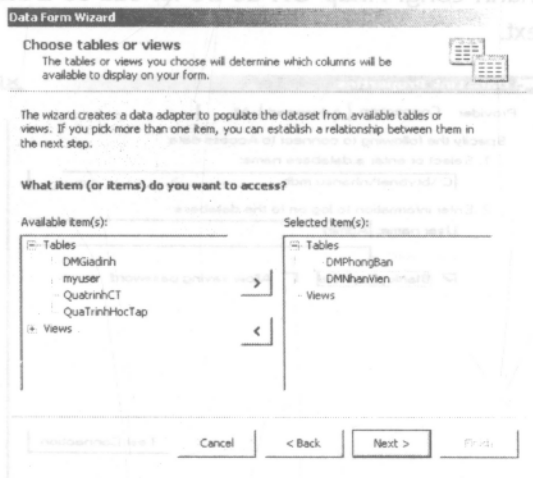
Hình 11.12 Cửa sổ Data Link Properties – trang Connection.

- Danh sách các table và query trong cơ sở dữ liệu Nhansu.mdb sẽ được liệt kê trong cửa sổ Data Form Wizard (Hình 11.13).



Hình 11.13 Danh sách table và query của Nhansu.mdb trong cửa sổ Data Form Wizard.

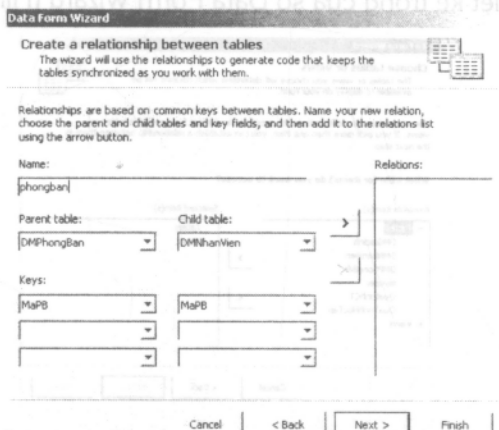
Lần lượt chọn table DMPHONGBAN, DMNHANVIEN trong mục Available item(s), nhấn nút ➡ để đưa sang mục Selected Item(s) (Hình 11.14). Xong nhấn Next.



Hình 11.14 Chọn table DMPHONGBAN và DMNHANVIEN.

- Tạo mối quan hệ giữa 2 table DMPHONGBAN với DMNHANVIEN.

Định thông tin quan hệ như Hình 11.15:



Hình 11.15 Cửa sổ tạo quan hệ giữa 2 table.

Name = Phongban (tên quan hệ sẽ tạo)

Parent table = DMPhongban, Keys = Mapb (field khóa quan hệ)

Child table = DMNhanvien, Keys = Mapb (field khóa quan hệ)

Nhấp button ➡ để đưa sang mục Relations như Hình 11.16. Xong nhấp Next.

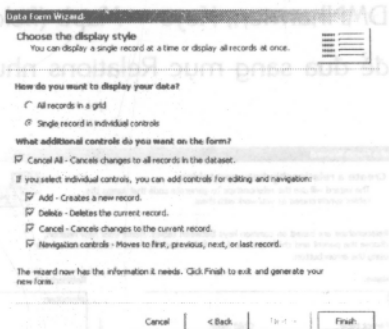
Hình 11.16 Mỗi quan hệ Phongban vừa tạo giữa 2 table DMPhongBan với DMNhanVien.

Đánh dấu chọn tất cả các field trong table DMPhongBan và DMNhanVien (Hình 11.17), xong tiếp tục nhấp Next.

Hình 11.17 Chọn các field trong table DMPhongBan và DMNhanVien.

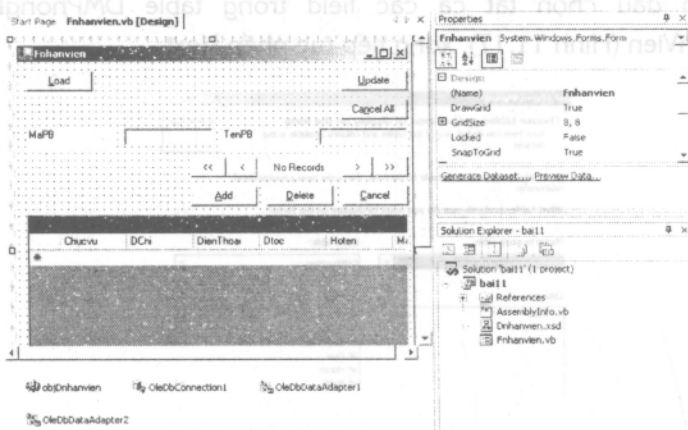
Trong cửa sổ chọn kiểu trình bày của Data Form Wizard, chọn tùy chọn:

- Single record in individual controls (Hình 11.18). Xong nhấn Finish.



Hình 11.18 Chọn kiểu trình bày form.

- Loại bỏ form trống (Form1.vb) khỏi project bằng cách nhấp nút phải mouse lên Form1 trong cửa sổ Solution Explorer, xong chọn lệnh Delete.
- Kết quả của sổ thiết kế form *FNhanvien* sẽ như Hình 11.19.



Hình 11.19 Cửa sổ thiết kế form *FNhanvien*.

Đặt form *FNhanvien* làm form khởi động bằng cách nhấp nút phải mouse lên project *Bai11* trong cửa sổ Solution Explorer, chọn lệnh Properties, định Startup Object = *FNhanvien*.

Bước 4. Chạy thử chương trình để xem kết quả.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start chạy thử chương trình. Ban đầu, bạn sẽ được một màn hình trống. Nhấp nút Load thông tin phòng ban đầu tiên và các nhân viên trong phòng sẽ hiển thị trên form như Hình 11.20. Nhấp nút X ở góc trên bên phải của sổ Form để kết thúc chương trình.

The screenshot shows a Windows form titled "FNhanvien". It contains several buttons: "Load", "Update", "Cancel All", "Add", "Delete", and "Cancel". Below these buttons is a table with the following data:

Chucvu	DChi	DienThoai	Dtcc	Hoten	Mi
GR Nhân số	19 Huang Vg	8356153	Kinh	Nguyễn Ngo	N0

Below the table, there is a large empty rectangular area, likely for additional data or a list.

Hình 11.20 Màn hình form *FNhanvien* khi vừa thi hành.

Bước 5. Điều chỉnh form *FNhanvien.vb* vừa tạo.

Form *FNhanvien* vừa tạo còn một số điểm bạn chưa vừa ý, chẳng hạn:

- Định lại kích thước form phóng lớn toàn màn hình.
- Dữ liệu chưa hiển thị được tiếng Việt.

- Bạn muốn điều chỉnh vị trí của các điều khiển trên form và thêm nút Close để kết thúc chương trình.
- Bạn muốn điều chỉnh trật tự các cột thông tin nhân viên, cũng như độ rộng của từng cột.

5a. Chỉnh sửa font chữ.

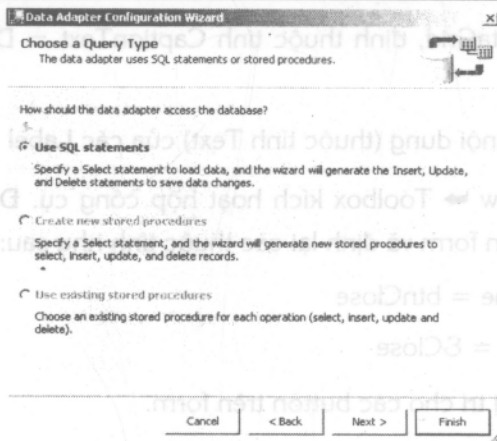
- Nhấp đúp lên form FNhanvien.vb trong cửa sổ Solution Explorer để trở lại giao diện thiết kế form.
- Định lại các thuộc tính của form như sau:
 - Text = Cap nhat nhan vien (*tiêu đề form*)
 - FormBorderStyle = Sizable (*kiểu viền form*)
 - BackColor = ControlLightLight (*màu nền của form*)
 - Name = FNhanvien (*tên form*)
 - WindowState = Maximized (*form phóng to toàn màn hình khi thi hành*)
 - Size = 812, 477 (*kích thước form lúc thiết kế với màn hình có độ phân giải 800x640*)
 - Localizable = False (*không thể định lại kích thước form khi đang thi hành*)
 - Opacity = 100% (*form hiển thị rõ ràng như khi thiết kế, không có độ trong suốt*)
- Định lại kích cỡ DataGrid grdDMNhanvien trong form với mouse.
- Lần lượt chọn các Textbox, Label và DataGrid và định lại font chữ cho chúng là VNI-Times.

- Chọn DataGridView, định thuộc tính CaptionText = Danh sách nhân viên.
- Chính lại nội dung (thuộc tính Text) của các Label
- Chọn View ➡ Toolbox kích hoạt hộp công cụ. Đặt 1 điều khiển Button lên form và định lại các thuộc tính như sau:
 - Name = btnClose
 - Text = &Close
- Định lại vị trí cho các button trên form.
- Nhấp đúp lên button Close để mở cửa sổ Code và nhập vào lệnh sau:

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

5b. Điều chỉnh DataGridView.

- Để điều chỉnh trật tự các cột thông tin nhân viên trong DataGridView:
 - Nhấp điều khiển OleDbDataAdapter2, xong chọn lệnh Data ➡ Configure Data Adapter..., nhấp Next.
 - Chọn Data Connection = Access.C:\Btvbnet\nhansu.mdb .Admin, nhấp Next.
 - Chọn tùy chọn Use SQL statements trong cửa sổ Choose Query Type (Hình 11.21), xong nhấp Next.



Hình 11.21 Cửa sổ Choose Query Type.

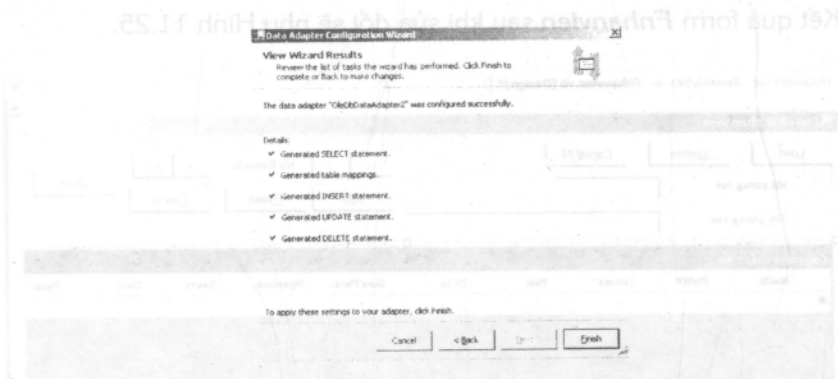
- Điều chỉnh thứ tự tên field trong câu lệnh SQL trong Hình 11.22 để được câu lệnh sau:

```
SELECT MaPB, Manv, Hoten, Phai, DChi, DienThoai, Ngaysinh,
      Chucvu, Dtoc, Tgiaio, Ngquan, Noisinh, Quoctich, SoCMND,
      Ngaycap, Noicap FROM DMNhanVien
```



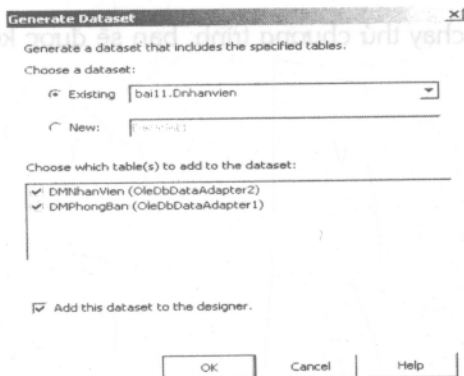
Hình 11.22 Câu lệnh SQL trước khi chỉnh sửa.

- Nhấp Next. Cuối cùng nhấp Finish đóng cửa sổ View Wizards Results (Hình 11.23).



Hình 11.23 Cửa sổ View Wizards Results.

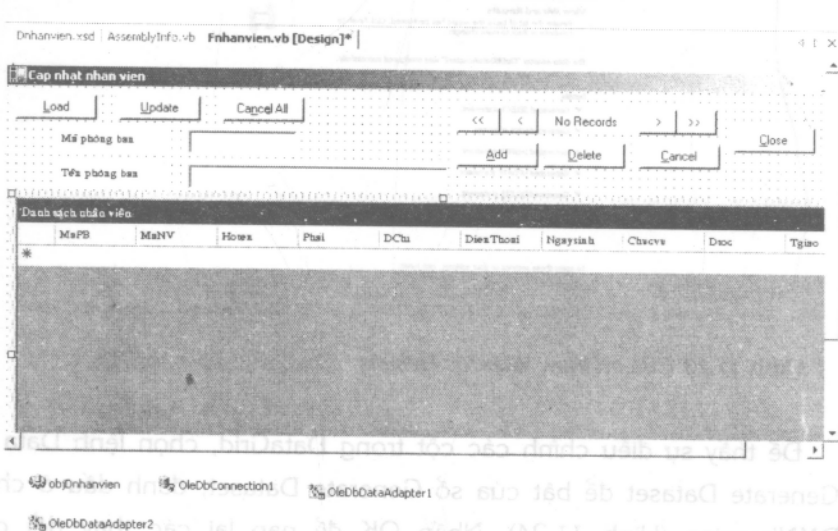
Để thấy sự điều chỉnh các cột trong DataGrid, chọn lệnh Data ➔ Generate Dataset để bật cửa sổ Generate Dataset, đánh dấu ô chọn DMNhanvien (Hình 11.24). Nhấp OK để nạp lại các thay đổi cho Dataset.



Hình 11.24 Cửa sổ Generate Dataset.

Chú ý: Không được đánh dấu ô chọn DMPhongBan, nếu không quan hệ 1 – nhiều giữa 2 table sẽ bị mất.

Kết quả form *Fnhanvien* sau khi sửa đổi sẽ như Hình 11.25.



Hình 11.25 Form *Fnhanvien* sau khi chỉnh sửa.

Biên dịch và chạy thử chương trình, bạn sẽ được kết quả như mong muốn.



Bài 12.

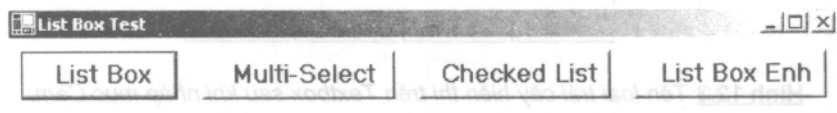
Chương trình ListBox

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với các loại Listbox và cách sử dụng chúng, biết cách nạp và lấy giá trị từ Listbox, cách tạo các Listbox cho chọn nhiều mục cùng lúc trong VB.Net. Điều này sẽ giúp bạn dễ hiểu và hoàn thành các bài thực hành tiếp sau hơn.

Hoạt động chương trình

Khi thi hành, form chính chương trình sẽ được nạp như Hình 12.1.

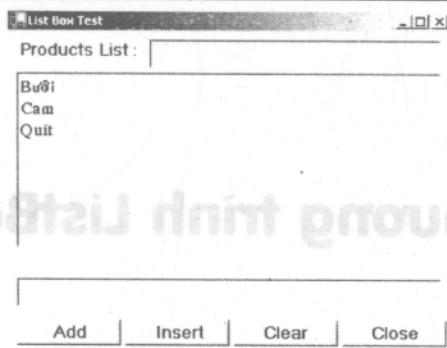


Hình 12.1 Chương trình Listbox khi vừa thi hành

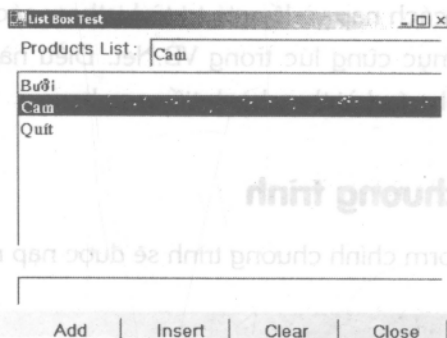
Khi người dùng nhấp nút List Box, form *frmListTest* sẽ được nạp như Hình 12.2.

Form này đầu tiên sẽ hiển thị 3 loại trái cây trong danh sách.

Khi nhấp tên 1 loại trái cây trong danh sách, tên sản phẩm sẽ hiển thị trên Textbox bên phải mục Products List (Hình 12.3).



Hình 12.2 Màn hình form frmListTest khi nhấp nút “List Box”.

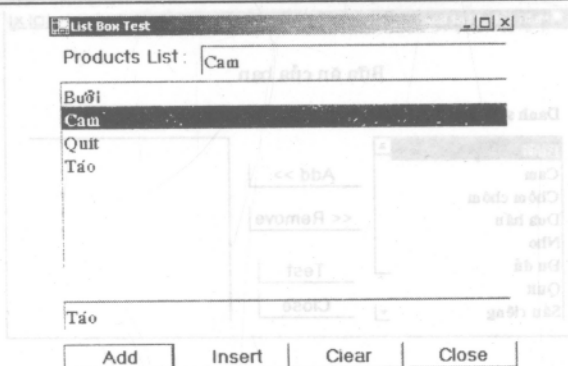


Hình 12.3 Tên loại trái cây hiển thị trên Textbox sau khi nhấp mục Cam.

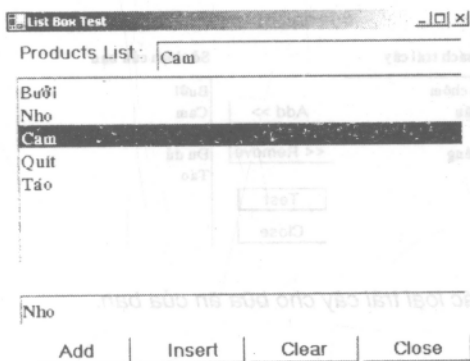
Người dùng cũng có thể thêm các loại trái cây mới vào Textbox bên dưới, xong nhấp Add để chèn vào cuối danh sách (Hình 12.4) hoặc nhấp Insert để chèn loại trái cây mới vào trước loại trái cây đang được chọn trong danh sách (Hình 12.5).

Người dùng cũng có thể xóa hết nội dung Listbox và nhập lại bằng cách nhấp nút Clear.

Nhấp Close để đóng form frmListTest trở lại màn hình chính.



Hình 12.4 Bổ sung loại trái cây “Táo” vào cuối danh sách.

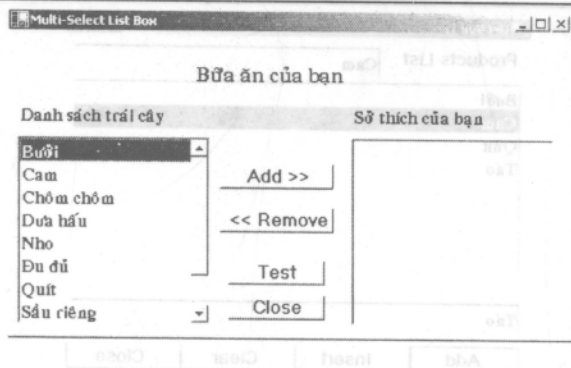


Hình 12.5 Chèn loại trái cây “Nho” vào danh sách trước mục đang chọn là

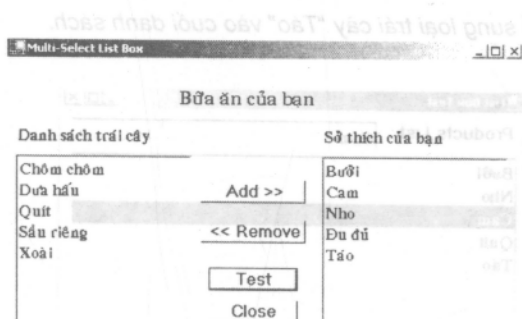
“Cam”.

Khi người dùng nhấn nút Multi Select, form *frmMultiList* sẽ được nạp như Hình 12.6.

Form này đầu tiên sẽ hiển thị danh sách 10 loại trái cây trong 1 Listbox để người dùng chọn dùng vào cuối bữa ăn của mình, rồi nhấn Add để đưa qua Listbox “Sở thích của bạn” (Hình 12.7).



Hình 12.6 Màn hình form frmMultiList khi nhấn nút “Multi Select”.



Hình 12.7 Chọn các loại trái cây cho bữa ăn của bạn.

Người dùng có thể nhấn nút **Test** để kiểm tra xem việc chọn của mình có phù hợp không, với qui ước là nếu chọn quá 5 loại trái cây, form sẽ hiển thị thông báo “Coi chừng bội thực”. Nếu tổng số điểm của các loại trái cây bạn chọn lớn hơn 19 thì form hiển thị thông báo “Nên đổi 1 vài loại trái cây khác”, ngược lại thông báo “Bữa ăn của bạn rất ngon” sẽ hiển thị (Hình 12.8). Thang điểm từng loại trái cây cụ thể như sau:

Cam = 1, Nho = 2, Đu đủ = 3, Bưởi = 4, Táo = 5, Quýt = 6,

Dưa hấu = 7, Xoài = 8, Sầu riêng = 9, Chôm chôm = 10.

ListBoxBasics [X]

Bữa ăn của bạn rất ngon!

OK

Hình 12.8 Hộp thông báo kết quả chọn.

GHI CHÚ

- ♦ Việc tính điểm các loại trái cây đều hoàn toàn dựa trên ý thích riêng của tác giả, mục đích chỉ để minh họa rõ ứng dụng của Multi Listbox. Vì thế độc giả có thể thay đổi thang điểm cho phù hợp với yêu cầu thực tiễn hơn, chẳng hạn tính điểm dựa trên nguồn vitamin, đặc tính nóng hay mát của từng loại trái cây,

Người dùng có thể loại bỏ một hay nhiều loại trái cây đã chọn khỏi danh sách *Sở thích của bạn* bằng cách chọn nó rồi nhấn nút **Remove**.

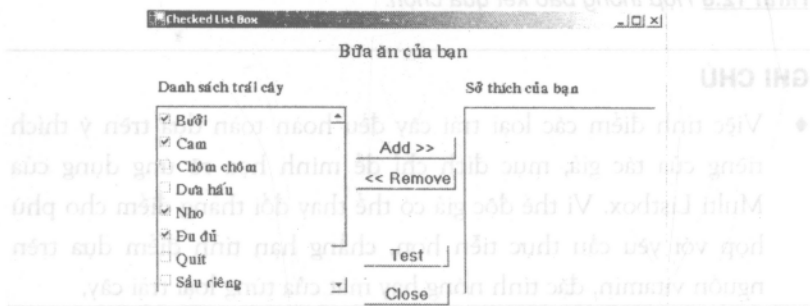
Có thể chọn nhiều loại trái cây cùng một lúc bằng cách kết hợp việc nhấp mouse với việc nhấn giữ phím **Ctrl** hay **Shift**.

Nhấp nút **Close** đóng form `frmMultiList` và trở lại form chính.

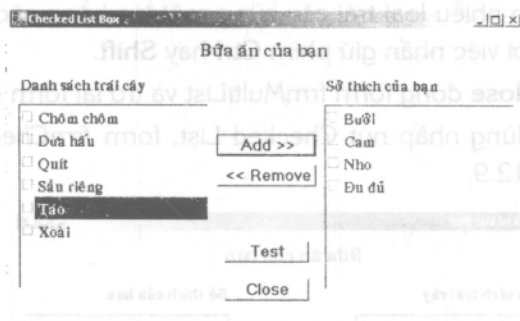
Khi người dùng nhấp nút **Checked List**, form `frmCheckList` sẽ được nạp như Hình 12.9.

Hình 12.9 Màn hình form `frmCheckList` khi nhấp nút “Checked List”.

Đây là 1 dạng thiết kế khác của form frmMultiList, hoạt động và ứng dụng của form tương tự như form frmMultiList. Điểm khác biệt duy nhất là để chọn nhiều loại trái cây, người dùng không cần phải giữ phím Ctrl hay Shift nữa, mà cứ từ từ nhấp vào các check box trước mục chọn (Hình 12.10), xong nhấp nút Add để chuyển qua Listbox bên phải (Hình 12.11).



Hình 12.10 Chọn các loại trái cây theo ý người dùng.



Hình 12.11 Sau khi nhấp Add, các loại trái cây đã chọn chuyển sang Listbox bên phải.

Khi người dùng nhấp nút List Box Enh, form frmListEnh sẽ được nạp như Hình 12.12.

Product Information	
Cam	
Nhỏ	
Bưởi	
Product ID	
Product Name	
Unit Price	
Units In Stock	
Close	

Hình 12.12 Màn hình form frmListEnh khi nhấp nút “List Box Enh”.

Khi nhấp mục “cam” trong danh sách, thông tin mặt hàng “Cam” như mã số, tên mặt hàng, đơn giá, số lượng tồn kho (trái) sẽ hiển thị trên các textbox bên dưới (Hình 12.13).

Product Information	
Cam	
Nhỏ	
Bưởi	
Product ID	1
Product Name	Cam
Unit Price	10.95
Units In Stock	100
Close	

Hình 12.13 Hiển thị thông tin loại trái cây được chọn ra các Textbox.

Người dùng có thể dùng phím mũi tên để chuyển đến các mục khác, thông tin của mục vừa chuyển đến cũng sẽ hiển thị trên các Textbox.

Nhân xét: Form minh họa một khả năng mạnh của điều khiển Listbox trong VB.Net là có thể nhận vào các đối tượng (object), chứ không chỉ có khả năng nhận từng giá trị như trong VB6. Đây là một khả năng rất có ích, giúp đơn giản công việc thiết kế các form tra cứu, cập nhật dữ liệu trong các ứng dụng cơ sở dữ liệu mà bạn sẽ gặp trong các bài thực hành tới.

- Nhấp nút Close ở góc trên bên phải form chính để kết thúc chương trình.

Hướng dẫn thiết kế

Bước 1. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File ➡ New Project để gọi cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = **ListBox.vbproj**
- Định thư mục lưu project: Location = **C:\BTVBNET**
- Nhấp tùy chọn Close Solution.
- Chọn Project types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.
- Xong nhấp OK.

VB.NET sẽ tự động khởi tạo 1 project mới với 1 form trống mặc định là Form1.vb chứa trong thư mục **Listbox**, là thư mục con của C:\BTVBNET.

Bước 2. Thiết kế form chính – frmMain.

Chọn Form1 và định lại các thuộc tính sau trong cửa sổ Properties:

Name = FrmMain

Text = List Box Test

Size = 560, 77

FormBorderStyle = Sizable

Enabled = False

- StartPosition = Manual

Lần lượt đặt lên form 4 button và định lại các thuộc tính của chúng như sau:

- Button 1:

Name = btnList

Textbox = List Box

Size = 104, 32

Position = 8, 8

TextAlign = MiddleCenter

- Button 2:

Name = btnMulti

Textbox = Multi-Select

Size = 128, 32

Position = 128, 8

TextAlign = MiddleCenter

- Button 3:

Name = btnChecked

Textbox = Checked List

Size = 128, 32

Position = 272, 8

TextAlign = MiddleCenter

- Button 4:

Name = btnLBoxEnh

Textbox = List Box Enh

Size = 128, 32

Position = 416, 8

TextAlign = MiddleCenter

- Viết mã lệnh: chọn lệnh View > Code

Public Class frmMain

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

' Vùng mã lệnh do VB.Net tự phát sinh trong quá trình thiết kế form

.....

#End Region

' Khi nhấp nút Checked List

Private Sub btnChecked_Click(ByVal sender As Object, ByVal e As

System.EventArgs) Handles btnChecked.Click

Dim frm As frmCheckList = New frmCheckList ()

frm.Show() *' nạp form frmCheckList*

End Sub

' Khi nhấp nút Multi_Select

Private Sub btnMulti_Click(ByVal sender As Object, ByVal e As

System.EventArgs) Handles btnMulti.Click

Dim frm As frmMultiList = New frmMultiList ()

frm.Show()

End Sub

' Khi nhấp nút List Box Enh

```
Private Sub btnLBoxEnh_Click(ByVal sender As Object, ByVal e As
```

```
System.EventArgs) Handles btnLBoxEnh.Click
```

```
Dim frm As frmListEnh = New frmListEnh ()
```

```
frm.Show()
```

End Sub

' Khi nhấp nút List Box

```
Private Sub btnList_Click(ByVal sender As Object, ByVal e As
```

```
System.EventArgs) Handles btnList.Click
```

```
Dim frm As frmListTest = New frmListTest()
```

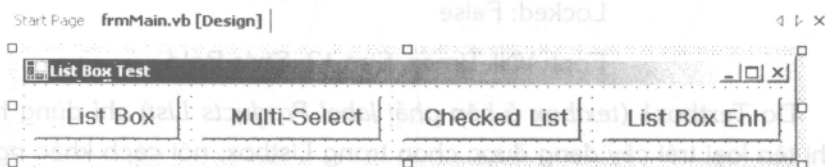
```
frm.Show()
```

End Sub

End Class

Lưu form với tên frmMain.

Kết quả thiết kế form **frmMain** như Hình 12.14.



Hình 12.14 Màn hình thiết kế form frmMain.

Bước 3. Thiết kế form frmListTest.

- Chọn lệnh Project ➡ Add Windows Form. Trong cửa sổ Add New Item, nhấp biểu tượng Windows Form trong mục Templates. Nhập tên form mới là **frmListTest** trong mục Name, xong nhấp Open.

- Định lại một số thuộc tính cho form frmListTest:

Name: frmListTest

Text: List Box Text

- Đặt 1 Label lên form và định lại các thuộc tính như sau:

Name: Label1

BackColor: ActiveCaptionText

Text: *Products List*:

- Đặt 2 Textbox lên form và định lại các thuộc tính như sau:

Textbox 1: Name: txtProduct

Text: (trống)

Locked: True

Font: VNI-Times, Size 12, Style Bold

Textbox 2: Name: txtName

Text: (trống)

Locked: False

Font: VNI-Times, Size 12, Style Bold

Do Textbox1 (textbox ở bên phải label *Products List*) chỉ dùng hiển thị tên loại trái cây đang được chọn trong Listbox, nói cách khác người dùng chỉ được xem chứ không được quyền thay đổi nội dung Textbox1, nên định thuộc tính Locked: True.

- Đặt 1 Listbox lên form và định lại các thuộc tính như sau:

Name: lstNames

Sorted: True

BackColor: Window

Font: VNI-Times, Size 12, Style Bold

SelectionMode: One (*chỉ cho chọn 1 mục tại mỗi thời điểm*)

- Đặt 4 Button lên form và định lại các thuộc tính như sau:

Button 1: Name: btnAdd

Text: Add

TextAlign = MiddleCenter

Button 2: Name: btnInsert

Text: Insert

TextAlign = MiddleCenter

Button 3: Name: btnClear

Text: Clear

TextAlign = MiddleCenter

Button 4: Name: btnClose

Text: &Close

TextAlign = MiddleCenter

- Viết mã lệnh xử lý form frmListTest:

Public Class frmListTest

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

'Phần mã lệnh do Windows Form Designer tự phát sinh

'Bạn không cần điều chỉnh gì trong đoạn lệnh này

.....

#End Region

' Khi nhấp nút Clear

Private Sub btnClear_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnClear.Click

lstNames.Items.Clear() *' Xóa trống nội dung Listbox*

txtproduct.Text = "" *' và Textbox hiển thị loại trái cây đang chọn*

End Sub*' Khi nhấp nút Add*

Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnAdd.Click

' Nếu chưa nhập tên loại trái cây mới trong textbox txtName

If txtName.Text = "" Then

MsgBox("Please Input New Product") *' Hiển thị hộp thông báo*Else *' ngược lại*lstNames.Items.Add(txtName.Text) *' Thêm loại trái cây mới
' vào danh sách*

End If

End Sub

' Khi nhấp nút Insert

Private Sub btnInsert_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnInsert.Click

Dim i As Integer

i = 0

' Nếu chưa nhập tên loại trái cây mới trong textbox txtName

If txtName.Text = "" Then

MsgBox("Please Input New Product") *' Hiển thị hộp thông báo*Else *' Ngược lại**' Vòng lặp xác định loại trái cây đang được chọn trong danh sách*

For i = lstNames.Items.Count - 1 To 0 Step -1

If lstNames.GetSelected(i) Then

Exit For

End If

Next

' ListBox không tự sắp xếp lại các mục trong danh sách

' nhằm đảm bảo mục được chèn vào nằm đúng vị trí của nó

lstNames.Sorted = False

' Nếu chưa có mục nào được chọn, thì xem mục đầu tiên

' trong danh sách mặc định được chọn

If i < 0 Then

i = 0

End If

' Chèn nội dung mới vào vị trí của mục đang chọn

lstNames.Items.Insert(i, txtName.Text)

' Đặt mục đang chọn là nội dung mới chèn vào danh sách

lstNames.SetSelected(i, True)

End If

End Sub

' Khi nhấp chọn một mục trong danh sách lstNames

Private Sub lstNames_SelectedIndexChanged(ByVal sender As Object,

ByVal e As System.EventArgs) Handles lstNames.SelectedIndexChanged

' Gán tên loại trái cây của mục được chọn cho text box txtProduct

txtproduct.Text = lstNames.Items(lstNames.SelectedIndex).ToString()

End Sub

' Khi form frmListTest được nạp

Private Sub frmListTest_Load(ByVal sender As System.Object, ByVal e

As System.EventArgs) Handles MyBase.Load

' Tạo mảng gồm 3 phần tử

Dim astrValues() As String = {"Cam", "Quýt", "Bưởi"}

' Xóa trống danh sách lstNames

lstNames.Items.Clear()

' Chèn nội dung mảng astrValues vào danh sách

lstNames.Items.AddRange(astrValues)

End Sub

' Khi nhấp nút Close hay nhấn tổ hợp phím Alt+C

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

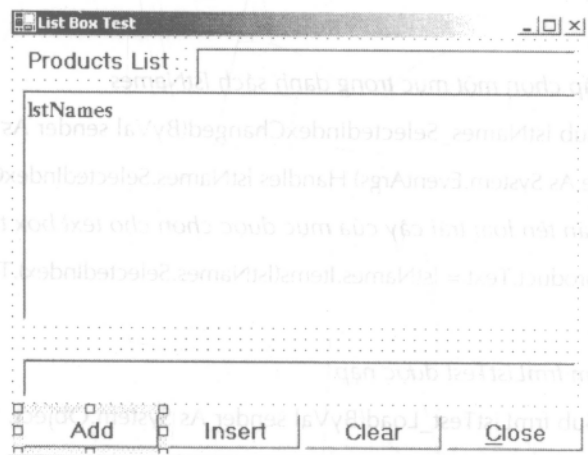
Me.Close() ' Đóng form frmListTest, trở lại form chính

End Sub

End Class

Kết quả thiết kế form frmListTest sẽ như Hình 12.15.

Start Page frmListTest.vb [Design] | frmListTest.vb |



Hình 12.15 Màn hình thiết kế form frmListTest.

Bước 4. Thiết kế form frmMultiList

- Chọn lệnh Project ➡ Add Windows Form. Trong cửa sổ Add New Item, nhấp biểu tượng Windows Form trong mục Templates. Nhập tên form mới là **frmMultiList** trong mục Name, xong nhấp Open.

- Định lại một số thuộc tính cho form frmMultiList:

Name: frmMultiList

Text: Multi-Select List Box

- Đặt 3 Label lên form và định lại các thuộc tính như sau:

Label 1: Name: Label1

Text: *Bữa ăn của bạn*

Font: VNI-Times, size 14pt, Style Bold

Label 2: Name: Label2

Text: *Danh sách trái cây*

Font: VNI-Times, size 12pt, Style Bold

Label 3: Name: Label3

Text: *Sở thích của bạn*

Font: VNI-Times, size 12pt, Style Bold

- Đặt 2 Listbox lên form và định lại các thuộc tính như sau:

Listbox 1 (Listbox nằm bên dưới label “Danh sách trái cây”):

Name: lstUnSelected

TabIndex: 0

Listbox 2 (Listbox nằm bên dưới label “Sở thích của bạn”):

Name: lstSelected

TabIndex: 1

Cả 2 Listbox đều có cùng các thuộc tính sau:

BackColor: Window ‘ màu nền Listbox

ForeColor: WindowText ‘ màu chữ trong Listbox

SelectionMode: MultiExtended ' cho chọn nhiều mục cùng lúc

Sorted: True ' Danh sách mục tự động sắp xếp theo abc

Font: VNI-Times, Size 12, Style Bold

Locked: False ' không khóa các mục chọn trong Listbox

Enabled: True ' cho phép truy cập Listbox

- Đặt 4 Button lên form và định lại các thuộc tính như sau:

Button 1: Name: btnAdd

Text: Add >>

TextAlign = MiddleCenter

TabIndex = 2

Button 2: Name: btnRemove

Text: << Remove

TextAlign = MiddleCenter

TabIndex = 4

Button 3: Name: btnTest

Text: Test

TextAlign = MiddleCenter

TabIndex = 3

Button 4: Name: btnClose

Text: &Close

TextAlign = MiddleCenter

TabIndex = 5

- Viết mã lệnh xử lý form frmMultiList:

Public Class frmmultilist

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

'Phần mã lệnh do Windows Form Designer tự phát sinh

'Bạn không cần điều chỉnh gì trong đoạn lệnh này

.....

#End Region

'Thủ tục khởi tạo nội dung ban đầu cho Listbox lstUnSelected

Private Sub ListLoad()

' Khởi tạo mảng gồm 10 phần tử

Dim astrValues() As String = _

["Cam", "Xoài", "Quýt", "Chôm chôm", "Táo", "Bưởi", "Dưa hấu",

"Đu đủ", "Sầu riêng", "Nho"]

' Chèn nội dung mảng mới khởi tạo vào danh sách lstUnSelected

lstUnSelected.Items.AddRange(astrValues)

End Sub

' Khi nhấp nút Add >>

Private Sub btnAdd_Click(ByVal sender As Object, ByVal e As

System.EventArgs) Handles btnAdd.Click

' Khai báo biến nguyên intLoop để sử dụng trong vòng lặp For

Dim intLoop As Integer

' Vòng lặp xác định các mục đang được chọn trong lstUnSelected

' để chuyển qua danh sách bên phải lstSelected

'Duyệt tất cả các mục chọn trong danh sách lstUnSelected

'từ mục chọn cuối đến mục chọn đầu tiên

For intLoop = lstUnSelected.Items.Count - 1 To 0 Step -1

' Nếu mục chọn đang được chọn

If lstUnSelected.GetSelected(intLoop) Then

' Thêm nó vào danh sách bên phải lstSelected

lstSelected.Items.Add(lstUnSelected.Items(intLoop))

' Xóa nó khỏi danh sách lstUnSelected

lstUnSelected.Items.RemoveAt(intLoop)

End If

Next

End Sub

' Khi form MultiList được nạp

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

' Gọi thủ tục ListLoad để khởi tạo nội dung các danh sách trên form
ListLoad()

End Sub

' Khi nhấp nút Test

Private Sub btnTest_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnTest.Click

' Nếu số mục chọn trong danh sách lstSelected nhiều hơn 5
If lstSelected.Items.Count > 5 Then

MsgBox("Coi chừng bội thực !") *' Hiển thị thông báo*

Else *' ngược lại*

Dim i, diem As Integer

diem = 0

' Vòng lặp tính điểm các loại trái cây được chọn

For i = 0 To lstSelected.Items.Count - 1 Step 1

Select Case lstSelected.Items(i)

Case "Cam": diem = diem + 1

```

        Case "Nhỏ": diem = diem + 2
        Case "Đu đủ": diem = diem + 3
        Case "Bưởi": diem = diem + 4
        Case "Táo": diem = diem + 5
        Case "Quýt": diem = diem + 6
        Case "Dưa hấu": diem = diem + 7
        Case "Xoài": diem = diem + 8
        Case "Sầu riêng": diem = diem + 9
        Case "Chôm chôm": diem = diem + 10
    End Select
Next
' Nếu điểm nhỏ hơn hoặc bằng 3
If diem <= 3 Then
    MsgBox("Nên thêm trái cây khác!") ' Hiển thị thông báo
ElseIf diem <= 19 Then ' Nếu điểm <=19 và >=4
    MsgBox("Bữa ăn của bạn rất ngon!")
Else ' Nếu điểm > 19
    MsgBox("Nên đổi 1 vài loại trái cây khác!")
End If
End If
End Sub

' Nếu nhấp nút << Remove
Private Sub btnremove_Click(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles btnremove.Click
    Dim intLoop As Integer
    ' Vòng lặp tìm và gỡ bỏ các mục đang chọn trong danh sách
    ' lstSelected
    ' và chuyển trở lại danh sách lstUnSelected
    ' Duyệt các mục trong danh sách lstSelected từ dưới lên trên
    For intLoop = lstSelected.Items.Count - 1 To 0 Step -1

```

' Nếu mục được chọn

If lstSelected.GetSelected(intLoop) Then

' Thêm nó vào danh sách UnSelected

lstUnSelected.Items.Add(lstSelected.Items(intLoop))

' Gỡ bỏ nó khỏi danh sách lstSelected

lstSelected.Items.RemoveAt(intLoop)

End If

Next

End Sub

' Khi nhấp nút Close

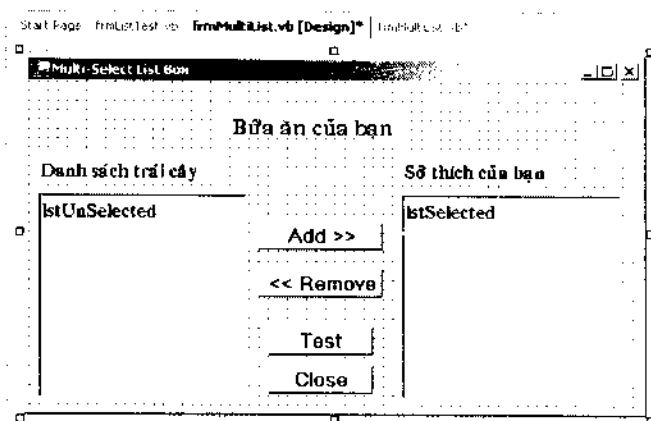
Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnClose.Click

Me.Close() ' Đóng form frmMultiList, trở lại form chính

End Sub

End Class

Kết quả màn hình thiết kế form frmMultiList sẽ như Hình 12.16.



Hình 12.16 Màn hình thiết kế form frmMultiList.

Bước 5. Thiết kế form frmCheckList.

- Chọn lệnh Project ➡ Add Windows Form. Trong cửa sổ Add New Item, nhấp biểu tượng Windows Form trong mục Templates. Nhập tên form mới là **frmCheckList** trong mục Name, xong nhấp Open.

- Định lại một số thuộc tính cho form frmCheckList:

Name: frmCheckList

Text: Checked List Box

- Đặt 3 Label lên form và định lại các thuộc tính như sau:

Label 1: Name: Label1

Text: *Bữa ăn của bạn*

Font: VNI-Times, size 14pt, Style Bold

Label 2: Name: Label2

Text: *Danh sách trái cây*

Font: VNI-Times, size 12pt, Style Bold

Label 3: Name: Label3

Text: *Sở thích của bạn*

Font: VNI-Times, size 12pt, Style Bold

- Đặt 2 điều khiển CheckedListbox lên form và định lại các thuộc tính như sau:

CheckedListbox 1 (nằm bên dưới label "*Danh sách trái cây*):

Name: clstUnSelected

TabIndex: 0

CheckedListbox 2 (nằm bên dưới label "*Sở thích của bạn*):

Name: clstSelected

TabIndex: 1

Cả 2 `CheckedListBox` đều có cùng các thuộc tính sau:

`BackColor`: `Window` ' màu nền trong `CheckedListBox`

`ForeColor`: `WindowText` ' màu chữ trong `CheckedListBox`

`Sorted`: `True` ' Danh sách mục tự động sắp xếp theo abc

`CheckOnClick`: `True` ' Đánh dấu ✓ khi nhấp mục chọn

`Font`: `VNI-Times`, `Size 12`, `Style Bold`

`Locked`: `False` ' không khóa các mục chọn trong `CheckedListBox`

`Enabled`: `True` ' cho phép truy cập `CheckedListBox`

- Đặt 4 `Button` lên form và định lại các thuộc tính như sau:

Button 1: `Name`: `btnAdd`

`Text`: `Add >>`

`TextAlign` = `MiddleCenter`

`TabIndex` = 2

Button 2: `Name`: `btnRemove`

`Text`: `<< Remove`

`TextAlign` = `MiddleCenter`

`TabIndex` = 4

Button 3: `Name`: `btnTest`

`Text`: `Test`

`TextAlign` = `MiddleCenter`

`TabIndex` = 3

Button 4: `Name`: `btnClose`

`Text`: `&Close`

`TextAlign` = `MiddleCenter`

`TabIndex` = 5

- Viết mã lệnh xử lý form frmCheckList:

Public Class frmCheckList

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

'Phần mã lệnh do Windows Form Designer tự phát sinh

'Bạn không cần điều chỉnh gì trong đoạn lệnh này

.....

#End Region

'Thủ tục khởi tạo nội dung ban đầu cho CheckedListbox clstUnSelected

Private Sub ListLoad()

Dim astrValues() As String = _

{"Cam", "Xoài", "Quít", "Chôm chôm", "Táo", "Bưởi", "Dưa hấu",

"Đu đủ", "Sầu riêng", "Nho"}

clstUnSelected.Items.AddRange(astrValues)

End Sub

' Khi nhấp nút Add >>

Private Sub btnAdd_Click(ByVal sender As Object, ByVal e As

System.EventArgs) Handles btnAdd.Click

Dim oltems As CheckedListBox.CheckedItemCollection

Dim intLoop As Integer

' Di chuyển các mục chọn từ clstUnselected sang clstSelected

oltems = clstUnSelected.CheckedItems

' Vòng lặp duyệt từng mục trong clstUnselected

For intLoop = 0 To oltems.Count - 1

' Thêm các mục đang được chọn trong clstUnSelected vào

' clstSelected


```
clstSelected.Items.Add(oltems(intLoop))
```

```
Next
```

```
oltems = clstUnSelected.CheckedItems
```

```
' Vòng lặp duyệt từng mục trong clstUnselected
```

```
For intLoop = oltems.Count - 1 To 0 Step -1
```

```
' Xóa các mục đang được chọn trong clstUnSelected
```

```
clstUnSelected.Items.Remove(oltems(intLoop))
```

```
Next
```

End Sub

```
' Khi form frmCheckList được nạp
```

```
Private Sub frmCheckList_Load(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles MyBase.Load
```

```
ListLoad()
```

End Sub

```
' Khi nhấp nút Test
```

```
Private Sub btnTest_Click(ByVal sender As System.Object, ByVal e As
```

```
System.EventArgs) Handles btnTest.Click
```

```
If clstSelected.Items.Count > 5 Then
```

```
MsgBox("Coi chừng bội thực !")
```

```
Else
```

```
Dim i, diem As Integer
```

```
diem = 0
```

```
For i = 0 To clstSelected.Items.Count - 1 Step 1
```

```
Select Case clstSelected.Items(i)
```

```
Case "Cam": diem = diem + 1
```

```
Case "Nho": diem = diem + 2
```

```

Case "Đu đủ": diem = diem + 3
Case "Bưởi": diem = diem + 4
Case "Táo": diem = diem + 5
Case "Quýt": diem = diem + 6
Case "Dưa hấu": diem = diem + 7
Case "Xoài": diem = diem + 8
Case "Sầu riêng": diem = diem + 9
Case "Chôm chôm": diem = diem + 10

```

End Select

Next

If diem <= 3 Then

 MsgBox("Nên thêm trái cây khác!")

ElseIf diem <= 19 Then

 MsgBox("Bữa ăn của bạn rất ngon!")

Else

 MsgBox("Nên đổi 1 vài loại trái cây khác!")

End If

End If

End Sub

' Khi nhấp nút << Remove

Private Sub bntremove_Click(ByVal sender As System.Object, ByVal e

As System.EventArgs) Handles bntremove.Click

 Dim oltems As CheckedListBox.CheckedItemCollection

 Dim intLoop As Integer

' Di chuyển các mục chọn từ clstSelected sang clstUnSelected

 oltems = clstSelected.CheckedItems

```

For intLoop = 0 To oltems.Count - 1
    clstUnSelected.Items.Add(oltems(intLoop))
Next
oltems = clstSelected.CheckedItems
For intLoop = oltems.Count - 1 To 0 Step -1
    clstSelected.Items.Remove(oltems(intLoop))
Next

```

End Sub

' Khi nhấp nút Close hay nhấn tổ hợp phím Alt+C

```

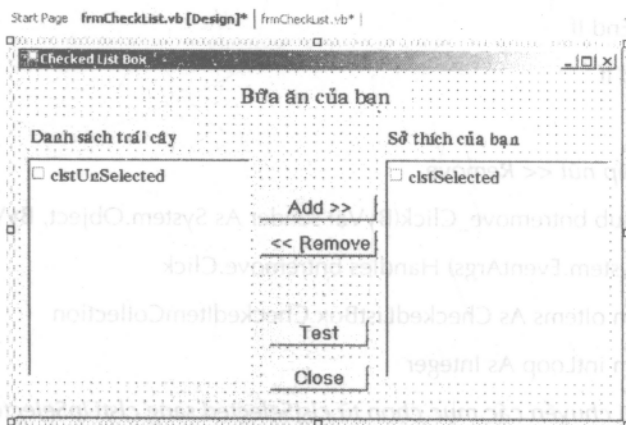
Private Sub bntclose_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles bntclose.Click
    Me.Close()

```

End Sub

End Class

Kết quả thiết kế form frmCheckList sẽ như Hình 12.17.



Hình 12.17 Màn hình thiết kế form frmCheckList.

Bước 6. Thiết kế form frmListEnh.

- Chọn lệnh Project ➡ Add Windows Form. Trong cửa sổ Add New Item, nhấp biểu tượng Windows Form trong mục Templates. Nhập tên form mới là **frmListEnh** trong mục Name, xong nhấp Open.

- Định lại một số thuộc tính cho form frmListEnh:

Name: frmListEnh

Text: Product Information

- Đặt 4 Label lên form và định lại các thuộc tính như sau:

Label 1: Name: Label1

Text: *Product ID*

Label 2: Name: Label2

Text: *Product Name*

Label 3: Name: Label3

Text: *Unit Price*

Label 4: Name: Label4

Text: *Units In Stock*

Cả 4 Label đều có cùng thuộc tính:

Font: Tahoma, size 12pt, Style Bold

- Đặt 1 điều khiển Listbox lên form và định lại các thuộc tính như sau:

Name: lstProducts

DisplayMember: ProductName ' Tên thuộc tính sẽ hiển thị trên
' Listbox của đối tượng chứa
' dữ liệu

TabIndex: 0

BackColor: Window * màu nền trong CheckedListBox

ForeColor: WindowText *màu chữ trong CheckedListBox*

SelectionMode: One 'Chí cho chon 1 muc

Font: VNI-Times, Size 12, Style Bold

Locked: False ' không khóa các mục chọn trong
' *CheckedListbox*

Enabled: True ' cho phép truy cập CheckedListBox

- Đặt 1 Button lên form và định lại các thuộc tính như sau:

Name: btnClose

Text: &Close

TextAlign = MiddleCenter

- Đặt 4 Textbox lên form và định lại các thuộc tính như sau:

Textbox 1: Name: txtProductID

TabIndex: 1

Textbox 2: Name: txtProductName

TablIndex: 2

Textbox 3: Name: txtUnitPrice

TabIndex: 3

Textbox 4: Name: txtUnitsInStock

TabIndex: 4

Các Textbox đều có cùng các thuộc tính sau:

Text: (để trống)

BackColor: Window

ForeColor: WindowText

Font: VNI-Times, Size 12, Style Bold

- Viết mã lệnh xử lý form *frmListEnh*:

‘ Định nghĩa lớp Product gồm 4 thuộc tính có thể đọc / ghi

‘ đó là *ProductID*, *ProductName*, *UnitPrice* và *UnitsInStock*

Public Class Product

‘ Khai báo các biến riêng của lớp

Private mintProductID As Integer

Private mstrProductName As String

Private mdecUnitPrice As Decimal

Private msrtUnitsInStock As Short

‘ Định nghĩa thuộc tính *ProductID*

Property ProductID() As Integer

Get

Return mintProductID

End Get

Set(ByVal Value As Integer)

mintProductID = Value

End Set

End Property

‘ Định nghĩa thuộc tính *ProductName*

Property ProductName() As String

Get

Return mstrProductName

End Get

Set(ByVal Value As String)

mstrProductName = Value

End Set

End Property

' Định nghĩa thuộc tính UnitPrice

Property UnitPrice() As Decimal

Get

Return mdecUnitPrice

End Get

Set(ByVal Value As Decimal)

mdecUnitPrice = Value

End Set

End Property

' Định nghĩa thuộc tính UnitsInStock

Property UnitsInStock() As Short

Get

Return msrtUnitsInStock

End Get

Set(ByVal Value As Short)

msrtUnitsInStock = Value

End Set

End Property

End Class

' Định nghĩa lớp frmListEnh (form frmListEnh)

Public Class frmListEnh

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

' Phần mã lệnh do Windows Form Designer tự phát sinh

' Bạn không cần điều chỉnh gì trong đoạn lệnh này

.....

#End Region

' Thù tục khởi tạo và nạp giá trị cho các đối tượng Product

Private Sub ListLoad()

Dim oProd As Product

' Khởi tạo 1 đối tượng Product mới

oProd = New Product()

' Gán các giá trị cho đối tượng vừa khởi tạo

With oProd

.ProductID = 1

.ProductName = "Cam"

.UnitPrice = CDec(10.95)

.UnitsInStock = 100

End With

' Thêm đối tượng vào Listbox lstProducts

lstProducts.Items.Add(oProd)

' Khởi tạo 1 đối tượng Product mới thứ 2

oProd = New Product()

' Gán các giá trị cho đối tượng vừa khởi tạo

With oProd

.ProductID = 2

.ProductName = "Nho"

.UnitPrice = CDec(5.54)

.UnitsInStock = 200

End With

' Tiếp tục thêm đối tượng vào Listbox lstProducts

lstProducts.Items.Add(oProd)

' Khởi tạo đối tượng Product mới thứ 3

oProd = New Product()

With oProd

.ProductID = 3

.ProductName = "Bưởi"

.UnitPrice = CDec(4.96)

.UnitsInStock = 250

End With

' Tiếp tục thêm đối tượng vào Listbox lstProducts

lstProducts.Items.Add(oProd)

End Sub

' Khi form frmListEnh được nạp

Private Sub frmListEnh_Load(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles MyBase.Load

Call ListLoad() *' gọi thủ tục ListLoad*

End Sub

' Khi mục thay đổi được chọn hiện hành trong Listbox

Private Sub lstProducts_SelectedIndexChanged(ByVal sender As

System.Object, ByVal e As System.EventArgs) Handles

lstProducts.SelectedIndexChanged

Dim oProd As Product

' Xác định mục (đối tượng Product) được chọn trong danh sách

oProd = CType(lstProducts.Items(lstProducts.SelectedIndex), Product)

' Nạp giá trị các thuộc tính của đối tượng được chọn ra các

' Textbox trên form

With oProd

txtProductID.Text = .ProductID.ToString()

txtProductName.Text = .ProductName

txtUnitPrice.Text = .UnitPrice.ToString()

txtUnitsInStock.Text = .UnitsInStock.ToString()

End With

End Sub

' Khi nhấp nút Close hay nhấn tổ hợp phím Alt+C

Private Sub bntclose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles bntclose.Click

Me.Close()

End Sub

End Class

MÁCH NƯỚC

- ♦ Trong 4 thuộc tính của đối tượng Product, chỉ có thuộc tính ProductName chứa giá trị chuỗi, còn lại đều là giá trị số. Vì thế cần dùng phương thức ToString để chuyển số thành chuỗi trước khi gán cho các Textbox.

Kết quả màn hình thiết kế form frmListEnh sẽ như Hình 12.18.

Hình 12.18 Màn hình thiết kế form frmListEnh.

Bước 7. Chạy thử chương trình để xem kết quả.

- Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.
- Chọn lệnh Build ➡ Build Solution để biên dịch project.
- Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start để chạy thử chương trình.

Bước 8. Thi hành chương trình dưới dạng tập tin EXE.

Để chạy chương trình ngoài Windows mà không cần khởi động VB.Net, chọn tập tin Listbox.exe trong thư mục C:\BTVBNET\Listbox\Bin.

Bài 13.

Chương trình Dialog

Mục đích chương trình

Trong bài thực hành trước để hiển thị các thông báo, chúng tôi đã sử dụng những hộp thông báo *Msgbox* mà *VB.Net* hỗ trợ sẵn. Tuy nhiên, việc sử dụng các hộp thông báo này có điểm bất tiện là khi bạn muốn hiển thị font tiếng Việt cần phải định lại font chữ tiếng Việt trong *Display Properties* của *Windows*, điều này làm ảnh hưởng đến các chương trình khác.

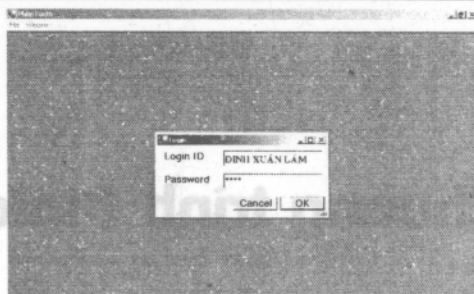
Bài thực hành này giúp bạn làm quen với cách tự tạo riêng cho mình các hộp thông báo trong *VB.Net*, đồng thời giúp bạn nắm được cách thiết kế các form dạng *MDI* (*Multiple-Document Interface*).

Hoạt động chương trình

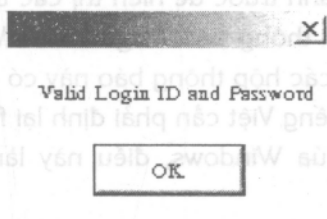
Khi thi hành, form chính chương trình sẽ được nạp nhưng người dùng chưa thể truy xuất được, form chính sẽ tự động gọi form *frmlogin* (hộp thoại tự tạo) và đợi người dùng nhập vào tên cùng mật mã login chương trình như Hình 13.1.

Nếu người dùng nhập tên và mật mã login qui định là "ĐINH XUÂN LÂM" và "XLTT", xong nhấp *OK*, thì thông báo "Valid Login ID and Password" sẽ hiển thị (Hình 13.2). Form login tự động đóng lại và menu chính sẽ hiện ra (Hình 13.3).

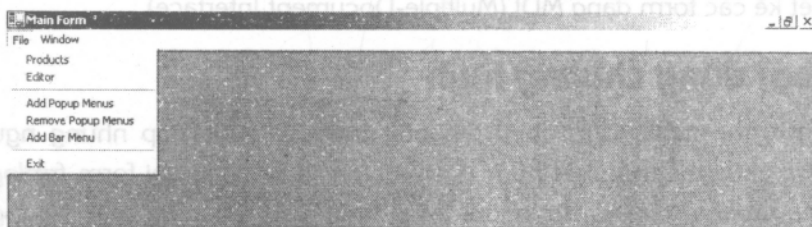
Nếu người dùng nhấp nút *Cancel* hoặc nhập sai tên mật mã login hệ thống rồi nhấp *OK*, form login sẽ đóng và chương trình kết thúc.



Hình 13.1 Chương trình Dialog khi vừa thi hành.



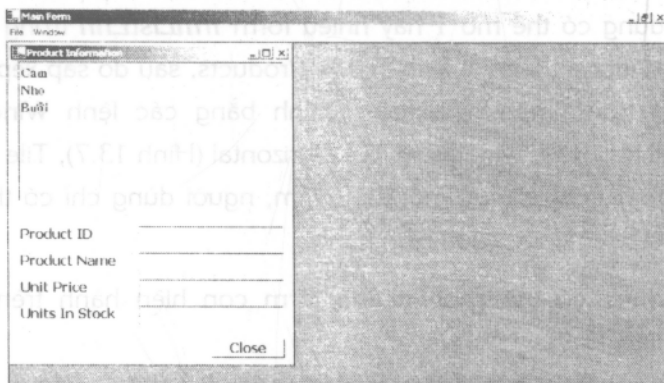
Hình 13.2 Hộp thông báo khi người dùng nhập đúng tên và mật mã.



Hình 13.3 Màn hình form chính sau khi login thành công.

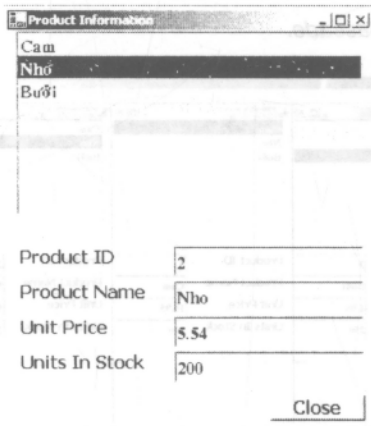
Chọn lệnh File ➤ Products, một form con *frmListEnh* sẽ hiển thị trong form chính (Hình 13.4).

Form này đầu tiên sẽ hiển thị tên 3 loại trái cây trong danh sách.



Hình 13.4 Form *frmListEnh* khi chọn lệnh *File* ➔ *Products*.

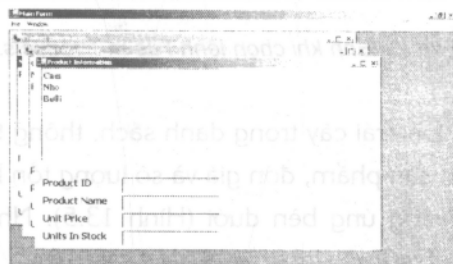
Khi nhấp tên 1 loại trái cây trong danh sách, thông tin về loại trái cây đó (như mã số, tên sản phẩm, đơn giá và số lượng tồn kho) sẽ được nạp lên các Textbox tương ứng bên dưới (Hình 13.5). Nhấp nút Close để đóng form con *frmListEnh*.



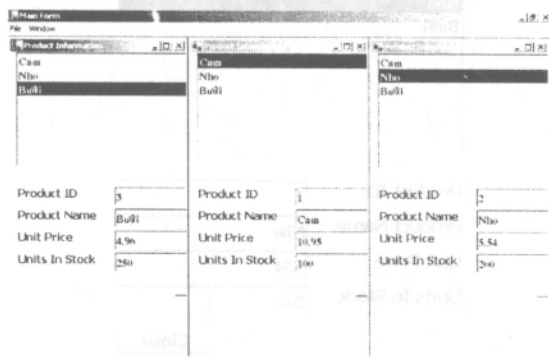
Hình 13.5 Kết quả form *frmListEnh* sau khi nhấp mục *Nho*.

Người dùng có thể mở 1 hay nhiều form *frmListEnh* trong cửa sổ chính của chương trình với lệnh File ➡ Products, sau đó sắp xếp lại các cửa sổ đó trong màn hình form chính bằng các lệnh Window ➡ Cascade (Hình 13.6), Window ➡ Tile Horizontal (Hình 13.7), Tile Vertical (Hình 13.8). Tuy nhiên, tại mỗi thời điểm, người dùng chỉ có thể thao tác trên mỗi form được chọn hiện hành.

Người dùng có thể chọn nhanh form con hiện hành trên menu Window (Hình 13.9).



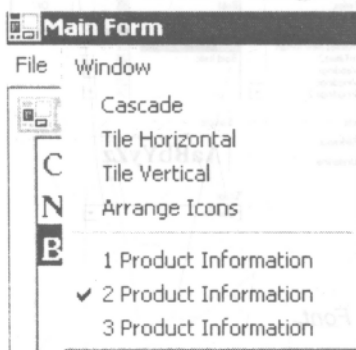
Hình 13.6 Kết quả form *frmListEnh* sau khi mở 3 cửa sổ con và chọn lệnh Window / Cascade.



Hình 13.7 Kết quả form *frmListEnh* sau khi mở 3 cửa sổ con và chọn lệnh Window / Tile Vertical.



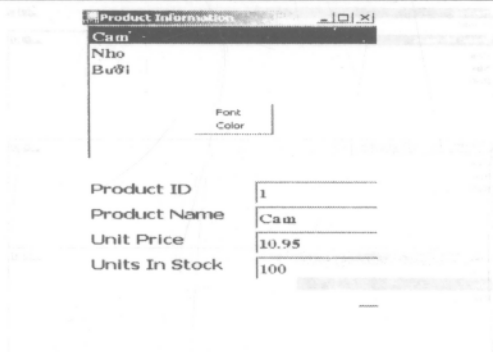
Hình 13.8 Kết quả form frmListEnh sau khi mở 3 cửa sổ con và chọn lệnh Window / Tile Horizontal.



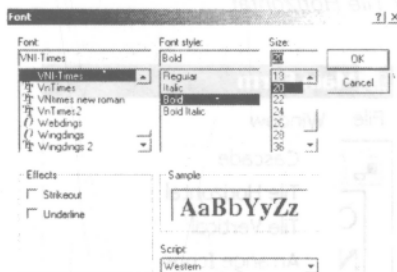
Hình 13.9 Menu Window sau khi mở 3 form con.

Khi nhấp nút phải mouse lên form hoặc 1 đối tượng bất kỳ trên form con frmListEnh, một menu nhanh gồm 2 lệnh Font và Color sẽ hiển thị (Hình 13.10).

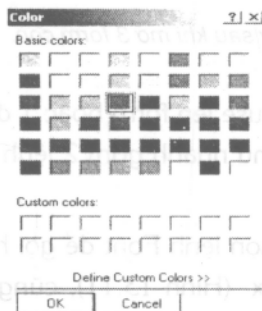
Người dùng có thể chọn lệnh Font để gọi hộp thoại Font, chọn font chữ cho nội dung Listbox (Hình 13.11), cũng như chọn lệnh Color để gọi hộp thoại Color (Hình 13.12), chọn màu nền cho form con frmListEnh hiện hành (Hình 13.13).



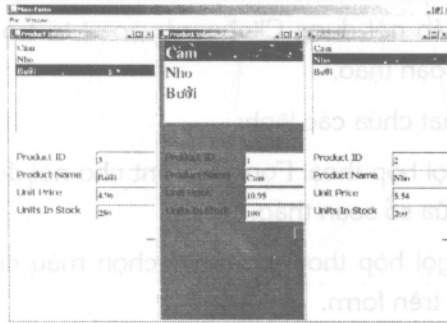
Hình 13.10 Menu nhanh khi nhấp nút phải mouse lên form con frmListEnh.



Hình 13.11 Hộp thoại Font.

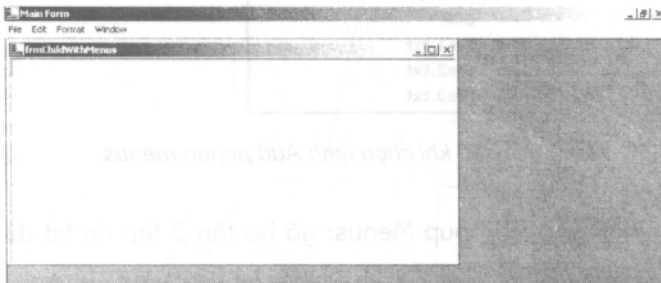


Hình 13.12 Hộp thoại Color.



Hình 13.13 Kết quả form con frmLishEnh sau khi chọn font và màu mới.

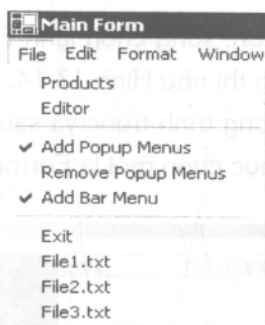
Giờ đây đóng các form con, xong chọn lệnh File ➡ Editor, một cửa sổ soạn thảo văn bản sẽ hiển thị như Hình 13.14. Để ý so sánh với Hình 13.13, menu chính của chương trình trước và sau khi chọn lệnh File ➡ Editor sẽ xuất hiện thêm 2 mục chọn mới là Format và Edit.



Hình 13.14 Cửa sổ soạn thảo văn bản được mở trong form chính.

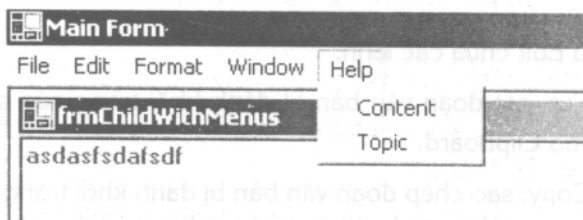
- Menu Edit chứa các lệnh:
 - Cut: cắt đoạn văn bản bị đánh khối trong cửa sổ soạn thảo vào Clipboard.
 - Copy: sao chép đoạn văn bản bị đánh khối trong cửa sổ soạn thảo vào Clipboard.

- Paste: dán nội dung Clipboard vào vị trí trỏ hiện hành trong cửa sổ soạn thảo.
- Menu Format chứa các lệnh:
 - Font: gọi hộp thoại Font đặt font cho nội dung đang hiển thị trong cửa sổ soạn thảo.
 - Color: gọi hộp thoại Color để chọn màu cho văn bản đang hiển thị trên form.
- Lệnh File ➡ Add Popup Menus: thêm danh sách 3 tập tin txt ở cuối menu Exit (Hình 13.15).



Hình 13.15 Menu File sau khi chọn lệnh Add popup menus.

- File ➡ Remove Popup Menu: gỡ bỏ tên 3 tập tin txt đã thêm ở cuối menu Exit.
- File ➡ Add Bar Menus: thêm menu bar Help (Hình 13.16).



Hình 13.16 Menu chương trình sau khi nhập lệnh Add Bar Menu.

- File ➡ Exit: đóng form chính và kết thúc chương trình.

Hướng dẫn thiết kế

Bước 1. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➡ Programs ➡ Microsoft Visual Studio .Net ➡ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File ➡ New Project kích hoạt cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = **MDI.vbproj**
- Định thư mục lưu project: Location = **C:\BTVBNET**
- Nhấp tùy chọn **Close Solution**.
- Chọn Project types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.
- Xong nhấp OK.

VB.NET sẽ tự động khởi tạo một project mới với một form trống mặc định là Form1.vb chứa trong thư mục **MDI**, là thư mục con của **C:\BTVBNET**.

Bước 2. Thiết kế form chính – frmMain.

Chọn Form1 và định lại các thuộc tính sau trong cửa sổ Properties:

Name = FrmMain

Text = Main Form

IsMDIContainer = True (xác định form thuộc loại MDI)

MaximizeBox = True

MinimizeBox = True

```
FormBorderStyle = Sizable
```

```
StartPosition = CenterScreen
```

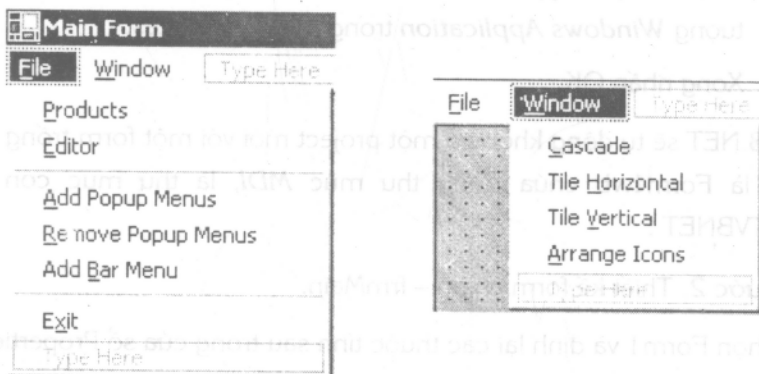
```
WindowState = Maximized
```

```
Menu = mnuMain
```

Bổ sung menu vào form bằng cách nhấp đúp điều khiển MainMenu trên Toolbox.

Nhấp vào mục Type Here và nhập File. Vb.Net tự động thêm các mục Type Here bên phải và dưới mục menu bạn vừa nhập để chờ bạn định nghĩa các lệnh trên menu ngang và các tùy chọn trong menu xổ xuống ứng với từng lệnh.

Tiếp tục thực hiện việc định nghĩa menu để được kết quả như Hình 13.17.



Hình 13.17 Menu form frmMain sau khi hoàn tất.

- Định lại tên cho các tùy chọn Menu:

Chọn mục File, trong cửa sổ Properties, đổi thuộc tính **Name** từ *MenuItem1* thành *mnuFile* và thuộc tính **Text** thành *&File*. Tương tự, bạn định lại tên cho các mục chọn trên menu như sau:

Tên mục chọn menu (Text)	Thuộc tính Name
&File	MnuFile
&Products	MnuFProducts
&Editor	MnuFChildMenus
-	MenuItem1
&Add Popup Menus	MnuFAddMenus
&Remove Popup Menus	MnuFRemoveMenus
Add &Bar Menus	MnuFAddNew
-	MenuItem4
E&xit	MnuFExit
&Window	MnuWindow
&Cascade	MnuWCascade
Tile &Horizontal	MnuWHorizontal
Tile &Vertical	MnuWVertical
&Arrange Icons	MnuWArrange

Nhập điều khiển MainMenu1 nằm dưới vùng thiết kế form và đặt tên cho nó trong cửa sổ Properties:

Name = MnuMain

- Viết mã lệnh xử lý form: chọn lệnh View Code

Public Class frmMain

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

' Vùng mã lệnh do VB.Net tự phát sinh trong quá trình thiết kế form

.....

#End Region

' Khi chọn lệnh File ➡ Products

Private Sub mnuFProducts_Click(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles mnuFProducts.Click

Dim frm As frmListEnh

frm = New frmListEnh()

frm.MdiParent = Me

Call CenterChildForm(frm)

frm.Show()

End Sub

' Thủ tục canh form con giữa form chính frmMain

Public Sub CenterChildForm(ByVal frmChild As Form)

Dim ptLoc As Point = New Point()

ptLoc.X = (Me.ClientRectangle.Width / 2) _

- (frmChild.Size.Width / 2)

ptLoc.Y = (Me.ClientRectangle.Height / 2) _

- (frmChild.Size.Height / 2)

frmChild.Location = ptLoc

End Sub

' Khi chọn lệnh Window ➡ Cascade

Private Sub mnuWCascade_Click(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles mnuWCascade.Click

Me.LayoutMdi(MdiLayout.Cascade)

End Sub

' Khi chọn lệnh File ➡ Add Popup Menus

Private Sub mnuFAddMenus_Click(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles mnuFAddMenus.Click

If Not mnuFAddMenus.Checked Then

' Bật dấu kiểm trước lệnh Add Popup Menus

mnuFAddMenus.Checked = True

' Tắt dấu kiểm trước lệnh Remove Popup Menus

mnuFRemoveMenus.Checked = False

' Thêm các mục chọn menu vào cuối menu popup File

mnuFile.MenuItems.Add("File1.txt")

mnuFile.MenuItems.Add("File2.txt")

mnuFile.MenuItems.Add("File3.txt")

End If

End Sub

' Khi chọn lệnh File ➡ Remove Popup Menus

Private Sub mnuFRemoveMenus_Click(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles mnuFRemoveMenus.Click

' Nếu dấu kiểm trước lệnh Remove Popup Menus chưa được bật

' nghĩa là các mục chọn menu ở cuối menu popup File chưa được gỡ

If Not mnuFRemoveMenus.Checked Then

' Tắt dấu kiểm trước lệnh Add Popup Menus

mnuFAddMenus.Checked = False

' Bật dấu kiểm trước lệnh Remove Popup Menus

mnuFRemoveMenus.Checked = True

With mnuFile.MenuItems

' Gỡ bỏ 3 mục chọn menu đã thêm vào cuối menu popup File


```
.RemoveAt(.Count - 1)
.RemoveAt(.Count - 1)
.RemoveAt(.Count - 1)
```

```
End With
```

```
End If
```

End Sub

' Khi chọn lệnh File ➡ Exit

```
Private Sub mnuFExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuFExit.Click
```

```
Me.Close()      ' đóng form chính và kết thúc chương trình
```

End Sub

' Khi chọn lệnh Window ➡ Tile Horizontal

```
Private Sub mnuWHorizontal_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuWHorizontal.Click
```

```
Me.LayoutMdi(MdiLayout.TileHorizontal)
```

End Sub

' Khi chọn lệnh Window ➡ Tile Vertical

```
Private Sub mnuWVertical_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuWVertical.Click
```

```
Me.LayoutMdi(MdiLayout.TileVertical)
```

End Sub

' Khi chọn lệnh Window ➡ Arrange

```
Private Sub mnuWArrange_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles mnuWArrange.Click
```

```
Me.LayoutMdi(MdiLayout.ArrangeIcons)
```

End Sub

' Khi chọn lệnh File ➡ Add Popup Menus

```
Private Sub mnuFAddNew_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles mnuFAddNew.Click
    ' Nếu chưa có dấu kiểm trước lệnh Add Bar Menus
    If Not mnuFAddNew.Checked Then
        ' Bật dấu kiểm trước lệnh Add Bar Menus
        mnuFAddNew.Checked = True
        ' Khai báo các biến đối tượng Mainmenu và MenuItem
        Dim mnuNew As MainMenu
        Dim mnuMaint As MenuItem
        ' Tạo một popup menu mới
        mnuNew = New MainMenu()
        mnuMaint = mnuNew.MenuItems.Add("Help")
        ' Đảm bảo chèn menu Help vào sau menu Window của chương trình
        mnuMaint.MergeOrder = 2
        mnuMaint.MergeType = MenuMerge.Add
        ' Bổ sung các mục chọn vào menu Help
        mnuMaint.MenuItems.Add("Content")
        mnuMaint.MenuItems.Add("Topic")
        ' Gắn vào menu chính của chương trình
        mnuMain.MergeMenu(mnuNew)
    End If
```

End Sub

' Khi chọn lệnh File ➡ Editor.

```
Private Sub mnuFChildMenus_Click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles mnuFChildMenus.Click
    Dim frm As frmChildWithMenus
    frm = New frmChildWithMenus()
```

```
frm.MdiParent = Me
```

```
frm.Show()
```

End Sub

' Khi form chính frmMain được nạp

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

```
Dim frmLogin As frmLogin
```

' Hiển thị form frmLogin

```
frmLogin = New frmLogin()
```

' Khi đóng form frmLogin bằng cách nhấp nút OK

```
If frmLogin.ShowDialog() = DialogResult.OK Then
```

' Kiểm tra user name và password

```
If frmLogin.txtLoginID.Text.ToString() = "DINH XUAN LAM" And
```

```
frmLogin.txtPassword.Text.ToString() = "XL TT" Then
```

```
MessageBox.Show("Valid Login ID and Password")
```

Else ' Nếu username hoặc password không đúng

```
Me.Close() ' kết thúc chương trình
```

```
End If
```

Else ' Khi đóng form frmLogin bằng nút Cancel

```
Me.Close() ' kết thúc chương trình
```

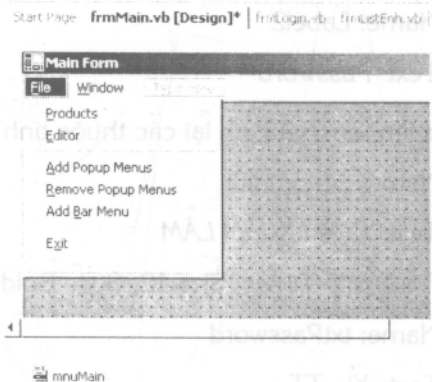
```
End If
```

End Sub

End Class

Lưu form với tên **frmMain.vb**.

Kết quả thiết kế form **frmMain** như Hình 13.18.



Hình 13.18 Màn hình thiết kế form frmMain.

Bước 3. Thiết kế form frmLogin.

- Chọn lệnh Project ➡ Add Windows Form. Trong cửa sổ Add New Item, click biểu tượng Windows Form trong mục Templates. Nhập tên form mới là **frmLogin** trong mục Name, xong nhấn Open.
- Định lại một số thuộc tính cho form frmLogin:
 Name: frmLogin
 Text: Login
 Size: 296, 149
 StartPosition: CenterScreen
 AcceptButton: btnOK (Nhấn Enter tương đương với việc nhấn nút OK)
 CancelButton: btnCancel (Nhấn ESC tương đương nhấn nút Cancel)
- Đặt 2 Label lên form và định lại các thuộc tính như sau:
 Label 1: Name: Label1
 Text: Login ID

Label 2: Name: Label2

Text: *Password*

- Đặt 2 Textbox lên form và định lại các thuộc tính như sau:

Textbox 1: Name: txtLoginID

Text: *ĐINH XUÂN LÂM*

Font: VNI-Times, Size 12, Style Bold

Textbox 2: Name: txtPassword

Text: *XL TT*

PasswordChar: *

Mặc định, nội dung các Textbox được xem là Username và password dùng để login chương trình.

- Đặt 2 Button lên form và định lại các thuộc tính như sau:

Button 1: Name: btnCancel

Text: &Cancel

TextAlign = MiddleCenter

DialogResult = Cancel (*Giá trị trả lại khi nút Cancel được nhấp*)

Button 2: Name: btnOK

Text: &OK

TextAlign = MiddleCenter

DialogResult = OK (*Giá trị trả lại khi nút OK được nhấp*)

- Mã lệnh xử lý form frmLogin: thực ra trong trường hợp này bạn chẳng cần viết thêm dòng lệnh nào.

Public Class frmLogin

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

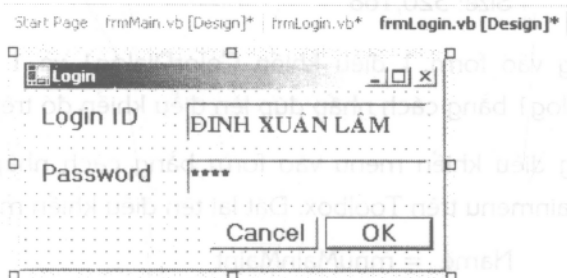
‘Phần mã lệnh do Windows Form Designer tự phát sinh

‘ Bạn không cần điều chỉnh gì trong đoạn lệnh này

#End Region

End Class

Kết quả thiết kế form frmLogin sẽ như Hình 13.19.



Hình 13.19 Màn hình thiết kế form frmLogin.

Bước 4. Thiết kế form frmChildWithMenus.

- Chọn lệnh Project ➡ Add Windows Form. Trong cửa sổ Add New Item. Nhấp biểu tượng Windows Form trong mục Templates, nhập tên form mới là **frmChildWithMenus** trong mục Name, xong nhấp Open.
- Định lại một số thuộc tính cho form frmChildWithMenus:

Name: frmChildWithMenus

Text: frmChildWithMenus

Menu: mnuMainMaint

Size: 328,208

- Đặt 1 Textbox lên form và định lại các thuộc tính như sau:

Textbox 1: Name: Textbox1

Text: (Rỗng)

MultiLine: True

TabIndex: 0

WordWrap: True

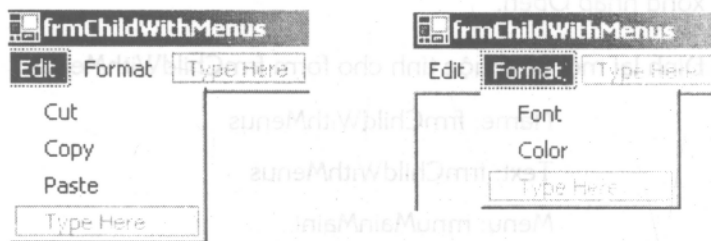
Size: 320,168

- Bổ sung vào form 1 điều khiển ColorDialog1 và 1 điều khiển FontDialog1 bằng cách nhấp đúp lên điều khiển đó trên Toolbox.
- Bổ sung điều khiển menu vào form bằng cách nhấp đúp điều khiển MainMenu trên Toolbox. Đặt lại tên điều khiển menu:

Name = mnuMainMaint

Nhấp vào mục Type Here và nhập Edit. VB.Net tự động thêm các mục Type Here bên phải và dưới mục menu bạn vừa nhập để chờ bạn định nghĩa các lệnh trên menu ngang và các tùy chọn trong menu xổ xuống ứng với từng lệnh.

Tiếp tục thực hiện việc định nghĩa menu để được kết quả như Hình 13.20.



Hình 13.20 Menu form frmChildWithMenus sau khi hoàn tất.

- Định lại tên cho các tùy chọn Menu:

Chọn từng mục trong menu, đổi thuộc tính **Name** trong cửa sổ Properties như sau:

Tên mục chọn menu (thuộc tính Text)	Thuộc tính Name
Edit	MnuEdit
Cut	MenuItem1
Copy	MenuItem2
Paste	MenuItem3
Format	MnuMaint
Font	MnuFont
Color	mnuColor

Nhấp điều khiển MainMenu1 nằm dưới vùng thiết kế form và đặt tên cho nó trong cửa sổ Properties:

Name = MnuMainMaint

- Viết mã lệnh xử lý form frmChildWithMenus:

Public Class frmChildWithMenus

Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

'Phần mã lệnh do Windows Form Designer tự phát sinh

'Bạn không cần điều chỉnh gì trong đoạn lệnh này

.....

#End Region

' Khi chọn lệnh Format ➡ Font

```
Private Sub mnuFont_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles mnuFont.Click  
    FontDialog1.ShowDialog()  
    TextBox1.Font = FontDialog1.Font
```

End Sub

' Khi chọn lệnh Format ➡ Color

```
Private Sub mnuColor_Click(ByVal sender As System.Object, ByVal e As  
    System.EventArgs) Handles mnuColor.Click  
    ColorDialog1.ShowDialog()  
    TextBox1.ForeColor() = ColorDialog1.Color
```

End Sub

' Khi chọn lệnh Edit ➡ Cut

```
Private Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e  
    As System.EventArgs) Handles MenuItem1.Click  
    TextBox1.Cut()
```

End Sub

' Khi chọn lệnh Edit ➡ Copy

```
Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e  
    As System.EventArgs) Handles MenuItem2.Click  
    TextBox1.Copy()
```

End Sub

' Khi chọn lệnh Edit ➡ Paste

Private Sub MenuItem3_Click(ByVal sender As System.Object, ByVal e

As System.EventArgs) Handles MenuItem3.Click

TextBox1.Paste()

End Sub

End Class

Kết quả màn hình thiết kế form *frmChildWithMenus* sẽ như Hình 13.21.



Hình 13.21 Màn hình thiết kế form *frmChildWithMenus*.

Bước 5. Thiết kế form *frmListEnh*.

Nhận xét: Dễ thấy form này tương tự như form cùng tên đã trình bày trong bài thực hành 12. Do vậy, thay vì tạo mới từ đầu, bạn có thể copy 2 tập tin *frmListEnh.frm* và *frmListEnh.resx* từ thư mục C:\BTVBNET\Listbox sang thư mục C:\BTVBNETMDI. Xong thêm nó vào project và tiến hành một vài điều chỉnh cho nhanh.

- Chọn lệnh Project ➡ Add Existing Item, trong cửa sổ Add Existing Item, click form *frmListEnh*, xong nhấp Open để thêm form *frmListEnh* vào project.

- Định lại một số thuộc tính cho form frmListEnh:

Name: frmListEnh

Text: Product Information

ContextMenu: ContextMenu1

- Lần lượt bổ sung thêm điều khiển ColorDialog1 và FontDialog1 vào form.
- Để tạo menu ngữ cảnh (được gọi khi nhấp nút phải mouse), bổ sung điều khiển ContextMenu từ Toolbox vào form với tên ContextMenu1 (*thuộc tính Name*).
- Định nghĩa các mục chọn trong menu Context như sau:

Tên mục chọn menu (thuộc tính Text)	Thuộc tính Name
Font	MenuItem1
Color	MenuItem2

- Viết mã lệnh xử lý form *frmListEnh*:

Tương tự như đã trình bày trong bài thực hành trước. Giờ đây, chúng tôi chỉ trình bày thêm các đoạn lệnh cần bổ sung.

Public Class Product

' Phần mã lệnh đã trình bày trong bài thực hành trước

End Class

Public Class frmListEnh

' Phần mã lệnh đã trình bày trong bài thực hành trước

.....

' Phần mã lệnh cần bổ sung thêm

' Khi chọn lệnh Font trên menu nhanh

```
Private Sub MenuItem1_Click(ByVal sender As System.Object, ByVal e
```

```
As System.EventArgs) Handles MenuItem1.Click
```

```
FontDialog1.ShowDialog()
```

```
lstProducts.Font = FontDialog1.Font
```

End Sub

' Khi chọn lệnh Color trên menu nhanh

```
Private Sub MenuItem2_Click(ByVal sender As System.Object, ByVal e
```

```
As System.EventArgs) Handles MenuItem2.Click
```

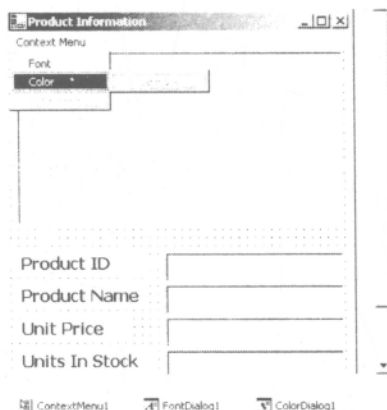
```
ColorDialog1.ShowDialog()
```

```
Me.BackColor = ColorDialog1.Color
```

End Sub

End Class

Kết quả màn hình thiết kế form frmListEnh sẽ như Hình 13.22.



Hình 13.22 Màn hình thiết kế form frmListEnh.

Bước 6. Chạy thử chương trình để xem kết quả.

Đặt form frmMain làm form chạy chính trong chương trình.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project

Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start chạy thử chương trình.

Bước 7. Thi hành chương trình dưới dạng tập tin EXE.

Để chạy chương trình ngoài Windows mà không cần khởi động VB.Net, chọn tập tin MDI.exe trong thư mục C:\BTVBNET\MDI\Bin.

Bài 14.

Chương trình cơ sở dữ liệu – Thiết kế form tra cứu nhân viên dạng 1

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với kỹ thuật thiết kế form tra cứu, cách truy cập và chọn lọc dữ liệu trong cơ sở dữ liệu, cách định nghĩa các đối tượng kết nối dữ liệu mà không cần sử dụng Wizard, cách sắp xếp thứ tự và độ rộng của các cột thông tin hiển thị trong điều khiển DataGrid.

Hoạt động chương trình

Khi thi hành chương trình, bạn sẽ được màn hình có giao diện như Hình 14.1. Chương trình sẽ tự động mở màn hình xem thông tin nhân viên theo từng phòng ban.

Người dùng có thể chọn phòng ban bằng cách nhấp vào mũi tên cuối combo box để mở danh sách phòng ban (Hình 14.2). Sau khi người dùng chọn tên 1 phòng ban, thông tin các nhân viên thuộc phòng ban đó sẽ được hiển thị trên DataGrid (Hình 14.1).

Tra cứu nhân viên theo phòng

_ □ ×

Phòng ban : Phòng Tài Chính

	Mã phòng	Mã nhân viên	Họ và tên	Phái	Địa chỉ	Điện thoại	Ngày sinh
▶	N0001	N006	Lương Bửu Thắng	Trực	92 Lê Thái Tổ Q1	8940549	10/24/1980
	N0001	N007	Lâm Sơn Hoàng	Trực	74 Kỳ Co Q1	8740374	7/15/1979
	N0001	N111	Nguyễn Hoàng Anh	Phải	2 Nguyễn Công Trứ	8111111	9/20/1973
*							

Hình 14.1 Thông tin các nhân viên thuộc phòng Tài chính.

Tra cứu nhân viên theo phòng

Phòng ban : Phòng Tài Chính

Phòng Nhân Sự
Phòng Kinh Doanh
Phòng Kế Toán
Phòng Marketing
Phòng Tài Chính
Phòng Thư Ký
Phòng GD
Phòng Kỹ Thuật

Hình 14.2 Danh sách các phòng ban trong combo box.

CHÚ Ý

Form phải bảo đảm đáp ứng các yêu cầu sau:

- ◆ Danh sách các phòng ban trong combo box được đọc từ table DMPhongban thuộc cơ sở dữ liệu Access nhansu.mdb.
- ◆ DataGridView chỉ hiển thị một số thông tin cần biết của từng nhân viên theo một trật tự xác định, chứ không hiển thị toàn bộ các field trong table DMNhanvien.

Hướng dẫn thiết kế

Bước 1. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➤ Programs ➤ Microsoft Visual Studio .Net ➤ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File ➤ New Project để gọi cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = **Bai14**
- Định thư mục lưu project: Location = **C:\BTVBNET**
- Nhấp tùy chọn **Close Solution**.
- Chọn Project types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.
- Xong nhấp OK.

VB.NET sẽ tự động khởi tạo 1 project mới với 1 form trống mặc định là Form1.vb chứa trong thư mục Bai14, là thư mục con của C:\BTVBNET.

Bước 2. Tạo form FTracuu.vb dùng tra cứu thông tin nhân viên theo phòng ban.

Chọn lệnh View ➤ Solution Explorer để bật cửa sổ Solution Explorer nếu chưa thấy.

- Đặt lại tên tập tin cho form1:

Name = Ftracuu.vb

- Định lại các thuộc tính cho Form1 như sau:

Name = Ftracuu

Text = Tra cuu nhan vien theo phong

BackColor = ActiveCaptionText

ForeColor = ControlText

Size = 688, 352

- Định nghĩa nối kết dữ liệu tới table DMPhongban để truy xuất thông tin các phòng ban:

Trong Toolbox, chọn tab Data (Hình 14.3).



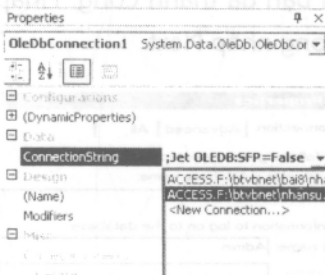
Hình 14.3 Tab Data trong Toolbox.

- Đặt 1 điều khiển OleDbConnection vào form bằng cách nhấp đúp điều khiển đó trên Toolbox. Điều khiển sẽ hiển thị ngay dưới form với tên mặc định là:

Name = OleDbConnection1.

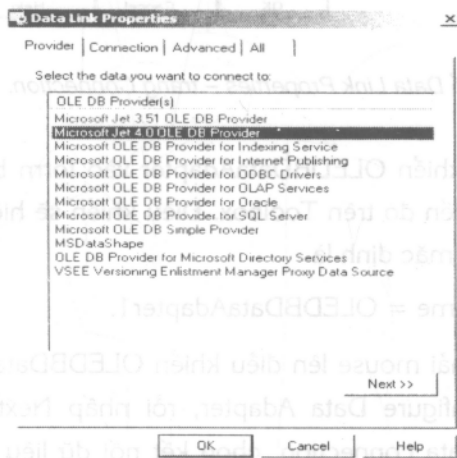
Nhấp nút phải mouse lên điều khiển OleDbConnection1 và chọn lệnh Properties... trên menu nhanh để bật cửa sổ Properties.

Chọn thuộc tính `ConnectionString`, nhấp nút mũi tên ở cuối popup và chọn lệnh `New Connection...` (Hình 14.4) để gọi cửa sổ `Data Link Properties`.



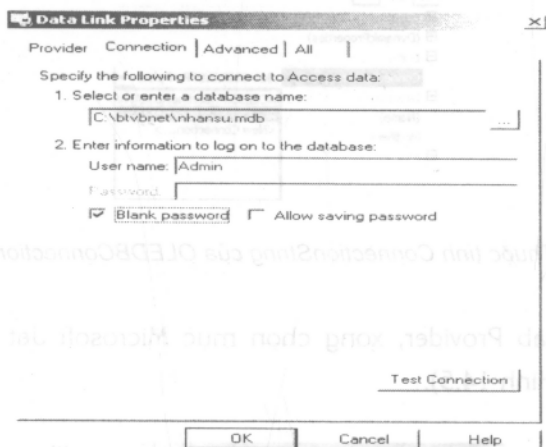
Hình 14.4 Thuộc tính `ConnectionString` của `OleDbConnection1`.

Chọn tab `Provider`, xong chọn mục `Microsoft Jet 4.0 OLE DB Provider` (Hình 14.5).



Hình 14.5 Cửa sổ `Data Link Properties` – trang `Provider`.

Chọn tab Connection, nhấp nút ... ở phía sau text box *Select or enter a database name* để chọn cơ sở dữ liệu kết nối (Hình 14.6). Xong nhấp Test Connection, nếu thấy thông báo *Test connection succeeded*, xem như bạn đã thành công. Nhấp OK để trở lại cửa sổ Data Form Wizard, xong nhấp Next.



Hình 14.6 Cửa sổ Data Link Properties – trang Connection.

- Đặt 1 điều khiển OLEDBDataAdapter vào form bằng cách nhấp đúp điều khiển đó trên Toolbox. Điều khiển sẽ hiển thị ngay dưới form với tên mặc định là

Name = OLEDBDataAdapter1.

Nhấp nút phải mouse lên điều khiển OLEDBDataAdapter1, xong chọn lệnh Configure Data Adapter, rồi nhấp Next, trong cửa sổ Choose your data connection, chọn kết nối dữ liệu mà bạn đã tạo Access.C:\Btvbn\nhansu.mdb.Admin (Hình 14.7), xong nhấp Next.



Hình 14.7 Chọn cơ sở dữ liệu Access muốn kết nối dữ liệu.

Trong cửa sổ *Choose a query type*, chọn tùy chọn *Use SQL Statements*, xong tiếp tục nhấn *Next*.

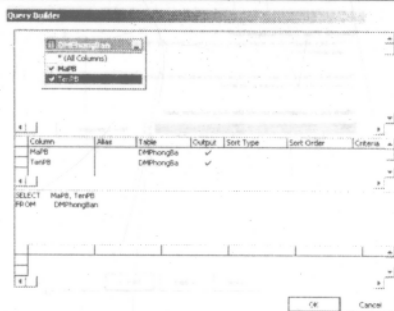
Trong cửa sổ *Generate the SQL statements*, nhấn nút *Query Builder*.

Chọn table *DMPHONGBAN* trong cửa sổ *Add Table*, nhấn *Add* để thêm vào cửa sổ tạo query (14.8), xong nhấn *Close*.



Hình 14.8 Cửa sổ *Add Table*.

Đánh dấu các field trong table *DMPHONGBAN* (Hình 14.9), xong nhấn *OK* để trở lại cửa sổ *Generate the SQL Statements* (Hình 14.10).



Hình 14.9 Cửa sổ tạo query.



Hình 14.10 Kết quả của sổ Generate the SQL Statements sau khi tạo query.

Nhấp Next khi trở lại cửa sổ Generate the SQL statements.

Cuối cùng nhấp Finish trong cửa sổ View Wizard Results.

- Phát sinh điều khiển Dataset1 tương ứng với:

Nhấp nút phải mouse lên điều khiển OleDbDataAdapter1, chọn lệnh **Generate Dataset** trên menu nhanh. Trong cửa sổ Generate Dataset, chọn tùy chọn New, nhập vào tên điều khiển DataSet sẽ phát sinh là DataSet1 (Hình 14.11). Xong nhấp OK.

Generate Dataset

Generate a dataset that includes the specified tables.

Choose a dataset:

☒ Existing ☐ New

Choose which table(s) to add to the dataset:

☒ DMNhanvien (OLEDBDataAdapter)

☐ DMHuongdan (OLEDBDataAdapter)

☒ Add this dataset to the designer.

OK Cancel Help

Hình 14.11 Cửa sổ Generate DataSet.

Định nghĩa nối kết dữ liệu tới table DMNhanvien để truy xuất có chọn lọc theo phòng ban một số thông tin nhân viên:

- Đặt 1 điều khiển OLEDBConnection thứ hai vào form bằng cách nhấp đúp điều khiển đó trên Toolbox. Điều khiển sẽ hiển thị ngay dưới form với tên mặc định là

Name = OLEDBConnection2

Nhấp nút phải mouse lên điều khiển OLEDBConnection2 và chọn lệnh Properties... trên menu nhanh để kích hoạt cửa sổ Properties.

Chọn thuộc tínhConnectionString trong cửa sổ Properties, nhấp nút mũi tên ở cuối popup và chọn kết nối cơ sở dữ liệu đã tạo sẵn *Access.C:\Btvbnet\nhansu.mdb* (xem lại Hình 14.4).

- Đặt 1 điều khiển OLEDBDataAdapter thứ hai vào form bằng cách nhấp đúp điều khiển đó trên Toolbox. Điều khiển sẽ hiển thị ngay dưới form với tên mặc định là

Name = OLEDBDataAdapter2

Nhấp phải điều khiển OleDbDataAdapter2, xong chọn lệnh Configure Data Adapter, rồi nhấp Next. Trong cửa sổ Choose your data connection, chọn kết nối dữ liệu mà bạn đã tạo Access.C:\Btvbnet\nhansu.mdb.Admin, xong nhấp Next.

Trong cửa sổ Choose a query type, chọn tùy chọn Use SQL Statements, xong tiếp tục nhấp Next.

Trong cửa sổ Generate the SQL statements, nhấp nút Query Builder.

Chọn table DMNhanvien trong cửa sổ Add Table, nhấp Add để thêm vào cửa sổ Query, xong nhấp Close.

Đánh dấu các field và sắp chúng lại theo thứ tự mapb, manv, hoten, phai, dchi, dienthoai, ngaysinh trong table DMNhanvien. Xong nhập điều kiện truy vấn =? vào cột Criteria của field Mapb (Hình 14.12). Sau đó nhấp OK để trở lại cửa sổ Generate the SQL Statements (Hình 14.13).



Hình 14.12 Cửa sổ tạo query truy vấn table DMNhanvien.

Nhấp Next khi trở lại cửa sổ Generate the SQL statements.

Cuối cùng nhấp Finish trong cửa sổ View Wizard Results.



Hình 14.13 Kết quả của sổ *Generate the SQL Statements* sau khi tạo query.

- Phát sinh điều khiển Dataset2 tương ứng với:

Nhấp nút phải mouse lên điều khiển `OleDbDataAdapter2`, chọn lệnh **Generate Dataset** trên menu nhanh. Trong cửa sổ *Generate Dataset*, chọn tùy chọn **New**, nhập vào tên điều khiển Dataset sẽ phát sinh là `Dataset2`. Xong nhấp **OK**.

+ Đặt lên form 1 điều khiển Combo box và định lại các thuộc tính như sau:

Name = `Combo1`

CaptionFont = `VNI-Times`, style **Bold**, size 8

Font = `VNI-Times`, style **Regular**, size 8

DataMember = `DMNhanvien` ' lấy dữ liệu từ `DMNhanvien`

DataSource = `Dataset21`

Location = 8,64

Size = 664,240

+ **Đặt lên form 1 điều khiển DataGrid** và định lại các thuộc tính như sau:

Name = DataGrid1

Font = VNI-Times, style Regular, size 10

DataSource = Dataset1.Dmphonngban

DisplayMember = Tenpb

ValueMember = Mapb

Location = 120,32

Size = 184,25

+ Bổ sung tiêu đề các cột, định lại thứ tự hiển thị các field và độ rộng mỗi cột trên Datagrid1.

Nhấp nút ... bên phải thuộc tính TableStyles của DataGrid1 (Hình 14.14) để gọi cửa sổ DataGridTableStyle Collection Editor (Hình 14.15).



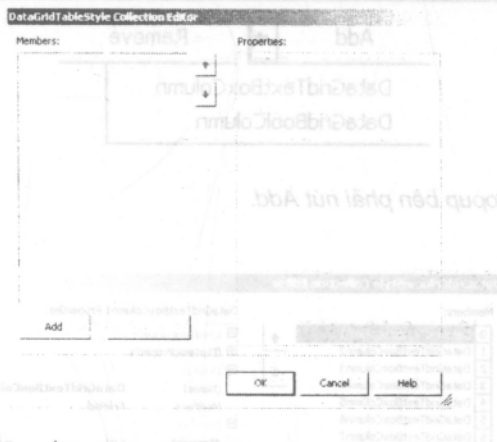
Hình 14.14 Nút ... bên phải thuộc tính TableStyles của DataGrid1.

Nhấp nút Add để bổ sung thành viên DataGridTableStyle1 vào cửa sổ và định lại các thuộc tính như sau:

Name = DataGridTableStyle1

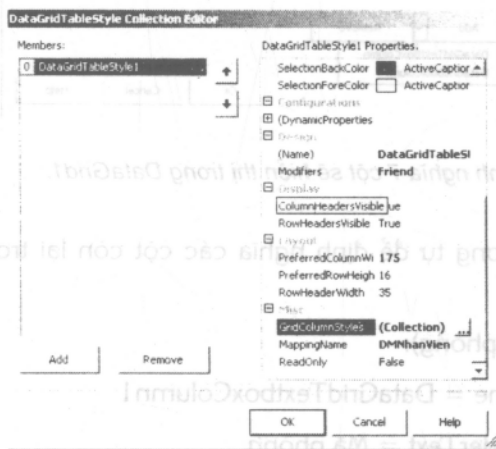
MappingName = DMNhanvien

HeaderFont = VNI-Times, Bold, 8



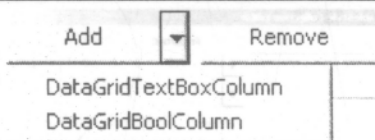
Hình 14.15 Cửa sổ DataGridTableStyle Collection Editor.

Nhấp nút ... bên phải thuộc tính GridColumnStyles (Hình 14.16) để gọi cửa sổ DataGridColumnStyle Collection Editor.

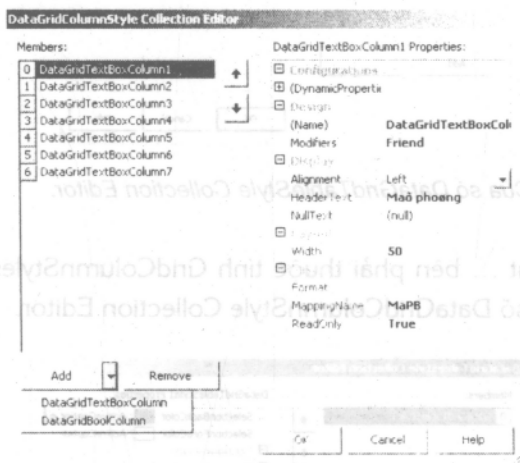


Hình 14.16 Nút ... bên phải thuộc tính GridColumnStyles.

Nhấp popup bên phải nút Add, chọn DataGridTextBoxColumn (Hình 14.17) để bổ sung 1 cột mới vào điều khiển DataGrid1. Xong định lại giá trị các thuộc tính cho nó (Hình 14.18).



Hình 14.17 Popup bên phải nút Add.



Hình 14.18 Định nghĩa 7 cột sẽ hiển thị trong DataGrid1.

Thực hiện tương tự để định nghĩa các cột còn lại trong DataGrid1, theo bảng sau:

- Cột 1 (Mã phòng):

Name = DataGridTextboxColumn1

HeaderText = Mã phòng

Width = 50

MappingName = MaPb

- Cột 2 (Mã nhân viên):

Name = DataGridTextboxColumn2

HeaderText = Mã nhân viên

Width = 75

MappingName = MaNv

- Cột 3 (Họ và tên):

Name = DataGridViewTextBoxColumn3

HeaderText = Họ và tên

Width = 150

MappingName = Hoten

- Cột 4 (Phái):

Name = DataGridViewTextBoxColumn4

HeaderText = Phái

Width = 40

MappingName = Phai

- Cột 5 (Địa chỉ):

Name = DataGridViewTextBoxColumn5

HeaderText = Địa chỉ

Width = 150

MappingName = Dchi

- Cột 6 (Điện thoại):

Name = DataGridViewTextBoxColumn6

HeaderText = Điện thoại

Width = 75

MappingName = Dienthoai

- Cột 7 (Ngày sinh):

Name = DataGridViewTextBoxColumn7

HeaderText = Ngày sinh

Width = 50

MappingName = Ngaysinh

- Các cột đều có chung thuộc tính:

ReadOnly = True

Alignment = Left

- Viết mã lệnh xử lý form *Ftracuu*

Public Class Ftracuu

Inherits System.Windows.Forms.Form

' Vùng mã lệnh do VB.Net tự phát sinh trong quá trình bạn thiết kế form

#Region " Windows Form Designer generated code "

' Bạn không nên điều chỉnh tay các mã lệnh đang có trong vùng này

#End Region

' Thủ tục được gọi khi nạp form Ftracuu

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

' Xóa trống nội dung DataSet11

DataSet11.Clear()

' Nạp lại nội dung table DMPHONGBAN vào điều khiển Dataset11

OleDbDataAdapter1.Fill(DataSet11, "DMPHONGBAN")

End Sub

' Thủ tục được gọi khi chọn một phòng ban trong ComboBox1

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles

ComboBox1.SelectedIndexChanged

```

' Gán giá trị mã phòng ban được chọn (giá trị Combobox1)
' cho tham số mà OleDbDataAdapter2 yêu cầu
With OleDbDataAdapter2.SelectCommand.Parameters
    .Item(0).Value = ComboBox1.Selectedvalue
End With

' Xóa trống nội dung DataSet21
DataSet21.Clear()

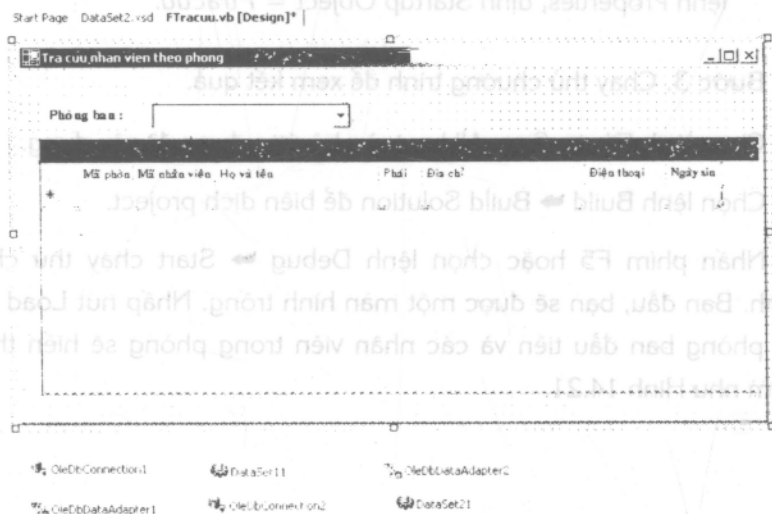
' Nạp lại nội dung table DMNHANVIEN vào điều khiển Dataset21 có
' chọn lọc theo mã phòng ban được chọn trong Combobox1
OleDbDataAdapter2.Fill(DataSet21, "DMNHANVIEN")

```

End Sub

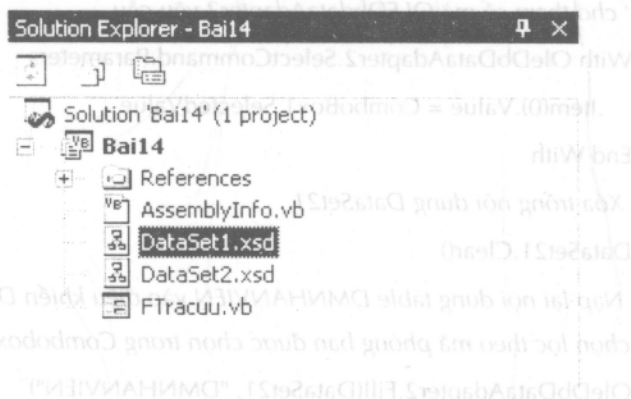
End Class

- Kết quả của sự thiết kế form *FTracuu* sẽ như Hình 14.19.



Hình 14.19 Cửa sổ thiết kế form *Ftracuu*.

- Kết quả của sổ Solution Explorer sẽ như Hình 14.20.



Hình 14.20 Cửa sổ Solution Explorer sau khi thiết kế xong form *Ftracuu*.

- Đặt form *Ftracuu* làm form khởi động bằng cách nhấp nút phải mouse lên project *Bai14* trong của sổ Solution Explorer, chọn lệnh Properties, định Startup Object = *Ftracuu*.

Bước 3. Chạy thử chương trình để xem kết quả.

Chọn lệnh File ➤ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➤ Build Solution để biên dịch project.

Nhấn phím F5 hoặc chọn lệnh Debug ➤ Start chạy thử chương trình. Ban đầu, bạn sẽ được một màn hình trống. Nhấp nút Load thông tin phòng ban đầu tiên và các nhân viên trong phòng sẽ hiển thị trên form như Hình 14.21.

Hình 14.21 Màn hình form FTracuu khi vừa thi hành.

Nhấp popup Phòng ban để chọn “Phòng tài chính”, bạn sẽ được kết quả như Hình 14.1. Giả sử, dữ liệu bạn đã nhập vào các table DMPhongban và DMNhanvien trong cơ sở dữ liệu Access Nhansu.mdb như các bảng sau:

DMPhongBan : Table		
	Mã PB	Tên PB
+	G0001	Phòng Nhân Sự
+	G0002	Phòng Kinh Doanh
+	K0001	Phòng Kế Toán
+	M0001	Phòng Marketing
+	N0001	Phòng Tài Chính
+	N0002	Phòng Thư Ký
+	P0001	Phòng GD
+	T0001	Phòng Kỹ Thuật

Mã NV	Mã PB	Họ và tên	Phái	Chức vụ	Ngày sinh	Nơi sinh	Bộ chỉ	Điện thoại
N001	C0001	Nguyễn Ngọc Nga	Nữ	GD Nhân sự	12/5/1970	TP HCM	19 Hồng Vương P4 Q5	0968155
N002	C0002	Bà Minh Phúc	Nam	GD Kinh doanh	2/14/1975	An Giang	89 Đồng Khởi Q1	0750074
N003	N0002	Tiểu Tuyết Thanh	Nữ	Thư ký văn phòng	9/20/1978	Cần Thơ	26 Lê Quý Đôn P5 Q8	0557796
N004	K0001	Nguyễn Kim Ngọc	Nữ	Kế toán trưởng	6/1/1974	Hà Nội	178 Hùng Vương P6 Q6	0553270
N005	T0001	Trương Duy Hùng	Nam	Trưởng phòng Kỹ Thuật	3/10/1971	TP Hồ Chí Minh	77 Trương Định P6 Q3	0940799
N006	N0001	Lương Bá Thắng	Nam	Nhân viên	10/24/1980	Đồng Nai	92 Lê Thánh Tôn Q1	0946546
N007	N0001	Lâm Sơn Hoàng	Nam	Nhân viên	7/15/1979	TP Hồ Chí Minh	74 Kỳ Con Q1	0740374
N008	P0001	Nguyễn Minh Hồng	Nam	Phó Tổng GD	11/30/1965	Hà Nội	274 Lạc Long Quân Q1	0382576
N009	M0001	Vương Mạc Lan	Nữ	Trưởng phòng Marketing	1/5/1973	TP Hồ Chí Minh	227 Hai Bà Trưng Q3	0851636
N111	N0001	Nguyễn Hoàng Anh	Nữ	Nhân viên	9/20/1973	Long An	2 Nguyễn Công Trứ	0111111

Nhấp nút X ở góc trên bên phải của sổ Form để kết thúc chương trình.

Bài 15.

Chương trình cơ sở dữ liệu – Thiết kế form tra cứu nhân viên dạng 2

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với kỹ thuật duyệt nhanh thông tin từng record trong 1 table thông qua con đường viết mã lệnh, sử dụng đối tượng DataReader, cũng như tìm hiểu sâu hơn về ứng dụng của kỹ thuật lập trình hướng đối tượng trong VB.Net.

Hoạt động chương trình

Khi thi hành chương trình, bạn sẽ được màn hình có giao diện như Hình 15.1. Chương trình sẽ tự động nạp danh sách tên các nhân viên trong table DMNhanvien vào Listbox bên trái và hiển thị đầy đủ thông tin của nhân viên đầu tiên trong Listbox ở bên phải.

Để xem thông tin của nhân viên khác, người dùng chỉ việc nhấp vào tên nhân viên trong Listbox, hoặc dùng các phím mũi tên để cuộn danh sách lên hoặc xuống cho đến khi thanh sáng dịch chuyển đến vị trí có tên nhân viên đó (Hình 15.2).

Thông tin nhân viên			
Nguyễn Ngọc Nga	Mã nhân viên	N001	
Hà Vinh Phát	Họ và tên	Nguyễn Ngọc Nga	
Trần Tuyết Canh	Phòng ban	G0001	Phòng ban khác
Nguyễn Kim Ngọc	Địa chỉ	19 Hùng Vương P4 Q5	
Trương Duy Hùng	Điện thoại	8356153	Phái (x : Nam) ☐
Lương Bá Thắng	Ngày sinh	12/ 5/1979	Tôn giáo ☐ Phật Giáo
Lâm Sơn Hoàng	Chức vụ	GD Nhân sự	
Nguyễn Minh Hồng	Nơi sinh	TPHCM	Dân tộc ☐ Kinh
Vương Ngọc Lan	Nguyên quán	TP Hồ Chí Minh	Quốc tịch ☐ Việt Nam
Nguyễn Hoàng Anh	Số CMND	012444564	Ngày cấp ☐ 31/12/1957
			Nơi cấp ☐ CA Q5
Exit			

Hình 15.1 Giao diện chương trình khi mới thi hành.

Thông tin nhân viên			
Nguyễn Ngọc Nga	Mã nhân viên	N002	
Hà Vinh Phát	Họ và tên	Hà Vinh Phát	
Trần Tuyết Canh	Phòng ban	G0002	Phòng ban khác
Nguyễn Kim Ngọc	Địa chỉ	89 Đồng Khởi Q1	
Trương Duy Hùng	Điện thoại	8352074	Phái (x : Nam) ☒
Lương Bá Thắng	Ngày sinh	20/01/1975	Tôn giáo ☐ Phật Giáo
Lâm Sơn Hoàng	Chức vụ	GD Kinh doanh	
Nguyễn Minh Hồng	Nơi sinh	An Giang	Dân tộc ☐ Kinh
Vương Ngọc Lan	Nguyên quán	An Giang	Quốc tịch ☐ Việt Nam
Nguyễn Hoàng Anh	Số CMND	024567893	Ngày cấp ☐ 31/12/1957
			Nơi cấp ☐ CA Q1
Exit			

Hình 15.2 Xem thông tin nhân viên Hà Vinh Phát.

CHÚ Ý

Form phải bảo đảm đáp ứng các yêu cầu sau:

- ◆ Người dùng không thể chỉnh sửa các thông tin nhân viên hiển thị ở bên phải.
- ◆ Mặc dù trong table DMPhongban chỉ có field mapb (mã phòng) mà không có field tenpb (tên phòng ban), nhưng trên form thì hiển thị đủ cả 2 field.

Hướng dẫn thiết kế

Bước 1. Tạo cơ sở dữ liệu Access Nhansu.mdb.

Định nghĩa cấu trúc và cập nhật nội dung 2 table DMPhongban và DMNhanvien trong cơ sở dữ liệu Access C:\BT\VBNet\Nhansu.mdb (có thể xem lại cách tạo và cập nhật dữ liệu trong các bài thực hành trước) như nhóm Hình 15.2.

Microsoft Access - [DMNhanVien : Table]		
File Edit View Insert Tools Window Help		
	Field Name	Data Type
	MaNV	Text
	MaPB	Text
	Hoten	Text
	Phai	Yes/No
	Chucvu	Text
	Ngaysinh	Date/Time
	Noisinh	Text
	Quoctich	Text
	Ngquan	Text
	Dtoc	Text
	Tgiaio	Text
	DChi	Text
	DienThoai	Text
	SoCMND	Text
	Noicap	Text
	Ngaycap	Date/Time

Microsoft Access - [DMPNhanVien : Table]

Mã NV	Mã PB	Họ và tên	Phái	Chức vụ	Ngày sinh	Nơi sinh	Địa chỉ
N001	G0001	Nguyễn Ngọc Nga	Nữ	GD Nhân sự	12/5/1970	TPHCM	19 Hồng Vương P4 Q5
N002	G0002	Hà Vinh Phát	Nam	GD Kinh doanh	2/14/1975	An Giang	89 Đồng Khởi Q1
N003	N0002	Trần Tuyết Canh	Nữ	Thư ký văn phòng	9/20/1978	Cần Thơ	26 Lê Quý Đôn P5 Q3
N004	K0001	Nguyễn Kim Ngọc	Nữ	Kế toán trưởng	6/1/1974	Hà Nội	178 Hữu Giang P6 Q6
N005	T0001	Trương Duy Hùng	Nam	Trưởng phòng Kỹ Thuật	3/10/1971	TP Hồ Chí Minh	77 Trương Định P5 Q3
N006	N0001	Lương Bá Thắng	Nam	Nhân viên	10/24/1980	Đồng Nai	92 Lê Thánh Tôn Q1
N007	N0001	Lâm Sơn Hoàng	Nam	Nhân viên	7/1/5/1979	TP Hồ Chí Minh	74 Kỳ Con Q1
N008	P0001	Nguyễn Minh Hồng	Nam	Phó Tổng GD	11/30/1965	Hà Nội	224 Lạc Long Quân Q11
N009	M0001	Vương Ngọc Lan	Nữ	Trưởng phòng Marketing	1/31/1973	TP Hồ Chí Minh	227 Hai Bà Trưng Q5
N111	N0001	Nguyễn Hoàng Anh	Nữ	Nhân viên	9/20/1973	Long An	2 Nguyễn Công Trứ

Microsoft Access - [DMPhongBan : Table]

Mã PB	Tên PB
+ G0001	Phòng Nhân Sự
+ G0002	Phòng Kinh Doanh
+ K0001	Phòng Kế Toán
+ M0001	Phòng Marketing
+ N0001	Phòng Tài Chính
+ N0002	Phòng Thư Ký
+ P0001	Phòng GD
+ T0001	Phòng Kỹ Thuật

Microsoft Access - [DMNhanVien : Table]

Mã NV	Điện thoại	Quốc tịch	Nguyên quán	Dân tộc	Tôn giáo	Số CMND	Nơi cấp	Ngày cấp
N001	8356153	Việt Nam	TP Hồ Chí Minh	Kinh	Phật Giáo	012444564	CA Q5	4/12/1987
N002	8352074	Việt Nam	An Giang	Kinh	Phật Giáo	024567893	CA Q1	4/25/1991
N003	8557798	Việt Nam	Cần Thơ	Kinh	Thiên Chúa Giáo	020154678	CA Q3	12/30/1993
N004	8553278	Việt Nam	Hà Nội	Kinh	Thiên Chúa Giáo	025461023	CA HN	9/10/1990
N005	8940295	Việt Nam	TP Hồ Chí Minh	Kinh	Phật Giáo	020001348	CA Q3	2/15/1987
N006	8940549	Việt Nam	Quảng Bình	Kinh	Thiên Chúa Giáo	024236656	CA Q1	11/11/1995
N007	8740374	Việt Nam	Đồng Tháp	Kinh	Phật Giáo	021455124	CA Q11	12/30/1994
N008	8352578	Việt Nam	Hà Nội	Kinh	Thiên Chúa Giáo	024568782	CA HN	1/10/1997
N009	8651636	Việt Nam	TP Hồ Chí Minh	Kinh	Phật Giáo	024798865	CA Q3	10/14/1990
N111	8111111	Việt Nam	Long An	Kinh	Phật Giáo	052221111	CA Q5	9/20/1989

Microsoft Access - [DMPhongBan : Table]

File Edit View Insert Tools Window Help

	Field Name	Data Type
MaPB		Text
TenPB		Text

Hình 15.2 Cấu trúc và nội dung các table DMPhongban, DMNhanvien.

Bước 2. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➔ Programs ➔ Microsoft Visual Studio .Net ➔ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File ➔ New Project để gọi cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = **Bai15**
- Định thư mục lưu project: Location = **C:\BTVBNET**
- Nhấp tùy chọn **Close Solution**.
- Chọn Project types = **Visual Basic Projects**, xong nhấp biểu tượng **Windows Application** trong mục Templates.
- Xong nhấp OK.

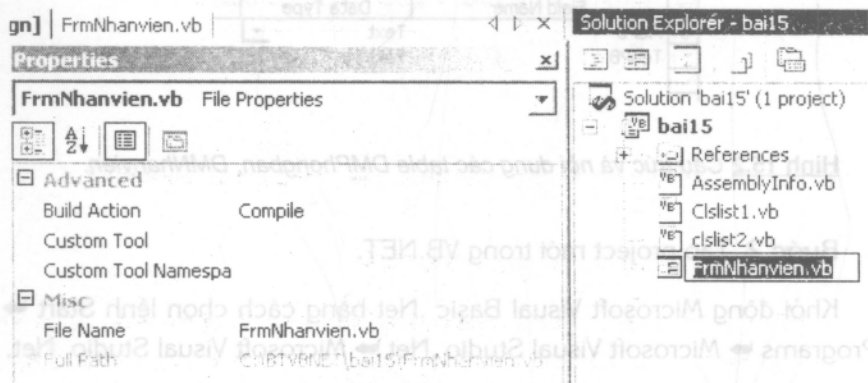
VB.NET sẽ tự động khởi tạo 1 project mới với 1 form trống mặc định là Form1.vb chứa trong thư mục BAI15, là thư mục con của C:\BTVBNET.

Bước 3. Tạo form FrmNV (file FrmNhanvien.vb) dùng tra cứu thông tin nhân viên.

Chọn lệnh View ➔ Solution Explorer kích hoạt cửa sổ Solution Explorer nếu chưa thấy.

- **Đặt lại tên tập tin cho form1:**

Name = FrmNhanvien.vb (Hình 15.3)



Hình 15.3 Đặt lại tên tập tin lưu form1.

- **Định lại các thuộc tính cho Form1** như sau:

Name = Frmnv

Text = Thông tin nhân viên

BackColor = ActiveCaptionText

ForeColor = ControlText

Size = 664, 400

- **Đặt lên form frmnv 1 điều khiển List box** và định lại các thuộc tính:

Name = Listbox1

TabIndex = 0

Font = VNI-Times, style Bold, size 10

Location = 8,8

Size = 192, 310

- **Đặt lên form 16 điều khiển Label** và định lại các thuộc tính dưới đây:

Label 1: Name = Label1
 Text = Mã nhân viên

Label 2: Name = Label2
 Text = Họ và tên

Label 3: Name = Label3
 Text = Phòng ban

Label 4: Name = Label4
 Text = Địa chỉ

Label 5: Name = Label5
 Text = Điện thoại

Label 6: Name = Label6
 Text = Ngày sinh

Label 7: Name = Label7
 Text = Chức vụ

Label 8: Name = Label8
 Text = Nơi sinh

Label 9: Name = Label9
 Text = Nguyên quán

Label 10: Name = Label10
 Text = Số CMND

Label 11: Name = Label11
 Text = Phái (x: Nam)

Label 12: Name = Label12

Text = Tôn giáo

Label 13: Name = Label13

Text = Dân tộc

Label 14: Name = Label14

Text = Quốc tịch

Label 15: Name = Label15

Text = Ngày cấp

Label 16: Name = Label16

Text = Nơi cấp

Các label đều có cùng thuộc tính:

Font = VNI-Times, style Bold, size 11

AutoSize = True

- **Đặt lên form 13 điều khiển Textbox** và định lại các thuộc tính như sau:

Textbox 1: Name = txtManv

TabIndex = 1

Textbox 2: Name = txtHoten

TabIndex = 2

Textbox 3: Name = txtMapb

TabIndex = 3

Textbox 4: Name = txtDiachi

TabIndex = 5

Textbox 5: Name = txtDienthoai

TabIndex = 6

Textbox 6: Name = txtTongiao

TabIndex = 9

Textbox 7: Name = txtChucvu

TabIndex = 10

Textbox 8: Name = txtNoisinh

TabIndex = 11

Textbox 9: Name = txtDantoc

TabIndex = 12

Textbox 10: Name = txtngquan

TabIndex = 13

Textbox 11: Name = txtquoctich

TabIndex = 14

Textbox 12: Name = txtcmnd

TabIndex = 15

Textbox 13: Name = txtnoicap

TabIndex = 17

Các Textbox đều có cùng thuộc tính:

Text = (trống)

Font = VNI-Times, Style Bold, Size 10

ReadOnly = True

- **Đặt lên form 1 điều khiển Combobox** và định lại các thuộc tính như sau:

Name = cmbmapb

DropDownStyle = Simple

Font = VNI-Times, style Bold, size 10

TabIndex = 4

Enabled = false

Text = (trống)

- **Đặt lên form 1 điều khiển Check box** và định lại các thuộc tính như sau:

Name = chkPhai

CheckAlign = MiddleCenter

Font = VNI-Times, style Bold, size 11

TabIndex = 7

Enabled = false

- **Đặt lên form 2 điều khiển DateTimePicker** và định lại các thuộc tính:

DateTimePicker 1:

Name = dtngaysinh

TabIndex = 8

DateTimePicker 2:

Name = dtngaycap

TabIndex = 16

Cả 2 điều khiển DateTimePicker đều có chung các thuộc tính sau:

Format = Short

Font = VNI-Times, style Bold, size 9

CalendarFont = VNI-Times, style Bold, size 9

Enabled = False

MinDate = 1/1/1753

MaxDate = 12/31/9998

- **Đặt lên form 1 điều khiển Listbox** và định lại các thuộc tính như sau:

Name = Listbox1

TabIndex = 0

Font = VNI-Times, style Bold, size 10

Location = 8,8

Size = 192, 310

- **Đặt lên form 1 điều khiển Button** và định lại các thuộc tính như sau:

Name = BtnExit

Text = E&xit

TabIndex = 21

Font = Microsoft Sans Serif, 10pt, style=Bold

Bước 4. Định nghĩa lớp clsList1.vb có các thuộc tính chứa dữ liệu đọc từ table DMNhanvien để đặt vào Listbox1.

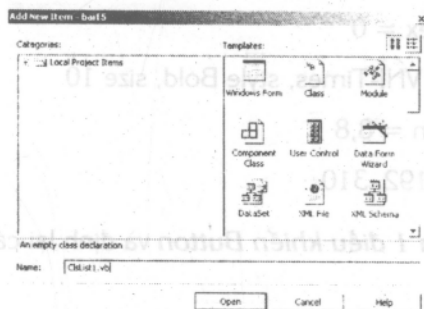
Lớp clsList1 phải có đủ các thuộc tính để chứa dữ liệu mà bạn cần đặt vào Listbox1, trong trường này là 3 thuộc tính. Trong đó, thuộc tính *hoten* được sử dụng để hiển thị dữ liệu trong phần văn bản của List box. Hai thuộc tính còn lại là *Mapb* và *Manv* dùng để lưu thông tin khóa chính của table DMNhanvien là Mã nhân viên và Mã phòng ban. Hai thuộc tính này còn giúp listbox1 dễ dàng xác định được record tương ứng với tên nhân viên đang được chọn trong listbox1. để lấy thông tin tương ứng ra form.

- Thêm 1 lớp mới vào project, bằng cách chọn lệnh:

Project ➡ Add Class

- Trong cửa sổ Add New Item – Bài15 (Hình 15.4), đặt cho lớp một cái tên, chẳng hạn:

Name = clsList1.vb, xong nhấn nút Open



Hình 15.4 Cửa sổ thêm lớp mới.

- Nhập vào mã lệnh định nghĩa lớp clsList1 có nội dung như sau:

Public Class clsList1

Private mstrhoten As String

Private mstrmanv As String

Private mstrmapb As String

Public Sub New()

End Sub

' Nạp giá trị cho 3 biến thành viên của lớp khi lớp được khởi tạo

Public Sub New(ByVal strhoten As String, ByVal strmanv As String,
ByVal strmapb As String)

mstrhoten = strhoten

mstrmanv = strmanv

mstrmapb = strmapb

End Sub

' Định nghĩa thuộc tính Manv

Property manv() As String

Get

Return mstrmanv

End Get

Set(ByVal Value As String)

mstrmanv = Value

End Set

End Property

‘ Định nghĩa thuộc tính Hoten

Property hoten() As String

Get

Return mstrhoten

End Get

Set(ByVal Value As String)

mstrhoten = Value

End Set

End Property

‘ Định nghĩa thuộc tính Mapb

Property mapb() As String

Get

Return mstrmapb

End Get

Set(ByVal Value As String)

mstrmapb = Value

End Set

End Property

' Định nghĩa lại phương thức ToString của lớp

Public Overrides Function ToString() As String

Return mstrhoten

End Function

End Class

CHÚ Ý

- ◆ Khi 1 Listbox hiển thị nội dung dữ liệu, nó luôn luôn gọi phương thức ToString để lấy dữ liệu. Do vậy bạn cần định nghĩa lại phương thức này để xác định giá trị mà Listbox1 hiển thị là tên nhân viên, chứ không phải mã nhân viên hay mã phòng ban.

Bước 5. Định nghĩa lớp clsList2.vb có các thuộc tính chứa dữ liệu đọc từ table DMPhongban để đặt vào Combo box **cmbMapb**.

Do table DMNhanvien chỉ có thông tin Mã phòng ban chứ không có thông tin tên phòng ban, nên bạn cần định nghĩa lớp clsList2 có đủ các thuộc tính để chứa thông tin phòng ban trong table DMPhongban. Làm như thế với mục đích cung cấp cho combo box **cmbMapb**, giúp combo box nhanh chóng hiển thị đúng tên phòng ban ứng với mã phòng ban của nhân viên đang được chọn.

Trong trường hợp này, lớp có 2 thuộc tính. Trong đó, thuộc tính **tenpb** được sử dụng để hiển thị dữ liệu trong phần văn bản của combo box. Thuộc tính **Mapb**, dùng để lưu thông tin khóa chính của table DMPhongban, và đó cũng là khóa ngoại của table DMNhanvien.

- Thêm một lớp mới vào project, bằng cách chọn lệnh:

Project ➡ Add Class...

- Đặt tên mới cho lớp:

Name = clsList2.vb, xong nhấp nút Open

- Nhập vào mã lệnh định nghĩa lớp clsList2 có nội dung như sau:

Public Class clslist2

Private mstrtenpb As String

Private mstrmapb As String

Public Sub New()

End Sub

' Nạp giá trị cho 2 biến thành viên của lớp khi lớp được khởi tạo

Public Sub New(ByVal strtenpb As String, ByVal strmapb As String)

mstrtenpb = strtenpb

mstrmapb = strmapb

End Sub

' Định nghĩa thuộc tính Tenpb

Property tenpb() As String

Get

Return mstrtenpb

End Get

Set(ByVal Value As String)

mstrtenpb = Value

End Set

End Property

' Định nghĩa thuộc tính Mapb

Property mapb() As String

Get

Return mstrmapb

End Get

Set(ByVal Value As String)


```

        mstrmapb = Value
    End Set

End Property

' Định nghĩa lại phương thức ToString của lớp
' Xác định thông tin hiển thị trên combo cmbMapb lấy từ
' thuộc tính Tenpb
Public Overrides Function ToString() As String
    Return mstrtenpb
End Function
End Class

```

Bước 6. Viết mã lệnh xử lý form *FrmNv.vb*.

```

Public Class FrmNv
    Inherits System.Windows.Forms.Form

    ' Vùng mã lệnh do VB.Net tự phát sinh trong quá trình bạn thiết kế form
#Region " Windows Form Designer generated code "
        ' Bạn không nên điều chỉnh tay các mã lệnh đang có trong vùng này
#End Region

    ' Thủ tục được gọi khi nạp form FrmNv
    Private Sub frmNv_Load(ByVal sender As System.Object, ByVal e As
        System.EventArgs) Handles MyBase.Load
        ' Gọi thủ tục deptload để nạp dữ liệu từ DMPhongban vào Combo
        ' cmbMapb
        deptload()

        ' Gọi thủ tục listload để nạp dữ liệu từ DMNhanvien vào Listbox
        Listbox1
        listload()
    End Sub

```

End Sub

' Thù tục nạp dữ liệu (họ tên nhân viên và thông tin khóa chính)

' từ table DMNhanvien vào Listbox1

Sub listload()

Dim ocmd As OleDb.OleDbCommand

Dim odr As OleDb.OleDbDataReader

Dim oitem As Clslist1

Dim strsql As String

' Loại cơ sở dữ liệu liên kết là Access và có tên là

' c:\btvbnet\nhansu.mdb

Dim strconn As String

strconn = connectStringbuild()

' Câu lệnh SQL xác định các field trong table DMNhanvien sẽ lấy

' dữ liệu

strsql = "select manv, mapb, hoten from DMNhanvien"

Try

ocmd = New OleDb.OleDbCommand()

' Mở kết nối tới cơ sở dữ liệu Access Nhansu.mdb

With ocmd

.Connection = New OleDb.OleDbConnection(strconn)

.Connection.Open()

' và thực hiện truy vấn lấy dữ liệu cho đối tượng DataReader odr

.CommandText = strsql

odr = .ExecuteReader(CommandBehavior.SequentialAccess)

End With

' Xóa trống nội dung Listbox1

ListBox1.Items.Clear()

' Vòng lặp đọc nội dung DataReader odr vào Listbox1

Do While odr.Read

' Khởi tạo đối tượng mới oitem thuộc lớp ClsList1

oitem = New Clslist1()

With odr

' Nạp thông tin record hiện hành vào các thuộc tính của Oitem

oitem.manv = .Item("manv")

oitem.mapb = .Item("mapb")

oitem.hoten = .Item("hoten")

End With

' Thêm vào listbox1

Listbox1.Items.Add(oitem)

Loop

' Nếu số mục trong Listbox1 > 0, nói cách khác table DMNhanvien có

' dữ liệu

If Listbox1.Items.Count > 0 Then

' Chọn mục đầu tiên trong Listbox1 làm mục được chọn hiện hành

' (nhân viên hiện hành đang hiển thị thông tin trên form)

Listbox1.SetSelected(0, True)

End If

' Bắt lỗi, nếu có lỗi xảy ra trong quá trình thi hành các lệnh trong thủ

Catch oExcept As Exception

' thì hiển thị thông báo lỗi

MessageBox.Show(oExcept.Message)

End Try

End Sub

' Hàm định nghĩa chuỗi xác định loại và tên cơ sở dữ liệu cần liên kết

Private Function connectStringbuild() As String

Dim strconn As String

strconn = "Provider=Microsoft.Jet.OLEDB.4.0;Password='';User
ID=Admin;Data Source=C:\btvbnet\nhansu.mdb"

Return strconn

End Function

' Thủ tục biến cố Click khi nhấp nút Exit hoặc nhấn tổ hợp phím Alt+X

Private Sub BtnExit_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles BtnExit.Click

Me.Close()

' Kết thúc chương trình

End Sub

' Thủ tục được gọi mỗi khi chọn tên nhân viên đang hiển thị trong Listbox1

Private Sub ListBox1_SelectedIndexChanged(ByVal sender As

System.Object, ByVal e As System.EventArgs) Handles

ListBox1.SelectedIndexChanged

' Gọi thủ tục Formshow, để hiển thị đầy đủ thông tin nhân viên

' có tên đang được chọn trong Listbox1 lên form

formshow()

End Sub

' Thủ tục hiển thị đầy đủ thông tin nhân viên

' có tên đang được chọn trong Listbox1 của form Frmnv

Private Sub formshow()

Dim ocmd As OleDb.OleDbCommand

Dim odr As OleDb.OleDbDataReader

Dim oitem As C#list1

Dim strSQL As String

Dim strconn As String

```
Dim ccmapb As String
```

```
strconn = connectStringbuild()
```

```
oitem = CType(ListBox1.SelectedItem, CIslist1)
```

```
' Câu lệnh SQL tìm nhân viên có tên đang được chọn trong listbox1
```

```
' theo mã nhân viên và mã phòng ban
```

```
strsql = "select * from DMNhanvien where manv = '" & oitem.manv & "'  
and mapb = '" & oitem.mapb & "'"
```

```
Try
```

```
ocmd = New OleDb.OleDbCommand()
```

```
With ocmd
```

```
.Connection = New OleDb.OleDbConnection(strconn)
```

```
.Connection.Open()
```

```
.CommandText = strsql
```

```
odr = .ExecuteReader(CommandBehavior.SequentialAccess)
```

```
End With
```

```
' Nếu tìm thấy thì đọc các thông tin của nhân viên đó lên form
```

```
' với thứ tự field đọc phải đúng như thứ tự field'
```

```
' bạn đã định nghĩa trong table DMNhanvien
```

```
If odr.Read() Then
```

```
With odr
```

```
txtManv.Text = .Item("Manv").ToString()
```

```
ccmapb = .Item("mapb").ToString()
```

```
txtmapb.Text = ccmapb.ToString()
```

```
txtHoten.Text = .Item("Hoten").ToString()
```

```
' Do field phai có kiểu Yes/No, nên cần ép kiểu thành Boolean
```

```
chkphai.Checked = CType(.Item("phai"), Boolean)
```

```
Txtchucvu.Text = .Item("chucvu").ToString()
```

*' Do field **dtngaysinh** có kiểu Date, nên cần ép thành kiểu Date*

`dtngaysinh.Value = CType(.Item("ngaysinh"), Date)`

`txtnoisinh.Text = .Item("noisinh").ToString()`

`txtquoctich.Text = .Item("quoctich").ToString()`

`txtngquan.Text = .Item("ngquan").ToString()`

`txtdantoc.Text = .Item("dtoc").ToString()`

`txttongiao.Text = .Item("tgiao").ToString()`

`txtdiachih.Text = .Item("dchi").ToString()`

`txtdienthoai.Text = .Item("dienthoai").ToString()`

`txtcmand.Text = .Item("socmnd").ToString()`

`txtnoicap.Text = .Item("noicap").ToString()`

*' Do field **dtngaysinh** có kiểu Date, nên cần ép thành kiểu Date*

`dtngaycap.Value = CType(.Item("ngaycap"), Date)`

' Gọi thủ tục finditem, hiển thị tên phòng tương ứng với

' mà phòng ban của nhân viên vừa được đọc thông tin

`Call finditem(cmbmapb, ccmapb)`

`End With`

`End If`

`odr.Close()`

`ocmd.Connection.Close()`

' Bắt các lỗi có thể xảy ra trong quá trình thực thi thủ tục

`Catch oException As Exception`

' Nếu có lỗi xảy ra thì hiển thị thông báo lỗi tương ứng

`MessageBox.Show(oException.Message)`

`End Try`

End Sub

*' Thù tục nạp dữ liệu (Tên phòng ban và thông tin khóa chính Mapb)
' từ table DMPhongban vào Combo CmbMapb*

Private Sub deptload()

Dim ocmd As OleDb.OleDbCommand

Dim odr As OleDb.OleDbDataReader

Dim oitem As clslist2

Dim strsql As String

Dim strconn As String

strconn = connectStringbuild()

' Câu lệnh SQL lấy thông tin các phòng ban trong table DMPhongban

strsql = "select mapb,tenpb from DMPhongban"

Try

ocmd = New OleDb.OleDbCommand()

With ocmd

.Connection = New OleDb.OleDbConnection(strconn)

.Connection.Open()

.CommandText = strsql

odr = .ExecuteReader(CommandBehavior.CloseConnection)

End With

Do While odr.Read()

' Khởi tạo đối tượng mới oitem thuộc lớp ClsList2

oitem = New clslist2()

' Nạp thông tin record hiện hành (phòng ban) vào các thuộc tính

' của Oitem

With odr

oitem.mapb = .Item("mapb")

oitem.tenpb = .Item("tenpb")

End With

' Thêm vào combo cmbmapb

cmbmapb.Items.Add(oitem)

Loop

odr.Close()

Catch oExcept As Exception

MessageBox.Show(oExcept.Message)

End Try

End Sub

' Hiển thị tên phòng tương ứng với mã phòng ban

' của nhân viên đang hiển thị thông tin trên form

Private Sub finditem(ByVal cmbcombo As ComboBox, ByVal strmapb As String)

Dim intloop As Integer

Dim bfound As Boolean

Dim oitem2 As clslist2

oitem2 = New clslist2()

' Vòng lặp duyệt danh sách các đối tượng trong Combo box

' cmbCombo

' Lưu ý cmbCombo là tham trị được cmbMapb truyền vào khi thủ tục

' được gọi

For intloop = 0 To cmbcombo.Items.Count - 1

oitem2 = CType(cmbcombo.Items(intloop), clslist2)

' nếu duyệt thấy mã phòng ban đang hiển thị tương ứng

' với đối tượng đang được duyệt trong danh sách

If oitem2.mapb = strmapb Then

' chọn đối tượng đang duyệt

cmbcombo.SelectedIndex = intloop

' Kết thúc công việc duyệt

bfound = True

Exit For

End If

Next

' Nếu không có đối tượng nào trong combo giống mã phòng ban,

' thì không chọn đối tượng nào cả

If Not bfound Then

cmbcombo.SelectedIndex = -1

End If

End Sub

End Class

MÁCH NƯỚC

- ♦ Chú ý rằng trong thiết kế đã trình bày, tên form thiết kế là *Frmnu*, cũng chính là tên lớp. Còn tập tin để lưu form này lại có tên *FrmNhanvien.vb*. Thông thường, chúng ta hay đặt tên form giống tên tập tin lưu form. Tuy nhiên, trong trường chúng tôi cố tình đặt khác nhau, để bạn dễ phân biệt và nhận ra sự khác biệt của 2 đặc điểm này.
 - ♦ Trong đoạn mã lệnh trên, để truy xuất dữ liệu trong các table Access, bạn đã sử dụng đối tượng ADO.Net DataReader, đây là loại con trỏ truy xuất tuần tự (SequentialAccess), cho phép bạn đọc nhanh dữ liệu trong các table hơn khi sử dụng các đối tượng truy xuất dữ liệu khác. Tuy nhiên, bạn cần chú ý nó đòi hỏi bạn phải lấy thông tin các cột (field) theo thứ tự bạn đã định nghĩa trong câu lệnh Select.
 - ♦ Trong trường hợp câu lệnh Select của bạn không xác định cụ thể tên field cần lấy, mà dùng Select * from DMNhanvien, thì thứ tự truy xuất thông tin phải theo đúng thứ tự các field mà bạn đã định nghĩa trong cấu trúc table DMNhanvien.
-

Ví dụ: Bạn sẽ gặp ngay thông báo lỗi nếu sửa lại thứ tự các câu lệnh truy xuất thông tin table DMNhanvien trong đoạn mã lệnh trên như sau:

With odr

```
ccmapb = .Item("mapb").ToString()
txtManv.Text = .Item("Mapv").ToString()
txtmapb.Text = ccmapb.ToString()
txtHoten.Text = .Item("Hoten").ToString()
chkphai.Checked = CType(.Item("phai"), Boolean)
```

.....

End With

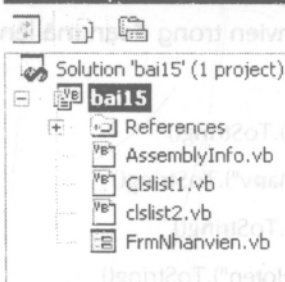
Bởi lẽ field **Manv** được định nghĩa trước field **Mapb** trong cấu trúc table DMNhanvien, nên cần truy xuất thông tin trước.

- Kết quả của sổ thiết kế form *Frmnv* sẽ như Hình 15.5.

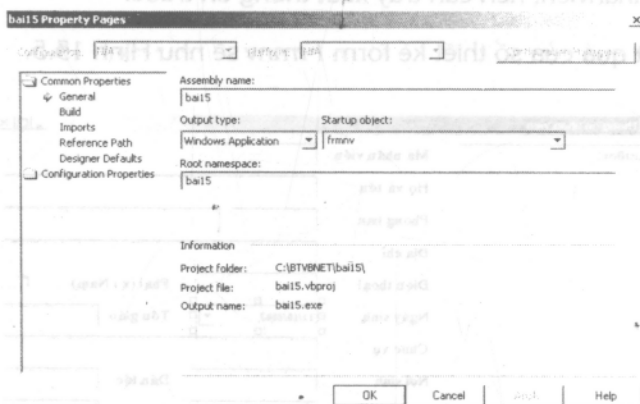
Hình 15.5 Cửa sổ thiết kế form *Frmnv*.

- Kết quả của sổ Solution Explorer sẽ như Hình 15.6.

Solution Explorer - bai15

**Hình 15.6** Cửa sổ Solution Explorer sau khi thiết kế xong form *Frmnv*.

- Đặt form *Frmnv* làm form khởi động bằng cách nhấp nút phải mouse lên project *Bai15* trong cửa sổ Solution Explorer, chọn lệnh Properties, định Startup Object = *Frmnv* (Hình 15.7).

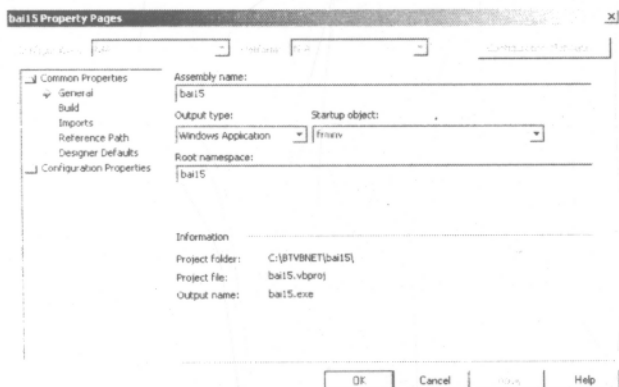
**Hình 15.7** Đặt form khởi động chương trình.

Bước 4. Chạy thử chương trình để xem kết quả.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start chạy thử chương trình. Ban đầu, bạn sẽ được một màn hình trống. Nhấp nút Load thông tin phòng ban đầu tiên và các nhân viên trong phòng sẽ hiển thị trên form như Hình 15.8.



Hình 15.8 Màn hình form Frmnv khi vừa thi hành.

Nhấp nút Exit hay nhấn tổ hợp phím Alt+X để kết thúc chương trình.

Bài 16.

Chương trình cơ sở dữ liệu – Thiết kế form cập nhật thông tin nhân viên

Mục đích chương trình

Bài thực hành này giúp bạn làm quen với kỹ thuật truy xuất (xem, thêm, xóa, sửa, ...) nội dung 1 table thông qua con đường viết mã lệnh, tìm hiểu cách sử dụng các đối tượng truy xuất dữ liệu mới trong VB.Net như DataTable, DataSet, từ đó hoàn thiện hơn khả năng thiết kế các form cập nhật và tra cứu dữ liệu.

Hoạt động chương trình

Hoạt động của chương trình tương tự như bài 15, tuy nhiên ngoài khả năng giúp tra cứu thông tin nhân viên, giờ đây người dùng còn có thể tiến hành thêm nhân viên mới, loại bỏ nhân viên khỏi table DMNhanvien hoặc điều chỉnh thông tin các nhân viên đang tồn tại.

Khi thi hành chương trình, bạn sẽ được màn hình có giao diện như Hình 16.1. Chương trình sẽ tự động nạp danh sách tên các nhân viên trong table DMNhanvien vào listbox bên trái và hiển thị đầy đủ thông tin của nhân viên đầu tiên trong listbox ở bên phải.

Thông tin nhân viên

Nguyễn Ngọc Nga	Mã nhân viên	N001	
Hà Vinh Phát	Họ và tên	Nguyễn Ngọc Nga	
Tôn Tuyết Cánh	Phòng ban	Phòng Nhân Sự	
Nguyễn Kim Ngọc	Địa chỉ	19 Hàng Vương P4 Q5	
Trương Duy Hùng	Điện thoại	8356153	Phái (x : Nam) ☐
Lương Bá Thắng	Ngày sinh	12/5/1978	Tôn giáo ☐ Phật Giáo
Lâm Sơn Hoàng	Chức vụ	GD Nhân sự	
Nguyễn Minh Hồng	Nơi sinh	TP HCM	Dân tộc ☐ Kinh
Vương Ngọc Lan	Nguyên quán	TP Hồ Chí Minh	Quốc tịch ☐ Việt Nam
	Số CMND	012444564	Ngày cấp ☐ 4/12/1987
			Nơi cấp ☐ CA Q5

Hình 16.1 Giao diện chương trình khi mới thi hành.

Để xem thông tin của nhân viên khác, người dùng chỉ việc nhấp vào tên nhân viên trong listbox, hoặc dùng các phím mũi tên để cuộn danh sách lên hoặc xuống cho đến khi thanh sáng dịch chuyển đến vị trí có tên nhân viên đó.

Add • Clear: Để bắt đầu nhập thông tin nhân viên mới (Hình 16.2).

Thông tin nhân viên

Nguyễn Ngọc Nga	Mã nhân viên		
Hà Vinh Phát	Họ và tên		
Tôn Tuyết Cánh	Phòng ban	Phòng Nhân Sự	
Nguyễn Kim Ngọc	Địa chỉ		
Trương Duy Hùng	Điện thoại		Phái (x : Nam) ☐
Lương Bá Thắng	Ngày sinh	11/12/2003	Tôn giáo ☐
Lâm Sơn Hoàng	Chức vụ		
Nguyễn Minh Hồng	Nơi sinh		Dân tộc ☐
Vương Ngọc Lan	Nguyên quán		Quốc tịch ☐
	Số CMND		Ngày cấp ☐ 11/12/2003
			Nơi cấp ☐

Hình 16.2 Màn hình nhập nhân viên mới.

- clear*
- **Add:** Để cập nhật nhân viên mới vào cơ sở dữ liệu. Ngược lại, chỉ cần nhấp tên 1 nhân viên khác trong listbox để hủy bỏ không lưu thông tin của nhân viên mới đang nhập.
- lưu*
- **Update:** để lưu các điều chỉnh thông tin nhân viên hiện hành.
 - **Delete:** để xóa nhân viên hiện hành khỏi table DMNhanvien.

CHÚ Ý

Form phải bảo đảm đáp ứng các yêu cầu sau:

- ♦ Danh sách phòng ban trong combo Mapb được nạp từ table DmPhongban, người dùng có thể truy xuất combo này để thay đổi phòng cho nhân viên hiện hành hoặc chọn phòng ban cho nhân viên mới (Hình 16.3).

Phòng ban	Phòng Nhân Sự
Địa chỉ	Phòng Nhân Sự
	Phòng Kinh Doanh
Điện thoại	Phòng Kế Toán
	Phòng Marketing
Ngày sinh	Phòng Tài Chính
	Phòng Thư Ký
Chức vụ	Phòng GD
	Phòng Kỹ Thuật

Hình 16.3 Chọn phòng ban từ combo cmbmapb.

- ♦ Việc thiết kế Form giả sử mã nhân viên trong toàn công ty là duy nhất, không trùng nhau. Nói cách khác có thể xem field Manv là khóa chính, thay vì cần tới 2 field Manv, Mapb làm khóa chính, như đã thiết kế trong cơ sở dữ liệu Access.
- ♦ Người dùng có thể nhập trực tiếp ngày sinh và ngày cấp CMND hoặc chọn nó từ điều khiển lịch (Hình 16.4).

Ngày sinh	11/ 5/2003	Tôn g																																																	
Chức vụ	November, 2003																																																		
Nơi sinh	<table border="1"> <thead> <tr> <th>Sun</th> <th>Mon</th> <th>Tue</th> <th>Wed</th> <th>Thu</th> <th>Fri</th> <th>Sat</th> </tr> </thead> <tbody> <tr> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>31</td> <td>1</td> </tr> <tr> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> </tr> <tr> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> </tr> <tr> <td>16</td> <td>17</td> <td>18</td> <td>19</td> <td>20</td> <td>21</td> <td>22</td> </tr> <tr> <td>23</td> <td>24</td> <td>25</td> <td>26</td> <td>27</td> <td>28</td> <td>29</td> </tr> <tr> <td>30</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> </tr> </tbody> </table>		Sun	Mon	Tue	Wed	Thu	Fri	Sat	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																													
26	27	28	29	30	31	1																																													
2	3	4	5	6	7	8																																													
9	10	11	12	13	14	15																																													
16	17	18	19	20	21	22																																													
23	24	25	26	27	28	29																																													
30	1	2	3	4	5	6																																													
Nguyên quán																																																			
Số CMND																																																			
Delete	Cle	Today: 11/5/2003																																																	

Hình 16.4 Chọn ngày sinh hoặc ngày cấp CMND.

Hướng dẫn thiết kế

Bước 1. Tạo cơ sở dữ liệu Access Nhansu.mdb.

Sử dụng lại cấu trúc và nội dung 2 table DMPhongban và DMNhanvien trong cơ sở dữ liệu Access C:\BTVBNet\Nhansu.mdb (đã thiết kế trong bài 15), với lưu ý Mã nhân viên không trùng nhau, kể cả những nhân viên khác phòng ban.

Bước 2. Tạo project mới trong VB.NET.

Khởi động Microsoft Visual Basic .Net bằng cách chọn lệnh Start ➤ Programs ➤ Microsoft Visual Studio .Net ➤ Microsoft Visual Studio .Net.

Nhấp nút New Project hay chọn lệnh File ➤ New Project để gọi cửa sổ New Project.

- Đặt tên tập tin lưu project: Name = **Bai16**
- Định thư mục lưu project: Location = **C:\BTVBNET**

- Nhấp tùy chọn Close Solution.
- Chọn Project types = *Visual Basic Projects*, xong nhấp biểu tượng *Windows Application* trong mục *Templates*.
- Xong nhấp OK.

VB.NET sẽ tự động khởi tạo một project mới với một form trống mặc định là *Form1.vb* chứa trong thư mục *BAI16*, là thư mục con của *C:\BTVBNET*.

Bước 3. Tạo form *FrmNV* (file *FrmNhanvien.vb*) dùng tra cứu, cập nhật thông tin nhân viên.

Chọn lệnh View ➡ Solution Explorer kích hoạt của sổ Solution Explorer nếu chưa thấy.

- **Đặt lại tên tập tin cho form1:**

Name = *FrmNhanvien.vb* (Hình 16.5)



Hình 16.5 Đặt lại tên tập tin lưu form1.

- **Định lại các thuộc tính cho Form1 như sau:**

Name = *Frmnv*

Text = *Thông tin nhan vien*

BackColor = ActiveCaptionText

ForeColor = ControlText

Size = 664, 400

- **Đặt lên form frmnv 1 điều khiển List box** và định lại các thuộc tính như sau:

Name = Listbox1

TabIndex = 0

Font = VNI-Times, style Bold, size 10

Location = 8,8

Size = 192, 310

- **Đặt lên form 1 điều khiển Combobox** và định lại các thuộc tính như sau:

Name = cmbmapb

DropDownStyle = Simple

Font = VNI-Times, style Bold, size 10

TabIndex = 4

Enabled = True

Text = (trống)

- **Đặt lên form 16 điều khiển Label** và định lại các thuộc tính y như đã trình bày trong bài 15.
- **Tương tự đặt lên form 13 điều khiển Textbox, 1 Checkbox và 2 DateTimePicker**, xong định lại các thuộc tính tên gọi của chúng như bài 15.

Riêng thuộc tính Enabled = True (cho phép truy xuất điều khiển)

- **Đặt lên form 5 điều khiển Button** và định lại các thuộc tính như sau:

Button 1	Name = BtnExit Text = E&xit TabIndex = 25 Font = Microsoft Sans Serif, 10pt, style=Bold
Button 2	Name = BtnAdd Text = &Add TabIndex = 21 Font = Microsoft Sans Serif, 10pt, style=Bold
Button 3	Name = BtnUpdate Text = &Update TabIndex = 22 Font = Microsoft Sans Serif, 10pt, style=Bold
Button 4	Name = BtnDelete Text = &Delete TabIndex = 23 Font = Microsoft Sans Serif, 10pt, style=Bold
Button 5	Name = BtnClear Text = &Clear TabIndex = 24 Font = Microsoft Sans Serif, 10pt, style=Bold

Bước 4. Định nghĩa lớp clsList1.vb có các thuộc tính chứa dữ liệu đọc từ table DMNhanvien để đặt vào Listbox1.

Tương tự như bài 15, lớp `clsList1` phải có đủ các thuộc tính để chứa dữ liệu mà bạn cần đặt vào `Listbox1`. Trong đó, thuộc tính *hoten* được sử dụng để hiển thị dữ liệu trong phần văn bản của list box. Hai thuộc tính còn lại là *Mapb* và *Manv*, dùng để lưu thông tin khóa chính của table `DMNhanvien` là Mã nhân viên và Mã phòng ban, vốn sẽ giúp `listbox1` dễ dàng xác định được record tương ứng với tên nhân viên đang được chọn trong `listbox1`, nhằm lấy thông tin tương ứng ra form.

- Thực hiện các thao tác và nhập vào mã lệnh định nghĩa lớp `clsList1` có mã lệnh như trong Bài 15.

Public Class Clslist1

Private mstrhoten As String

Private mstrmanv As String

Private mstrmapb As String

.....

End Class

Bước 5. Định nghĩa lớp `clsList2.vb` có các thuộc tính chứa dữ liệu đọc từ table `DMPhongban` để đặt vào Combo box *cmbMapb*.

(xem lại bài 15)

Public Class clslist2

Private mstrtenpb As String

Private mstrmapb As String

.....

End Class

Bước 6. Viết mã lệnh xử lý form *FrmNv.vb*.

Public Class Frmnv

Inherits System.Windows.Forms.Form

' Vùng mã lệnh do VB.Net tự phát sinh trong quá trình bạn thiết kế form

#Region " Windows Form Designer generated code "

' Bạn không nên điều chỉnh tay các mã lệnh đang có trong vùng này

#End Region

' Thủ tục được gọi khi nạp form Frmnhv

Private Sub frmnhv_Load(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles MyBase.Load

' Gọi thủ tục deptload để nạp dữ liệu từ DMPhongban vào Combo

' cmbMapb

deptload()

' Gọi thủ tục DatasetCreate để khởi tạo đối tượng dataset MODS

' sẽ được sử dụng trong suốt quá trình vận hành form

DatasetCreate()

' Gọi thủ tục listload để nạp dữ liệu từ DMNhanvien vào Listbox

Listbox1

listload()

End Sub

' Khởi tạo đối tượng dataset MODS, dùng để truy xuất table DMNhanvien

' (đọc, ghi, sửa, xóa) trong suốt quá trình vận hành form

Private Sub DatasetCreate()

Dim oAdapter As OleDb.OleDbDataAdapter

Dim strSQL As String

Dim strConn As String

' Lấy chuỗi kết nối

strConn = connectStringbuild()

' Câu lệnh SQL truy vấn dữ liệu trong table DMNhanvien

strSQL = "Select * from DMNhanvien"

' Khởi tạo 1 đối tượng DataSet mới

mods = New DataSet()

Try

' Tạo đối tượng DataAdapter mới

oAdapter = New OleDb.OleDbDataAdapter(strSQL, strConn)

' Điền dữ liệu từ DataAdapter vào Dataset

' và đặt tên cho DataTable vừa tạo là DMNhanvien

oAdapter.Fill(mods, "DMNhanvien")

' Tạo khóa chính cho table DMNhanvien vừa tạo trong dataset

' MODS

With mods.Tables("DMNhanvien")

.PrimaryKey = New DataColumn() {.Columns("Manv")}

End With

' Bắt các lỗi xảy ra trong quá trình thi hành thủ tục

Catch oExcept As Exception

MessageBox.Show(oExcept.Message)

End Try

End Sub

GHI CHÚ

- ◆ Đối tượng DataTable chứa các record dữ liệu của 1 table, thường dùng để nạp giá trị cho các combo box hay list box.
 - ◆ Đối tượng DataSet có thể chứa 1 hay nhiều đối tượng DataTable và các mối quan hệ giữa chúng, cho phép thực hiện các công việc nạp dữ liệu cho combo, thêm, sửa hoặc xóa dữ liệu trên form.
 - ◆ Đối tượng DataAdapter giúp việc nạp dữ liệu từ cơ sở dữ liệu đến các đối tượng DataTable hoặc DataSet.
-

- Có thể tạo khóa chính cho table DMNhanvien vừa tạo trong dataset MODS (trong trường hợp này là *Manv*) khác với khóa chính của table DMNhanvien trong cơ sở dữ liệu Access (2 field *Manv*, *Mapb*), nhưng phải đảm bảo rằng không có nhân viên nào cùng mã số.
- Có thể tham chiếu tới 1 DataTable riêng bằng cách sử dụng tập hợp Tables. Chẳng hạn câu lệnh **MODS.Tables("DMNhanvien")** sẽ trả lại đối tượng DataTable tên là DMNhanvien, đối tượng này đã được tạo khi thực hiện phương thức Fill trong câu lệnh **oAdapter.Fill(mods, "DMNhanvien")**.
- Có thể tạo khóa chính cho 1 table trong Dataset với nhiều hơn 1 field.

Ví dụ: Mã lệnh sau sẽ tạo khóa chính cho table DMNhanvien trong đối tượng dataset MODS gồm 2 field *Manv* và *Mapb*.

```
mods.Tables("DMNhanvien").PrimaryKey = New DataColumn()
    {Columns("Manv"), Columns("Mapb")}
```

- Nếu bạn sửa khóa chính của đối tượng DataTable DMNhanvien từ 1 field *Manv* thành 2 field *Manv* và *Mapb*, bạn phải định nghĩa lại các câu lệnh tìm kiếm dữ liệu trong DataTable này (phương thức *Find*) đã được trình bày trong các thủ tục bên dưới cho phù hợp.

Thủ tục nạp dữ liệu từ đối tượng dataset MODS vào listbox LISTBOX1

Sub listload()

Dim oitem As Clslist1

Dim orow As DataRow

Xóa trống ListBox1

ListBox1.Items.Clear()

Vòng lặp duyệt từng record trong dataTable DMNhanvien thuộc

dataset MODS

```
For Each orow In mods.Tables("DMNhanvien").Rows
```

```
    ' tạo đối tượng clsList1 mới để lưu họ tên, mã nhân viên,
```

```
    ' mã phòng ban của record hiện hành
```

```
    oitem = New Clslist1()
```

```
    With oitem
```

```
        .manv = orow.Item("manv")
```

```
        .mapb = orow.Item("mapb")
```

```
        .hoten = orow.Item("hoten").ToString()
```

```
    End With
```

```
    ' Thêm đối tượng clslist1 vào ListBox1
```

```
    ListBox1.Items.Add(oitem)
```

```
Next
```

```
    ' Nếu số mục trong ListBox1 > 0, nói cách khác table DMNhanvien
```

```
    ' có dữ liệu
```

```
If ListBox1.Items.Count > 0 Then
```

```
    ' Chọn mục đầu tiên trong ListBox1 làm mục được chọn hiện hành
```

```
    ' (nhân viên hiện hành đang hiển thị thông tin trên form)
```

```
    ListBox1.SetSelected(0, True)
```

```
End If
```

End Sub

```
    ' Hàm định nghĩa chuỗi xác định loại và tên cơ sở dữ liệu cần liên kết
```

```
Private Function connectStringbuild() As String
```

```
    Dim strconn As String
```

```
    ' Bạn cần đặt cơ sở dữ liệu Access nhansu.mdb theo đúng đường dẫn
```

```
    ' dưới đây
```

```
    strconn = "Provider=Microsoft.Jet.OLEDB.4.0;Password='';User
```

```
    ID=Admin;Data Source=C:\btvbnet\nhansu.mdb"
```


Return strconn

End Function

Thủ tục biến cố Click khi nhấp nút Exit hoặc nhấn tổ hợp phím Alt+X

```
Private Sub BtnExit_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles BtnExit.Click
    Me.Close()
```

End Sub

Thủ tục được gọi mỗi khi chọn tên nhân viên đang hiển thị trong ListBox1

```
Private Sub ListBox1_SelectedIndexChanged(ByVal sender As
    System.Object, ByVal e As System.EventArgs) Handles
    ListBox1.SelectedIndexChanged
```

*Gọi thủ tục Formshow, để hiển thị đầy đủ thông tin nhân viên
có tên đang được chọn trong ListBox1 lên form
formshow()*

End Sub

*Tìm record xác định trong Dataset, lấy thông tin và hiển thị lên
form Frmnv*

Private Sub formshow()

```
Dim odr As DataRow
Dim oitem As CList1
Dim cmapb As String
' Lấy thông tin khoa chính từ listbox ListBox1
oitem = CType(ListBox1.SelectedItem, CList1)
' Tìm record (nhân viên) trong dataset MODS '
' theo giá trị khóa chính (mã nhân viên)
With mods.Tables("DMNhanvien").Rows
    odr = .Find(oitem.manv)
```

End With

' Hiển thị dữ liệu tìm thấy lên form

txtManv.Text = odr("Manv").ToString()

ccmapb = odr("mapb").ToString()

txtHoten.Text = odr("Hoten").ToString()

chkphai.Checked = CType(odr("phai"), Boolean)

Txtchucvu.Text = odr("chucvu").ToString()

dngaysinh.Value = CType(odr("ngaysinh"), Date)

txtnoisinh.Text = odr("noisinh").ToString()

txtquoctich.Text = odr("quoctich").ToString()

txtngquan.Text = odr("ngquan").ToString()

txtdantoc.Text = odr("dtoc").ToString()

txttongiao.Text = odr("tgiao").ToString()

txtdiachi.Text = odr("dchi").ToString()

txtdienthoai.Text = odr("dienthoai").ToString()

txtcmnd.Text = odr("socmnd").ToString()

txtnoicap.Text = odr("noicap").ToString()

dngaycap.Value = CType(odr("ngaycap"), Date)

' Gọi thủ tục finditem, hiển thị tên phòng tương ứng với mã phòng ban

' của nhân viên vừa tìm thấy

Call finditem(cmbmapb, ccmapb)

End Sub

Nhận xét:

- Mã lệnh truy xuất dữ liệu table DMNhanvien với Dataset ngắn gọn hơn khi sử dụng DataReader, tuy nhiên tốc độ truy xuất dữ liệu với Dataset thì chậm hơn.
- Bạn chỉ có thể gọi phương thức Find khi đã đặt khóa chính với thuộc tính PrimaryKey trên DataTable.

- Thu tục nạp dữ liệu (Tên phòng ban và thông tin khóa chính Mapb)
- từ table DMPhongban vào Combo CmbMapb

Private Sub deptload()

```

Dim oAdapter As OleDb.OleDbDataAdapter
Dim oTable As DataTable = New DataTable()
Dim oitem As clslist2
Dim strSQL As String
Dim strconn As String
Dim intloop As Integer
strconn = connectStringbuild()
' Câu lệnh truy vấn dữ liệu trong DMPhongban
strSQL = "select mapb,tenpb from DMPhongban"
Try
    oAdapter = New OleDb.OleDbDataAdapter(strSQL, strconn)
    ' Điền dữ liệu từ table DMPhongban vào datatable oTable
    oAdapter.Fill(oTable)
    ' Vòng lặp duyệt từng record trong datatable oTable
    For intloop = 0 To oTable.Rows.Count - 1
        oitem = New clslist2()
        With oTable.Rows(intloop)
            oitem.tenpb = .Item("Tenpb").ToString()
            oitem.mapb = .Item("Mapb").ToString()
        End With
        ' Thêm đối tượng oitem vào combo cmdmapb
        cmbmapb.Items.Add(oitem)
    Next
Catch oExcept As Exception
    MessageBox.Show(oExcept.Message)

```

End Try

End Sub

Nhận xét:

- Có thể truy xuất dữ liệu trong các table đơn với đối tượng DataTable mà không cần phải định nghĩa đối tượng Dataset.

‘ Hiển thị tên phòng tương ứng với mã phòng ban

‘ của nhân viên đang hiển thị thông tin trên form

```
Private Sub finditem(ByVal cmbcombo As ComboBox, ByVal strmapb
As String)
```

```
Dim intloop As Integer
```

```
Dim bfound As Boolean
```

```
Dim oitem2 As clslist2
```

```
oitem2 = New clslist2()
```

‘ Vòng lặp duyệt danh sách các đối tượng clslist2 trong combo

‘ cmbCombo

‘ cmbCombo là tham trị được cmbMapb truyền vào khi thủ tục được gọi

```
For intloop = 0 To cmbcombo.Items.Count - 1
```

```
oitem2 = CType(cmbcombo.Items(intloop), clslist2)
```

‘ nếu mã phòng ban đang hiển thị tương ứng

‘ với mã phòng ban chứa trong đối tượng clslist2 đang được duyệt

```
If oitem2.mapb = strmapb Then
```

‘ chọn đối tượng clslist2 đang duyệt

```
cmbcombo.SelectedIndex = intloop
```

```
bfound = True
```

‘ Kết thúc công việc duyệt

```
Exit For
```

```
End If
```

```
Next
```

```

If Not bround Then
    cmbcombo.SelectedIndex = -1
End If

```

End Sub

** Khi nhấp nút Add hay nhấn tổ hợp phím Alt+A*

```

Private Sub BtnAdd_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button1.Click
    ' gọi thủ tục thêm record vào table DMNhanvien
    dataAdd()

```

End Sub

** Thêm 1 record vào Dataset hay tạo 1 table trong cơ sở dữ liệu*

Private Sub dataAdd()

```

Dim oAdapter As OleDb.OleDbDataAdapter
Dim oBuild As OleDb.OleDbCommandBuilder
Dim odr As DataRow
Dim strSQL As String
Dim strconn As String
' Tạo 1 đối tượng DataRow (chứa thông tin record) mới từ Dataset
odr = mods.Tables("DMNhanvien").NewRow()
odr.BeginEdit()
' Nạp dữ liệu mới vào record
odr("Manv") = txtManv.Text
odr("mapb") = CType(cmbmapb.SelectedItem, clslist2).mapb
odr("Hoten") = txtHoten.Text
odr("phai") = CType(chkphai.Checked, Boolean)
odr("chucvu") = txtchucvu.Text
odr("ngaysinh") = CType(dtnghaysinh.Value, Date)
odr("norsinh") = txtnorsinh.Text

```

```
odr("quoctich") = txtquoctich.Text
```

```
odr("ngquan") = txtngquan.Text
```

```
odr("dtoc") = txtlantoc.Text
```

```
odr("tgiao") = txttongiao.Text
```

```
odr("dchi") = txtdiachi.Text
```

```
odr("dienthoai") = txtvienthoai.Text
```

```
odr("socmnd") = txtcmnd.Text
```

```
odr("noicap") = txtnoicap.Text
```

```
odr("ngaycap") = CType(dngaycap.Value, Date)
```

```
' Bảo cho datarow biết đã hoàn thành việc thêm dữ liệu
```

```
odr.EndEdit()
```

```
' Thêm datarow ODR vào dataset MODS
```

```
mods.Tables("DMNhanvien").Rows.Add(odr)
```

```
Try
```

```
' Lấy chuỗi nối kết
```

```
strconn = connectStringbuild()
```

```
' Câu lệnh SQL truy xuất table DMNhanvien trong cơ sở dữ liệu
```

```
strsql = "Select * from DMNhanvien"
```

```
' Tạo đối tượng DataAdapter mới oAdapter
```

```
oAdapter = New OleDb.OleDbDataAdapter(strsql, strconn)
```

```
' Tạo đối tượng CommandBuild oBuild từ oAdapter
```

```
' cho phép thi hành các lệnh SQL thêm, sửa, xóa: INSERT,
```

```
' UPDATE, DELETE
```

```
oBuild = New OleDb.OleDbCommandBuilder(oAdapter)
```

```
' Lấy đối tượng lệnh SQL Insert
```

```
oAdapter.InsertCommand = oBuild.GetInsertCommand
```

```
' Thực thi việc chen record thuộc DataTable trong Dataset
```

```
' vào cơ sở dữ liệu thông qua Adapter
```

```

oAdapter.Update(mods, "DMNhanvien")
' Bảo cho Dataset biết các thay đổi dữ liệu nguồn đã hoàn thành
mods.AcceptChanges()
' Đóng kết nối
oAdapter.InsertCommand.Connection.Close()
' Tạo lại 1 đối tượng Dataset mới
DatasetCreate()
' Nạp lại nội dung table DMNhanvien vào Listbox1
listload()
Catch oException As Exception
    ' Nếu có lỗi xảy ra trong quá trình thi hành thủ tục
    ' thì hiển thị thông báo lỗi tương ứng
    MessageBox.Show(oException.Message)
End Try

```

End Sub

Giai thích mã lệnh:

- Để thêm 1 record mới vào table trong cơ sở dữ liệu, bạn cần thêm 1 record mới vào đối tượng DataTable trong Dataset. Có thể thực hiện điều này bằng phương thức NewRow để tạo đối tượng DataRow mới. Phương thức BeginEdit cho phép bạn cập nhật dữ liệu cho từng field một. Kết thúc việc cập nhật với phương thức EndEdit khi bạn đã cập nhật xong giá trị cho tất cả các field trong record mới. Thêm đối tượng DataRow mới vào DataTable trong Dataset bằng phương thức Add của tập hợp Rows trong đối tượng DataTable.
- Để gửi record mới trong Dataset tới cơ sở dữ liệu (table DMNhanvien), bạn phải thực thi câu lệnh SQL INSERT bằng cách sử dụng lớp OleDbCommandBuild để tạo đối tượng CommandBuilder.

' Khi nhấp nút Delete hay nhấn tổ hợp phím ALT+D

```
Private Sub BtnDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
```

```
' Gọi thủ tục xóa record hiện hành khỏi table DMNhanvien
    dataDelete()
```

End Sub

' Xóa 1 record khỏi Dataset

' nghĩa là xóa khỏi table DMNhanvien trong cơ sở dữ liệu

```
Private Sub dataDelete()
```

```
    Dim oAdapter As OleDb.OleDbDataAdapter
```

```
    Dim oBuild As OleDb.OleDbCommandBuilder
```

```
    Dim odr As DataRow
```

```
    Dim strsql As String
```

```
    Dim strconn As String
```

```
    Dim strmanv As String
```

```
' Lấy chuỗi nối kết
```

```
    strconn = connectStringbuild()
```

```
' Lấy giá trị khóa chính từ ListBox1
```

```
    strmanv = CType(ListBox1.SelectedItem, Clslist1).manv
```

```
' Tìm record trong DataTable DMNhanvien theo khóa chính
```

```
    odr = mods.Tables("DMNhanvien").Rows.Find(strmanv)
```

```
' đánh dấu xóa record được tìm thấy trong Dataset
```

```
    odr.Delete()
```

```
Try
```

```
' Câu lệnh SQL truy xuất table DMNhanvien trong cơ sở dữ liệu
```

```
    strsql = "Select * from DMNhanvien"
```

```
' Tạo đối tượng DataAdapter mới oAdapter
```



```

oAdapter = New OleDb.OleDbDataAdapter(strsql, strconn)
' Tạo đối tượng CommandBuild oBuild từ oAdapter
oBuild = New OleDb.OleDbCommandBuilder(oAdapter)
' Lấy đối tượng lệnh SQL Delete
oAdapter.DeleteCommand = oBuild.GetDeleteCommand
' Thực thi việc xóa record thuộc DataTable DMNhanvien trong
'   Dataset mods khỏi cơ sở dữ liệu thông qua Adapter
oAdapter.Update(mods, "DMNhanvien")
' Bảo cho dataset MODS biết quá trình xóa đã hoàn tất
mods.AcceptChanges()
' Đóng kết nối
oAdapter.DeleteCommand.Connection.Close()
' Nạp lại nội dung table DMNhanvien vào Listbox1
listload()

Catch oException As Exception
    ' Nếu có lỗi xảy ra trong quá trình thì hành thủ tục
    '   thì hiển thị thông báo lỗi tương ứng
    MessageBox.Show(oException.Message)

End Try

End Sub

' Khi nhấp nút Update hay nhấn tổ hợp phím ALT+U
Private Sub BtnUpdate_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button2.Click
    ' Gọi thủ tục cập nhật record hiện hành vào table DMNhanvien
    dataUpdated()

End Sub

' Cập nhật lại dữ liệu cho record hiện hành trong cơ sở dữ liệu

```

Private Sub dataUpdate()

```

Dim oAdapter As OleDb.OleDbDataAdapter
Dim oBuild As OleDb.OleDbCommandBuilder
Dim odr As DataRow
Dim strsql As String
Dim strconn As String
Dim strmanv As String
' Lấy giá trị khóa chính (mã nhân viên) từ Listbox1
strmanv = CType(ListBox1.SelectedItem, Clslist1).manv
' Tìm record trong DataTable DMNhanvien theo khóa chính
odr = mods.Tables("DMNhanvien").Rows.Find(strmanv)
' Bắt đầu quá trình điều chỉnh
odr.BeginEdit()
' Cập nhật dữ liệu mới cho từng field trong record hiện hành
odr("Manv") = txtManv.Text
odr("mapb") = CType(cmbmapb.SelectedItem, clslist2).mapb
odr("Hoten") = txtHoten.Text
odr("phai") = CType(chkphai.Checked, Boolean)
odr("chucvu") = Txtchucvu.Text
odr("ngaysinh") = CType(dtnghaysinh.Value, Date)
odr("noisinh") = txtnoisinh.Text
odr("quoctich") = txtquoctich.Text
odr("ngquan") = txtngquan.Text
odr("dtoc") = txtdtantoc.Text
odr("tgiao") = txttongiao.Text
odr("dchi") = txtdiachi.Text

```

```

odr("dienthoai") = txtDienthoai.Text
odr("socmnd") = txtcmnd.Text
odr("noicap") = txtnoicap.Text
odr("ngaycap") = CType(dngaycap.Value, Date)
' Kết thúc quá trình điều chỉnh
odr.EndEdit()

Try
    strconn = connectStringBuild()
    strsql = "Select * from DMNhanvien"
    ' Tạo đối tượng DataAdapter mới oAdapter
    oAdapter = New OleDb.OleDbDataAdapter(strsql, strconn)
    oBuild = New OleDb.OleDbCommandBuilder(oAdapter)
    ' Lấy đối tượng lệnh SQL Update
    oAdapter.UpdateCommand = oBuild.GetUpdateCommand
    ' Thực thi việc cập nhật record thuộc DataTable DMNhanvien trong
    '   Dataset mods vào cơ sở dữ liệu thông qua Adapter
    oAdapter.Update(mods, "DMNhanvien")
    mods.AcceptChanges()
    oAdapter.UpdateCommand.Connection.Close()
    listload()
Catch oException As Exception
    ' Nếu có lỗi xảy ra thì hiển thị thông báo lỗi tương ứng
    MessageBox.Show(oException.Message)
End Try

End Sub

' Khi nhấp nút Clear hay nhấn tổ hợp phím ALT+C
Private Sub BtnClear_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Button4.Click

```

- ' Xóa trống nội dung các Textbox để nhập thông tin nhân viên mới
- ' Riêng các điều khiển DateTimePicker thì đặt về ngày hiện hành của hệ thống

```
txtManv.Text = ""  
txtHoten.Text = ""  
Txtchucvu.Text = ""  
dtngaysinh.Value = CType(Date.Now(), Date)  
txtnoisinh.Text = ""  
txtquoctich.Text = ""  
txtingquan.Text = ""  
txtdantoc.Text = ""  
txttongiao.Text = ""  
txtdiachi.Text = ""  
txtdienthoai.Text = ""  
txtemnd.Text = ""  
txtnoicap.Text = ""  
dtngaycap.Value = CType(Date.Now(), Date)
```

End Sub

End Class

MÁCH NƯỚC

- ♦ Với các thủ tục ListLoad, DeptLoad và FormShow, bạn có thể vẫn sử dụng đối tượng DataReader, bằng cách bỏ nguyên xi các thủ tục này từ bài 15 sang, chương trình vẫn vận hành tốt. Tuy nhiên, bạn nên viết lại các thủ tục như đã trình bày trong bài 16 để thấy được điều khiển DataTable và DataSet cũng có thể giúp bạn đọc nội dung các table cho các combo box hay list box.
-

- Kết quả của số thiết kế form *Frmnv* sẽ như Hình 16.6.

The form 'Thông tin nhân viên' includes the following fields:

- Ma nhân viên (Employee ID)
- Họ và tên (Full Name)
- Phòng ban (Department) - dropdown menu
- Địa chỉ (Address)
- Điện thoại (Phone Number)
- Phái (Gender) - radio buttons (Nam, Nữ)
- Ngày sinh (Date of Birth) - date picker (11/12/2003)
- Tôn giáo (Religion)
- Chức vụ (Position)
- Nơi sinh (Place of Birth)
- Dân tộc (Ethnicity)
- Nguyên quán (Hometown)
- Quốc tịch (Nationality)
- Số CMND (ID Card Number)
- Ngày cấp (Issue Date) - date picker (11/12/2003)
- Nơi cấp (Issued at)

Buttons at the bottom: Add, Update, Delete, Clear, Exit.

Hình 16.6 Cửa sổ thiết kế form Frmnv.

- Đặt form Frmnv làm form khởi động bằng cách nhấp nút phải mouse lên project Bai15 trong cửa sổ Solution Explorer, chọn lệnh Properties, định Startup Object = Frmnv.

Bước 4. Chạy thử chương trình để xem kết quả.

Chọn lệnh File ➡ Save All lưu toàn bộ ứng dụng đã xây dựng.

Chọn lệnh Build ➡ Build Solution để biên dịch project.

Nhấn phím F5 hoặc chọn lệnh Debug ➡ Start chạy thử chương trình.

Bài tập nâng cao

Hãy cải tiến form Frmnv đã thiết kế trong bài 16, sao cho:

- Form có thể tìm kiếm dữ liệu (sử dụng phương thức Find của đối tượng DataSet) trong table DMNhanvien với khóa chính là 2 field Mapb, Manv; thay vì chỉ 1 field Manv như trong bài thực hành trên.

- Hiển thị câu cảnh báo “Bạn có chắc muốn xóa nhân viên này (Y/N) ?” để người dùng quyết định có xóa nhân viên hiện hành hay không, thay vì xóa ngay, rất dễ mất dữ liệu khi người dùng vô tình nhấn nhầm nút Delete.
- Trước khi lưu nhân viên mới có kiểm tra khóa chính, xem đã tồn tại trong table DMNhanvien hay chưa. Nếu chưa mới cho thêm mới, nếu rồi thì không cho thêm mới, tránh tình trạng ngắt chương trình giữa chừng do bị lỗi trong khi lưu.
- Khi điều chỉnh nội dung khóa chính (Manv, Mapb) của nhân viên đang tồn tại, form phải đảm bảo không trùng với giá trị khóa của các record đang tồn tại trong table DMNhanvien.

MỤC LỤC

Bài 1.	Cài đặt VB.NET trên Windows 2000 Server	5
Bài 2.	Chương trình Lunar	17
Bài 3.	Chương trình Menu	41
Bài 4.	Chương trình Calendar	65
Bài 5.	Chương trình Book	77
Bài 6.	Chương trình TextEditor	97
Bài 7.	Chương trình Notepad	125
Bài 8.	Chương trình cơ sở dữ liệu – Cập nhật danh mục phòng ban	147
Bài 9.	Chương trình Calculator	181
Bài 10.	Chương trình SystemTray	199
Bài 11.	Chương trình cơ sở dữ liệu – Thiết kế form đôi	215
Bài 12.	Chương trình Listbox	235
Bài 13.	Chương trình Dialog	271
Bài 14.	Chương trình cơ sở dữ liệu – Thiết kế form tra cứu nhân viên dạng 1	297
Bài 15.	Chương trình cơ sở dữ liệu – Thiết kế form tra cứu nhân viên dạng 2	317
Bài 16.	Chương trình cơ sở dữ liệu – Thiết kế form cập nhật nhân viên	345

Những bài thực hành VISUAL BASIC .NET CĂN BẢN

Chịu trách nhiệm xuất bản
CÁT VĂN THÀNH

Biên tập : HẠNH NGUYỄN
Trình bày : THIÊN TRUNG
Vẽ bìa : THIÊN ÂN
Sửa bản in : SONG HẰNG

NHÀ XUẤT BẢN THỐNG KÊ
98 Thụy Khuê - Tây Hồ - Hà Nội
CN : 16 Nguyễn Huệ, Quận 1, TP. Hồ Chí Minh
ĐT : 8290047

Liên kết xuất bản :

CTY VĂN HÓA MINH TRÍ - NS. VĂN LANG
25 Nguyễn Thị Minh Khai, Q.1, TP.HCM
ĐT : 8.242157 - 8233022 - Fax : 84.8.235079

In 1000 cuốn khổ 14.5x20.5cm tại Xưởng in CN Trung Tâm Hội Chợ Triển Lãm Việt Nam.
Giấy chấp nhận đăng ký KHXB số 92-357/XB-QLXB Cục xuất bản cấp ngày 26.03.2004.
In xong và nộp lưu chiểu quý 2 năm 2004.

NHỮNG BÀI THỰC HÀNH

VISUAL BASIC .NET CĂN BẢN

- Nắm bắt cách thức sử dụng điều khiển Timer, Check-Box, Picture Box, Progress Bar, cũng như cách thiết kế các ứng dụng có nhiều form.
- Làm quen với kỹ thuật lập trình hướng đối tượng trong Visual Basic .NET – một trong những thế mạnh so với Visual Basic 6.0.
- Xây dựng một trình soạn thảo văn bản, mô phỏng trình Notepad trong Windows.
- Thiết kế form cập nhật, điều chỉnh nội dung một table trong CSDL Access với Visual Basic .NET.
- Làm quen với các hàm xử lý dữ liệu được xây dựng sẵn trong Visual Basic .NET.
- Sử dụng các loại Listbox, biết cách nạp và lấy giá trị từ Listbox, cách tạo Listbox cho việc chọn nhiều mục cùng lúc trong Visual Basic .NET.
- Tự tạo hộp thông báo trong Visual Basic .NET và thiết kế các form dạng MDI.



Nhà Sách
VNN LING

25 Nguyễn Thị Minh Khai, Q.1, TP. HCM

ĐT: 8242157 - 8233022 - FAX: 8235079

9 Phan Đăng Lưu, Q.BT, TP. HCM - ĐT: 8413306

E-mail: minhtri.com@hcm.vnn.vn

những bài thực hành visual



8 935073 007514

giá: 40.000đ