

TỰ HỌC LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG & LẬP TRÌNH CƠ SỞ DỮ LIỆU



MỘT CÁCH NHANH CHÓNG VÀ CÓ HIỆU QUẢ NHẤT

Qua các chương trình mẫu :

- LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- LẬP TRÌNH DÙNG CƠ SỞ DỮ LIỆU ACCESS 2003
- LẬP TRÌNH DÙNG CƠ SỞ DỮ LIỆU SQL SERVER 2000
- PHÂN PHỐI ỨNG DỤNG QUA SETUP WIZARD
- BÁO CÁO QUA CRYSTAL REPORT

• (Dùng cho kỹ thuật viên & sinh viên ngành tin học) •

CUNG CÁCH TRÌNH BÀY :

- Ngắn gọn và dễ hiểu
- Hướng dẫn chi tiết và rõ ràng
- Các chương trình mẫu thực tế và đa dạng



Đậu Quang Tuấn

Cử nhân toán – Kỹ sư tin học

Tự học lập trình hướng đối tượng và lập trình cơ sở dữ liệu

C#

MỘT CÁCH NHANH CHÓNG VÀ CÓ HIỆU QUẢ NHẤT

Qua các chương trình mẫu

- ☒ Lập trình hướng đối tượng
- ☒ Lập trình dùng cơ sở dữ liệu Access 2003
- ☒ Lập trình dùng cơ sở dữ liệu SQL Server 2000
- ☒ Phân phối ứng dụng qua Setup Wizard
- ☒ Báo cáo qua Crystal Report

(Dùng cho kỹ thuật viên và sinh viên ngành tin học)

Cung cách trình bày:

- Ngắn gọn và dễ hiểu
- Hướng dẫn chi tiết và rõ ràng
- Các chương trình mẫu thực tế và đa dạng

NHÀ XUẤT BẢN GIAO THÔNG VẬN TẢI 2006

Lời mở đầu

Xin chào các bạn học sinh, sinh viên và quý độc giả yêu thích bộ môn tin học. Lời đầu tiên, chúng tôi xin chân thành cảm ơn các bạn vì trong thời gian qua đã đón đọc những tập sách về ứng dụng và lập trình của chúng tôi biên soạn. Những ý kiến đóng góp và nhận xét của các bạn sinh viên và quý độc giả xa gần, đã khích lệ vô cùng to lớn và động viên chúng tôi tiếp tục nghiên cứu những khía cạnh mới trong bộ môn tin học này. Sau các cuốn “Tự học lập trình Visual Basic.NET”, “Tự học lập trình cơ sở dữ liệu Access 2003 và Visual Basic.NET”, “Tự học lập trình cơ sở dữ liệu SQL Server 2000 và Visual Basic.NET”, “Tự học lập trình Visual C++.NET” đã được xuất bản. Hôm nay chúng tôi hoàn thành tập sách “Tự học lập trình hướng đối tượng và lập trình cơ sở dữ liệu Visual C#” qua các chương trình mẫu sao cho thật ngắn gọn và dễ hiểu. Nội dung của tập sách sẽ hướng dẫn các bạn lập trình hướng đối tượng căn bản qua các bài tập mẫu, lập trình dùng CSDL Access 2003, lập trình dùng CSDL SQL Server 2000 theo hai cách, tự thiết kế hoặc dùng Data Form Wizard. Chúng tôi có hướng dẫn thêm cách in các bảng bằng Crystal Report và cách thực hiện để phân phối ứng dụng khi đã hoàn thành xong chương trình. Nghiên cứu tập sách này, chắc chắn các bạn sẽ cảm thấy hài lòng, những kiến thức hướng dẫn trong các chương trình mẫu sẽ giúp các bạn tự thiết kế những chương trình khác một cách dễ dàng và các bạn có thể nghiên cứu thêm để viết tiểu luận hoặc đồ án ra trường. Hy vọng tập sách này sẽ là động cơ thúc đẩy các bạn yêu thích lập trình và sẽ đưa các bạn tiến sâu hơn trên con đường tin học của mình.

Mặc dầu đã có nhiều cố gắng khi biên soạn, nhưng cũng không tránh khỏi thiếu sót, chúng tôi mong được sự đóng góp ý kiến của quý độc giả xa gần, để lần tái bản sau sẽ hoàn chỉnh hơn. Chúc các bạn học giỏi và đóng góp trí tuệ mình cho đất nước thân yêu của chúng ta.

Đây là một cuốn sách không thể thiếu được trong tủ sách tin học của bạn. Nếu cảm nhận dễ hiểu xin giới thiệu với quý độc giả khác để cùng đón đọc.

Tác giả
Đậu Quang Tuấn

Phần I

cách vào cửa sổ

MS Development Environment

Tìm hiểu cửa sổ thiết kế

MS Development Environment

Đầu tiên các bạn cài **Visual Studio .NET** vào máy của mình, việc cài đặt bình thường như những ngôn ngữ khác.

Trong phần lý thuyết ngắn gọn này, chúng tôi sẽ giải thích về cửa sổ, nơi bạn sẽ thiết kế một chương trình, đó là **Microsoft Development Environment**.

Ghi chú:

Khi chúng tôi nói click chuột, các bạn hiểu là click nút trái của chuột.

Microsoft Development Environment

Đây là một môi trường triển khai tổng thể, cho phép làm việc ngay trên mọi khía cạnh của một dự án. Bạn tiến hành các bước để thấy được giao diện của Microsoft Development Environment.

Bạn *khởi động Visual C# .NET* một trong những cách sau:

① Cách 1:

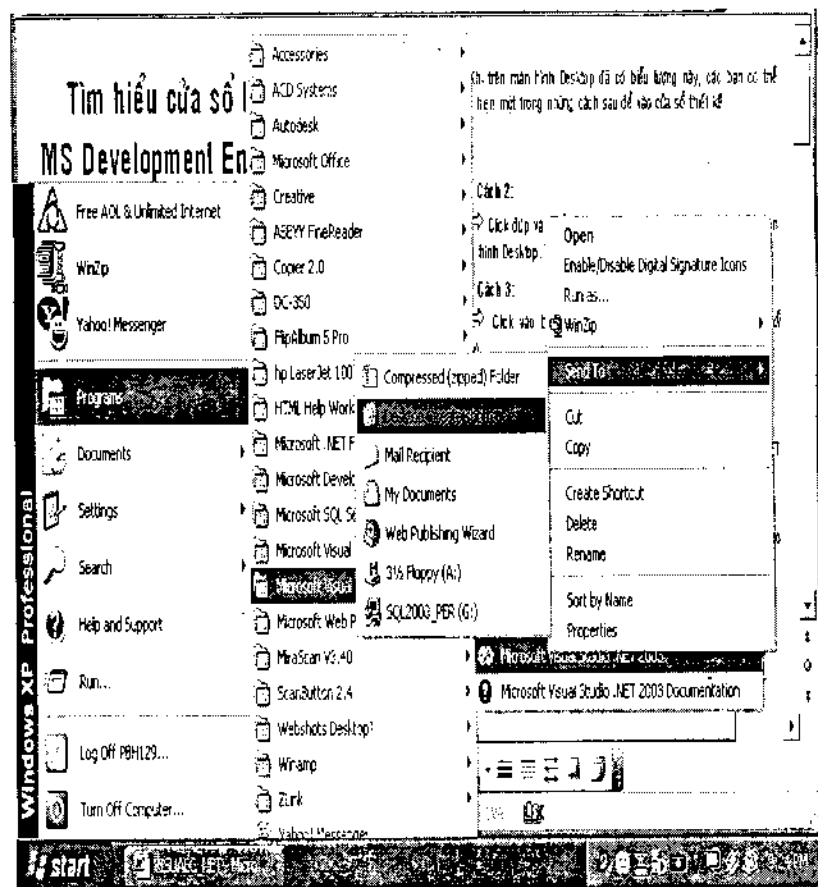
- ⇒ Vào **Start**.
- ⇒ Đưa vật sáng đến **Programs**.
- ⇒ Đưa vật sáng đến **Microsoft Visual Studio .NET 2003**.
- ⇒ Click chọn **Microsoft Visual Studio .NET 2003**.

Nếu biểu tượng Microsoft Visual Studio .NET 2003 chưa có trên màn hình Desktop, các bạn có thể tạo như sau:

- ⇒ Vào **Start**.

- ⇒ Đưa vệt sáng đến **Programs**.
- ⇒ Đưa vệt sáng đến **Microsoft Visual Studio .NET 2003**.
- ⇒ Click *nút phải chuột* tại **Microsoft Visual Studio .NET 2003**.
- ⇒ Di chuyển vệt sáng đến **Send To** trong menu hiện ra.
- ⇒ Click chọn **Desktop (create shortcut)**. Trên Desktop sẽ có biểu tượng của Microsoft Visual Studio .NET 2003.

Khi trên màn hình Desktop đã có biểu tượng này, các bạn có thể thực hiện một trong những cách sau để vào cửa sổ thiết kế.



H1 Tạo biểu tượng Microsoft Visual Studio .NET trên Desktop



H2 Biểu tượng Microsoft Visual Studio nằm trong hình tròn

② Cách 2:

⇒ Click đúp vào biểu tượng **Microsoft Visual Studio .NET 2003** trên màn hình **Desktop**.

③ Cách 3:

⇒ Click vào biểu tượng **Microsoft Visual Studio .NET 2003** (Đổi màu)

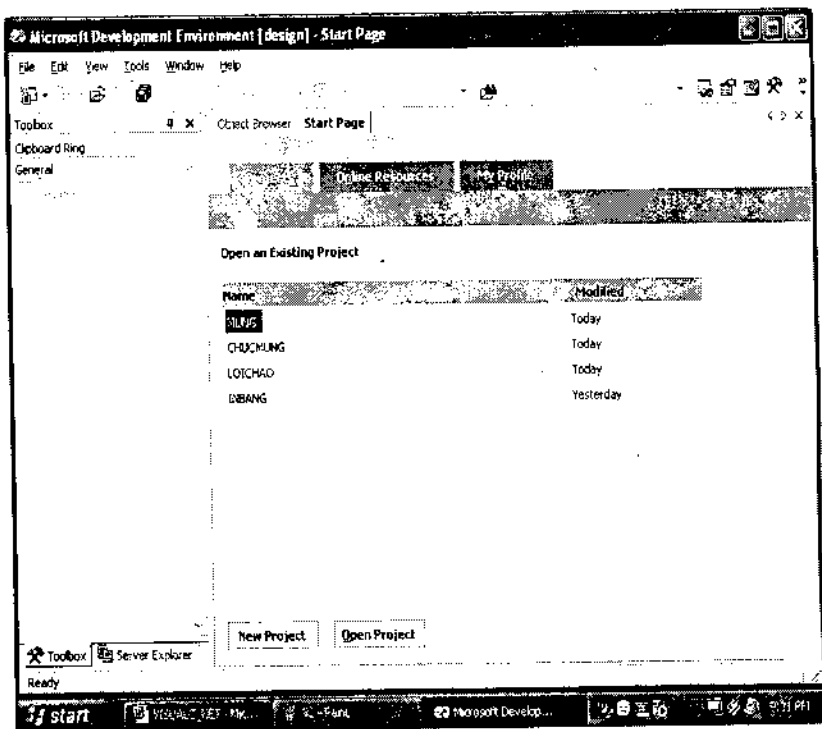
⇒ Nhấn phím **Enter**.

④ Cách 4:

⇒ Click phải chuột vào biểu tượng **Microsoft Visual Studio .NET 2003**

⇒ Click vào nút **Open** trong menu hiện ra

Khi thực hiện một trong những cách trên, các bạn sẽ thấy giao diện của Microsoft Development Environment như hình sau:



H3 Cửa sổ Microsoft Development Environment

Trên cửa sổ này các bạn có thể thiết kế các chương trình trên Visual Basic .NET, Visual C++ .NET, Visual C# .NET, ...

Ghi chú:

Trong Visual Studio .NET cửa sổ để vào các ngôn ngữ .NET chung, từ cửa sổ này, chúng ta có thể chọn ngôn ngữ mà mình muốn.

Phần 2

Dùng Visual C# Projects và Console Application

BÀI 1

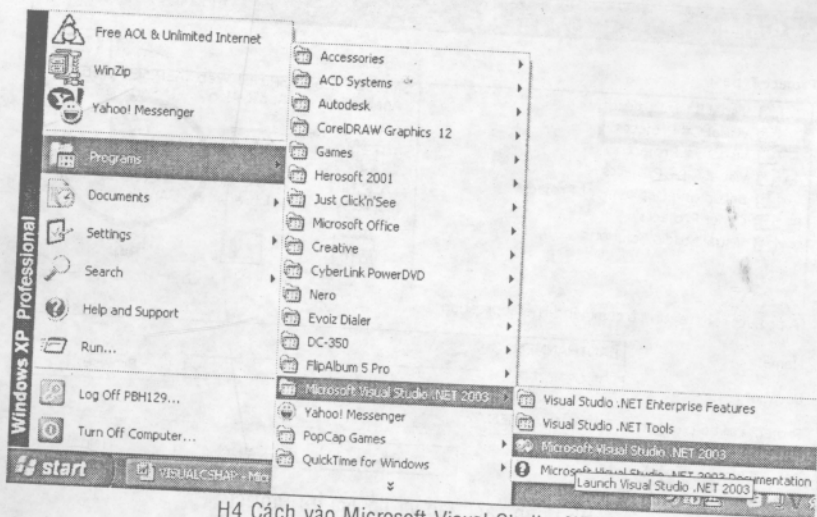
CHO XUẤT HIỆN MỘT CÂU CHÀO

Bài đầu tiên là một chương trình đơn giản, sẽ cho xuất hiện trên màn hình một câu "CHAO MUNG BAN DEN VOI VISUAL C# .NET"

Các chương trình dưới đây được tạo trong Visual C# dùng **Console Application**.

Bạn sẽ tiến hành như sau:

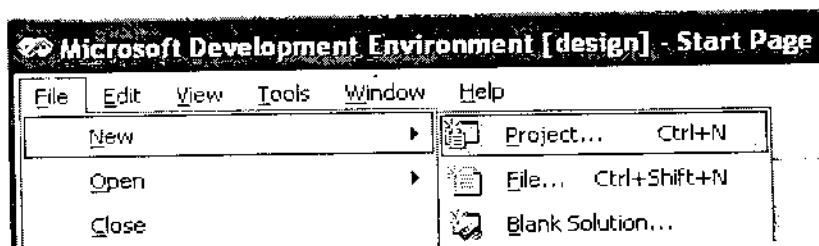
* Khởi động **Microsoft Visual Studio.NET**, bằng cách click nút **Start**, di chuyển vật sáng đến **Programs**, di chuyển vật sáng đến **Microsoft Visual Studio.NET 2003**, chọn **Microsoft Visual Studio.NET 2003** như hình sau:



H4 Cách vào Microsoft Visual Studio .NET

Khi chọn xong, bạn sẽ thấy cửa sổ **Microsoft Development Environment** như hình H3 ở trên

* Click chuột vào menu **File**, chọn **New**, chọn **Project** như hình sau:



H5 Tạo Project mới

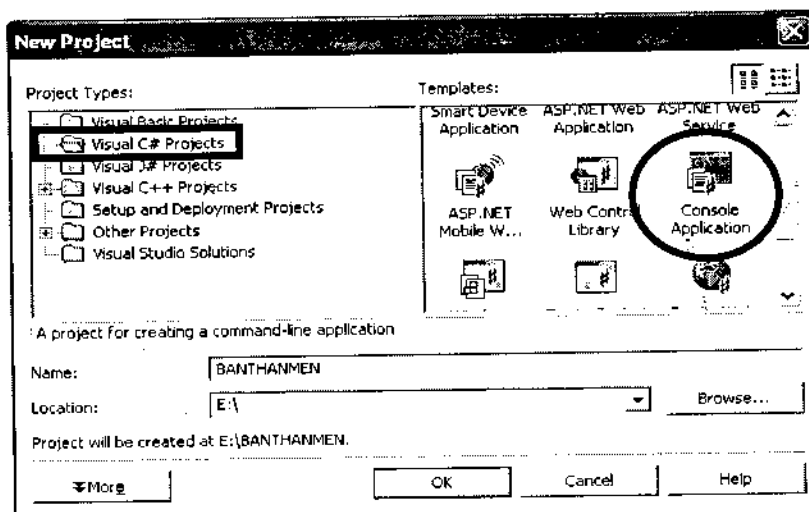
* Trong cửa sổ **New Project** hiện ra, bạn chọn:

* **Visual C# Project** trong khung bên trái có tên **Project Types**.

* Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.

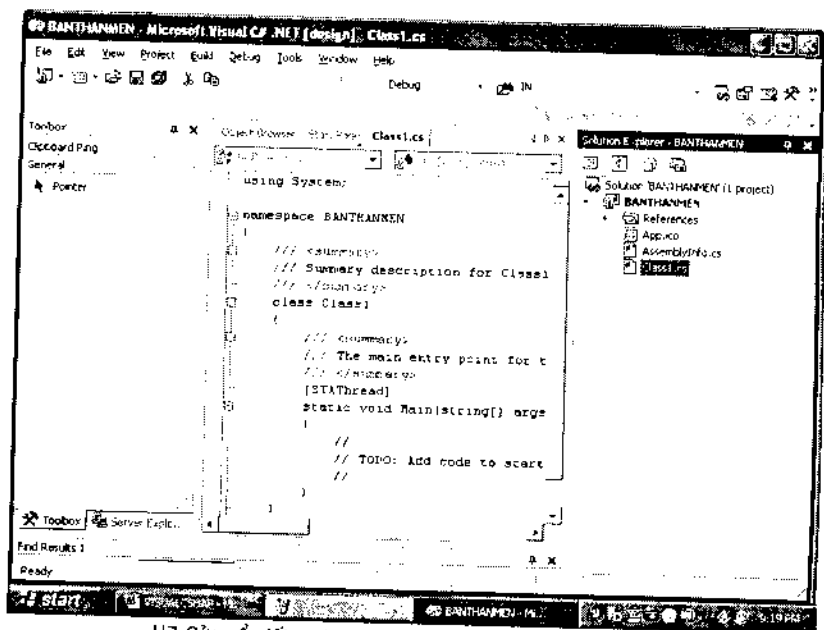
* Đặt tên trong khung **Name**, giả sử là tên **BANTHANMEN**

* Chọn đường dẫn chứa **Project**, giả sử chúng tôi chọn **E:** trong khung **Location**, bạn sẽ thấy **Project** chứa trong **E:\BANTHANMEN** như hình sau:



H6 Chọn cách tạo và nơi lưu trữ Project

* Click nút **OK**. Bạn sẽ có cửa sổ hiện ra như hình sau:



H7 Cửa sổ viết code hiện ra với những thông tin có sẵn

Bạn thấy phía bên trái là cửa sổ *viết code*, bên phải là cửa sổ **Solution Explorer** chứa thông tin của **Project**

Đoạn code có sẵn như sau:

```
using System;
namespace BANTHANMEN
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the
        /// application.
        /// </summary>
```

```

[STAThread]
static void Main(string[] args)
{
    //
    // TODO: Add code to start
    application here
    //
}

```

* Những dòng chữ ghi sau // là những dòng ghi chú, trình biên dịch sẽ bỏ qua.

* Nếu bắt đầu bằng /* và kết thúc là */. Kiểu này cho bạn kéo dài lời ghi chú lên nhiều hàng.

* Loại ghi chú thứ ba là bắt đầu bằng 3 dấu gạch chéo ///
Bạn sẽ viết đoạn code như sau vào dưới dòng

// TODO: Add code to start application here

```

Console.WriteLine("CHAO MUNG BAN DEN VOI VISUAL C# .NET");
Console.Read();

```

Nhập xong, bạn sẽ có nguyên đoạn code như sau:

```

using System;
namespace BANTHANMEN
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]

```

```

static void Main(string[] args)
{
    //
    // TODO: Add code to start application here
    // Hai lenh sau duoc them vao
    Console.WriteLine("CHAO MUNG BAN DEN VOI
        VISUAL C# .NET");
    Console.Read();
}
}
}

```

Lưu ý:

Trong Visual C# .NET tên của lớp hoặc tên của các phương thức cũng phân biệt chữ hoa và chữ thường như đặc thù của ngôn ngữ C hoặc C++. Nếu trong câu lệnh trên, bạn gõ write, read (w và r là kí tự thường thì trình biên dịch sẽ báo sai)

Giải thích:

☐ Việc nhập xuất dữ liệu thông qua thiết bị nhập xuất chuẩn (console) được thực hiện chủ yếu bởi các phương thức của lớp **Console**

☐ Phương thức **Write()** để xuất ra dữ liệu (có thể xuất ra chuỗi ký tự, số nguyên, số thực ...) ở trên là xuất ra chuỗi "CHAO MUNG BAN DEN VOI VISUAL C# .NET", chuỗi được bao bởi cặp nháy kép.

☐ Nếu không có **Console.Read()** hoặc **Console.ReadLine()** cuối chương trình, khi thực hiện chương trình, bạn thấy màn hình nháy lên và trở về cửa sổ viết code, không thấy được kết quả trên màn hình. Tương tự như trong C++ ta phải dùng hàm getch(), hoặc trong Pascal dùng Readln.

Để biên dịch và chạy chương trình, bạn thực hiện như sau:

- * Vào menu **Build**
- * Click **Build BANTHANMEN**
- * Nhấn phím **F5** để chạy thử chương trình

Nếu thực hiện đúng tất cả những điều trên, bạn sẽ có kết quả như sau:

■ E:\BANTHANMEN\bin\Debug\BANTHANMEN.exe

CHAO MUNG BAN DEN VOI VISUAL C# .NET

H8 Kết quả khi chạy chương trình

Bạn thử xóa bỏ câu lệnh `Console.Read();` hoặc đánh kí hiệu ghi chú phía trước dòng này, nhấn phím F5 để chạy chương trình, các bạn sẽ không thấy màn hình thể hiện kết quả, nó về cửa sổ viết code.

Cũng chương trình như trên, giả sử bạn muốn dòng một xuất hiện lời chào như trên, dòng hai xuất hiện câu "BAN SE THAY VISUAL C# KHONG KHO NHU MINH NGHI". Bạn thực hiện như thế nào? Chỉ đơn giản vào cửa sổ viết code, thêm Line vào sau Write của câu trên và viết thêm câu lệnh như sau:

```
Console.WriteLine("CHAO MUNG BAN DEN VOI VISUAL C# .NET");  
Console.Write("BAN SE THAY VISUAL C# KHONG KHO NHU MINH  
NGHI");  
Console.Read();
```

▣ Biên dịch và nhấn phím F5 để chạy thử chương trình, bạn sẽ có kết quả như hình sau:

■ E:\BANTHANMEN\bin\Debug\BANTHANMEN.exe

CHAO MUNG BAN DEN VOI VISUAL C# .NET
BAN SE THAY VISUAL C# KHONG KHO NHU MINH NGHI

H9 Kết quả nếu viết lại code như trên

▣ Nếu viết như sau (thêm Line vào Write của câu lệnh thứ hai) bạn cũng có kết quả như trên

```
Console.WriteLine("CHAO MUNG BAN DEN VOI VISUAL C# .NET");
```

```
Console.WriteLine("BAN SE THAY VISUAL C# KHONG KHO NHU MINH  
NGHI");
```

```
Console.Read();
```

☐ Nếu bỏ Line sau Write của câu lệnh thứ nhất, chúng ta sẽ có kết quả như hình sau:



H10 Nó sẽ thể hiện trên một dòng

Vì không có chữ "Line" sau câu lệnh thứ nhất, chuỗi của lệnh Write thứ hai sẽ in tiếp ngay sau chuỗi của lệnh WriteLine thứ nhất. Bạn thất câu thứ nhất và thứ hai liền nhau, khi thực hiện như vậy, bạn phải để thêm một khoảng trắng sau chuỗi thứ nhất, hoặc trước chuỗi thứ hai.

Vậy phương thức WriteLine() sẽ xuống dòng mới sau khi in ra màn hình, cũng giống như trong ngôn ngữ Pascal (Write hoặc WriteLn), trong ngôn ngữ C và C++ muốn xuống dòng thì dùng "\n"

Các bạn có thể viết chương trình để xuất ra trên màn hình một bài thơ mà bạn thích.

Giả sử chúng ta viết chương trình để xuất hiện trên màn hình bài thơ của Trịnh Công Sơn tặng cho Hoàng Anh như sau:

Đường xa vạn dặm em ngồi!

Nỗi đời xa vắng nỗi đời dần đo

Em là nhật nguyệt từ đây

Tuổi mười chín ấy cũng phai phai người

Em ơi hồng sẽ phai hồng

Đóa hoa hàm tiếu phiêu bồng nỗi đau

Hoa vàng một đóa lạ lùng

Gió chiều tịch mịch sẽ trùng trùng xa

Em ơi tịch mịch bây giờ

Ấy là nhan sắc đâu ngờ mất nhau

Bạn mở chương trình trên và sửa lại code như sau:


```

Console.WriteLine("Duong xa van dam em ngoi!");
Console.WriteLine("Noi doi xa vang noi thoi dan do");
Console.WriteLine("Em la nhat nguyet tu day");
Console.WriteLine("Tuoi muoi chin ay cung phai phai nguoi");
Console.WriteLine("Em oi hong se phai hong");
Console.WriteLine("Doa hoa ham tieu phieu bong noi dau");
Console.WriteLine("Hoa vang mot doa la lung");
Console.WriteLine("Gio chieu tích mich se trung trung xa");
Console.WriteLine("Em oi tích mich bay gio");
Console.WriteLine("Ay la nhan sac dau ngo mat nhau");
Console.Read();

```

▣ Biên dịch lại chương trình, nhấn phím F5 chúng ta sẽ có kết quả như hình sau:

```

E:\BANTHANMEN\bin\Debug\BANTHANMEN.exe
Duong xa van dam em ngoi!
Noi doi xa vang noi thoi dan do
Em la nhat nguyet tu day
Tuoi muoi chin ay cung phai phai nguoi
Em oi hong se phai hong
Doa hoa ham tieu phieu bong noi dau
Hoa vang mot doa la lung
Gio chieu tích mich se trung trung xa
Em oi tích mich bay gio
Ay la nhan sac dau ngo mat nhau

```

H11 Bài thơ vừa nhập

BÀI 2

CẤU TRÚC IF

Trước khi làm ví dụ, chúng tôi giới thiệu đến các bạn Kiểu dữ liệu cơ sở và các phép toán trong C#.

Các kiểu số nguyên với những phép toán: Cộng (dấu +), trừ (dấu -), nhân (dấu *), chia (dấu /) tính thương số và tính số dư (dấu %).

Các kiểu dữ liệu trong C#

Kiểu	Ý nghĩa
Byte	Kiểu số nguyên không dấu 1 byte (8 bit) Phạm vi từ 0 đến 255
Sbyte	Kiểu số nguyên có dấu 1 byte (8 bit) Phạm vi từ -128 đến 127
Short	Số nguyên có dấu, độ dài 16 bit Phạm vi từ -32768 đến 32767
Ushort	Số nguyên không dấu, độ dài 16 bit Phạm vi từ 0 đến 65535
Int	Số nguyên có dấu, độ dài 32 bit Phạm vi từ -2^{31} đến $2^{31}-1$
UInt	Số nguyên không dấu 32 bit Phạm vi từ 0 đến $2^{32}-1$
Long	Số nguyên có dấu 64 bit Phạm vi từ -2^{63} đến $2^{63}-1$
Ulong	Số nguyên không dấu 64 bit Phạm vi từ 0 đến $2^{64}-1$
Char	Lưu các kí tự unicode từ U+0000 đến U+FFFF
Float	Số thực 32 bit, chính xác đến 7 chữ số, số có trị tuyệt đối lớn nhất là $3,4 \times 10^{32}$. Số dương nhỏ nhất là $1,5 \cdot 10^{45}$.
Double	Số thực 64 bit, chính xác đến 15-16 chữ số, số có trị tuyệt đối lớn nhất là $1,7 \times 10^{308}$. Số dương nhỏ nhất là

	$4,9.10^{324}$.
Bool	Nhận giá trị true (đúng) hoặc false (sai).
String	Kiểu chuỗi ký tự, có thể dùng phép cộng chuỗi (+) để ghép hai chuỗi thành một. Các phép toán so sánh bằng (==) và khác (!=) để so sánh giá trị của các biến kiểu chuỗi.

Giống như C++, C# cũng có các phép toán như sau:

Các phép toán về số học

- * là phép toán nhân
- / là phép toán chia
- + là phép toán cộng
- là phép toán trừ
- % là phép toán tìm số dư

Toán tử gán

- Trong C# toán tử gán là "="

Ví dụ: $x=2;$
 $y=5;$

Các phép toán về luận lý

- && phép toán And
 Ví dụ: $(x>10)\&\&(y<1)$
- || phép toán Or
 Ví dụ: $(x\geq 5)\|\!(y\leq 4)$
- ! Phép toán phủ định

Các phép toán so sánh

- Trong C# các toán tử so sánh như sau:

- < nhỏ hơn
- > lớn hơn
- == bằng
- <= nhỏ hơn hoặc bằng
- >= lớn hơn hoặc bằng
- != khác

Ví dụ: $x<y$, $m>n$, $a==b$, $j<=l$, $z>0$, $u!=v$

Tác tử tăng giảm

Biến ++

Tăng Biến lên một đơn vị

Biến --

Giảm Biến một đơn vị

Ví dụ:

Nếu $x = 5$. $x++$ sẽ làm cho $x = 6$.

Nếu $y = 8$. $y--$ sẽ làm cho $y = 7$

Tác tử tăng giảm

Nếu tăng a lên 2000, bạn có thể làm như sau:

$a = a + 2000$;

Phép tính cộng xảy ra trước khi gán. Bạn có thể thực hiện

$a = a - 2000$;

$a = a * 2000$;

$a = a / 2000$;

$a = a \% 2000$;

Bạn có thể viết lại các dòng lệnh trên ngắn gọn hơn như sau:

$a += 2000$; // a tăng thêm 2000

$a -= 2000$; // trừ a bớt 2000

$a *= 2000$; // nhân a cho 2000

$a /= 2000$; // chia a cho 2000

$a \% = 2000$; // a mod cho 2000

Tác tử tiền tố và hậu tố

Bạn xem ví dụ sau để hiểu

Giả sử $B = 5$

Lệnh 1: $A = B++$; // dấu ++ nằm liền sau biến gọi là hậu tố

Lệnh 2: $A = ++B$; // dấu ++ nằm liền trước biến gọi là tiền tố

Lệnh 1 **gán trước rồi tăng sau**, nghĩa là $A = 10$, $B = 11$

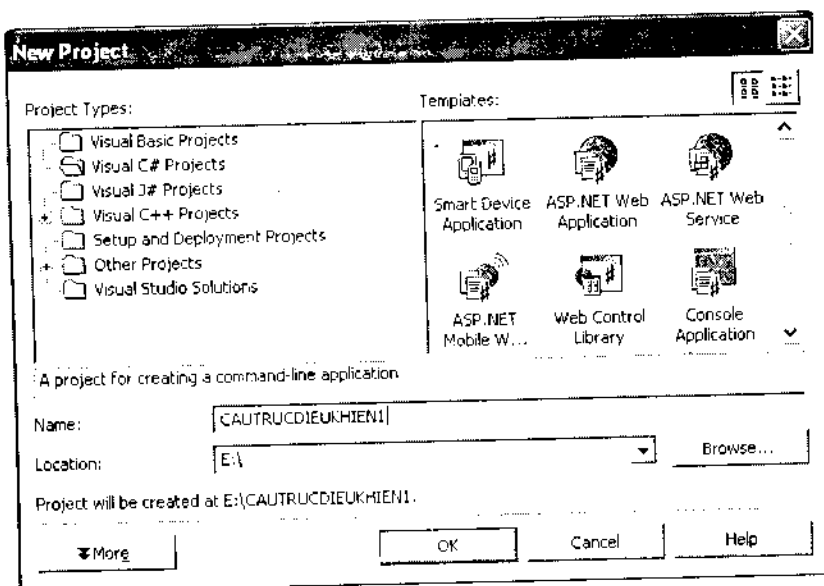
Lệnh 2 **tăng trước gán sau**, nghĩa là $A = 11$, $B = 11$

Chúng ta trở về với phát biểu if. Giả sử chúng ta viết chương trình nhập vào ba số nguyên a , b và c , nhập kết quả của tổng ba số đó. Nếu nhập đúng, trên màn hình sẽ báo ra câu "Hoan hô bé đã tính đúng.". Nếu nhập sai, chương trình sẽ báo ra câu "Rất tiếc bé đã tính sai!!!".

Chúng ta sẽ thực hiện như sau:

◆ Vào Microsoft Visual Studio .NET

◆ Tạo một **Project** mới có tên **CAUTRUCDIEUKHIEN1** bằng cách vào menu **File**, chọn **New**, chọn **Project**. Bạn thấy như hình sau:



H12

◆ Click nút **OK** trên hộp thoại.

Vào cửa sổ viết **code**, viết các câu lệnh ngay phía dưới câu ghi chú
// TODO: Add code to start application here như sau:

```
int a, b, c;
```

```
long kq;
```

```
Console.WriteLine("Ban nhap vao so a = ");
```

```
a=int.Parse(Console.ReadLine());
```

```
Console.WriteLine("Ban nhap vao so b = ");
```

```
b=int.Parse(Console.ReadLine());
```

```
Console.WriteLine("Ban nhap vao so c = ");
```

```
c=int.Parse(Console.ReadLine());
```

```
Console.WriteLine("Tong cua ba so tren = ");
```

```
kq=long.Parse(Console.ReadLine());
```

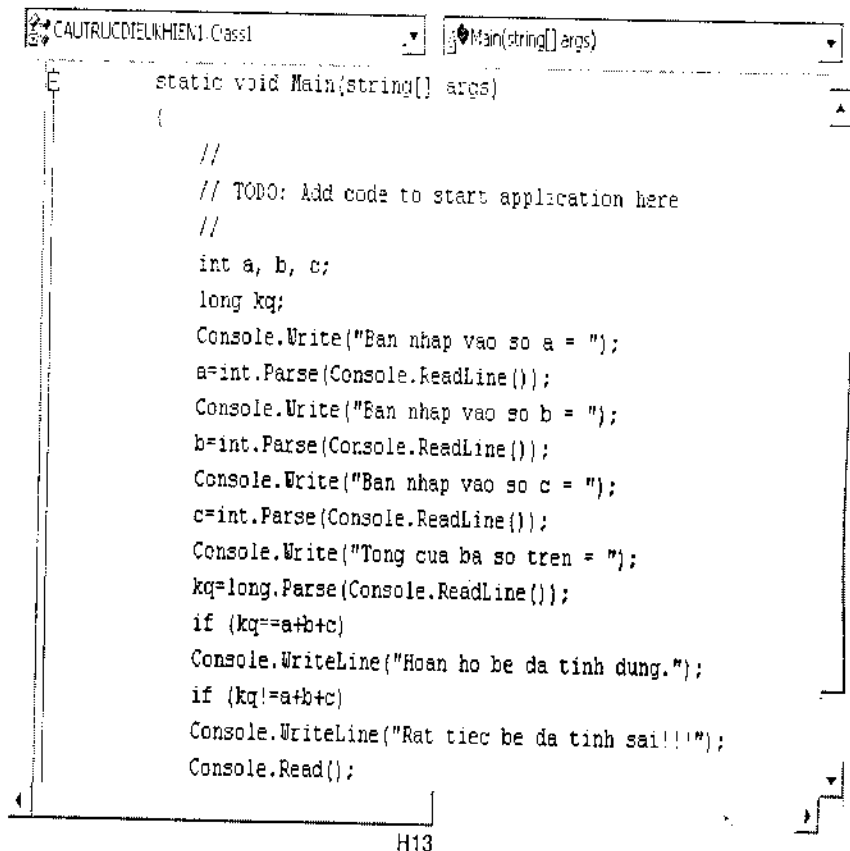
```
if (kq==a+b+c)
```

```
    Console.WriteLine("Hoan ho be da tinh dung.");
```

```
if (kq!=a+b+c)
```

```
Console.WriteLine("Rat tiếc be đã tính sai!!!");  
Console.Read();
```

Bạn có thể thấy một phần của cửa sổ viết code như hình dưới, còn các câu lệnh phía trên và các dấu móc bên dưới của chương trình, chúng có sẵn, đoạn code trên thêm vào vị trí như đã nêu ở trên.



```
CAUTRUCDIEUKHIEN1.Class1 Main(string[] args)  
{  
    //  
    // TODO: Add code to start application here  
    //  
    int a, b, c;  
    long kq;  
    Console.Write("Ban nhap vao so a = ");  
    a=int.Parse(Console.ReadLine());  
    Console.Write("Ban nhap vao so b = ");  
    b=int.Parse(Console.ReadLine());  
    Console.Write("Ban nhap vao so c = ");  
    c=int.Parse(Console.ReadLine());  
    Console.Write("Tong cua ba so tren = ");  
    kq=long.Parse(Console.ReadLine());  
    if (kq==a+b+c)  
        Console.WriteLine("Hoan ho be đã tính đúng.");  
    if (kq!=a+b+c)  
        Console.WriteLine("Rat tiếc be đã tính sai!!!");  
    Console.Read();  
}
```

Giải thích:

- Khai báo biến a, b và c có kiểu nguyên (int).
- Biến kq có kiểu long.
- Hiện ra câu "Ban nhap vao so a = "

Bạn đã biết là việc nhập xuất dữ liệu thông qua thiết bị nhập xuất chuẩn (console) được thực hiện chủ yếu bởi các phương thức của lớp **Console: Read()/ReadLine()** và **Console: Write()/WriteLine()**. Phương

thức **ReadLine()** được sử dụng để nhập vào một chuỗi ký tự, phương thức **Read()** chỉ cho phép nhập vào một ký tự.

Để đổi thành các dạng dữ liệu khác ta có thể dùng phương thức **Parse()** của kiểu dữ liệu mong muốn.

- Với câu lệnh **a=int.Parse(Console.ReadLine());** sẽ đổi chuỗi (kí tự số nhập vào) thành số nguyên kiểu **int**.
- Tương tự như trên với các câu lệnh nhập số **b**, số **c** và kết quả tổng của ba số bên dưới.
- Phát biểu **if** thứ nhất với điều kiện **(kq==a+b+c)**. Nếu đúng thì sẽ thực hiện câu lệnh theo sau **if**, tiếp tục xét câu lệnh tiếp theo của chương trình là phát biểu **if** kế tiếp, với điều kiện **(kq!=a+b+c)**. Nếu thỏa sẽ thực hiện câu lệnh theo sau phát biểu **if** này.

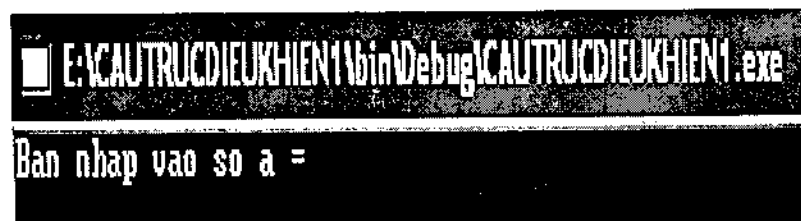
Biên dịch và chạy thử chương trình:

Biên dịch bằng cách:

◆ Vào menu **Build**, chọn **Build CAUTRUCDIEUKHIEN1**

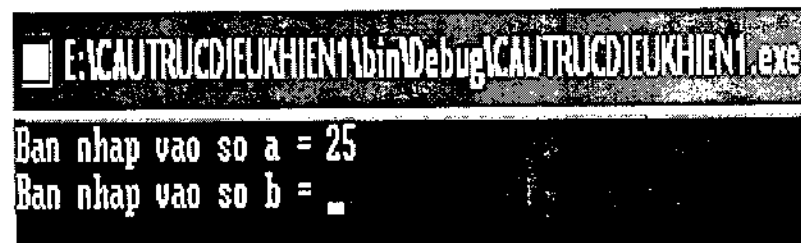
◆ Nhấn phím **F5** để chạy thử chương trình, bạn sẽ thấy như hình

sau:



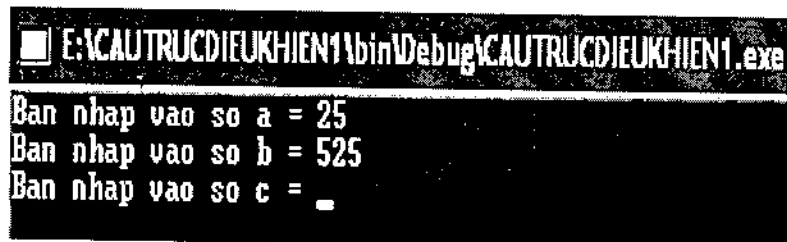
H14

◆ Bạn nhập số **a**, giả sử chúng ta nhập **25** sau đó gõ phím **Enter**, bạn sẽ thấy hình như sau:



H15

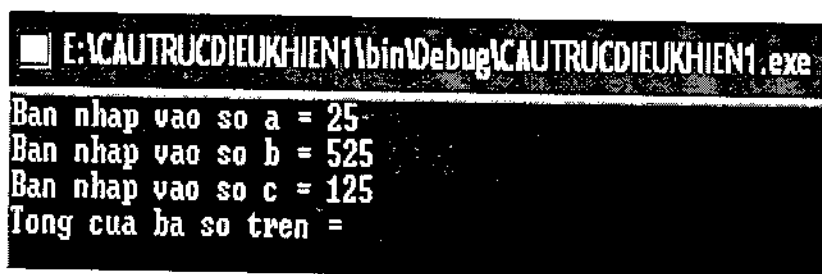
◆ Giả sử chúng ta nhập số b bằng 525, gõ phím **Enter**, hình sau hiện ra:



```
E:\CAUTRUCDIEUKHIEN1\bin\Debug\CAUTRUCDIEUKHIEN1.exe
Ban nhap vao so a = 25
Ban nhap vao so b = 525
Ban nhap vao so c = _
```

H16

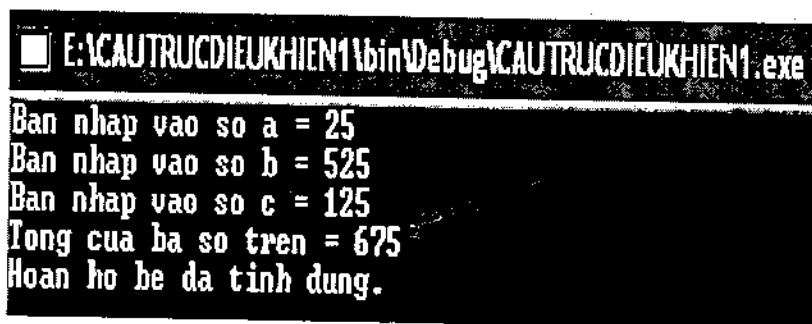
◆ Giả sử chúng ta nhập số c bằng 125, gõ phím **Enter**, chúng ta có hình như sau:



```
E:\CAUTRUCDIEUKHIEN1\bin\Debug\CAUTRUCDIEUKHIEN1.exe
Ban nhap vao so a = 25
Ban nhap vao so b = 525
Ban nhap vao so c = 125
Tong cua ba so tren =
```

H17

◆ Bạn nhập tổng của ba số trên, giả sử chúng ta nhập 675, gõ phím **Enter**, chúng ta thấy kết quả như hình sau:

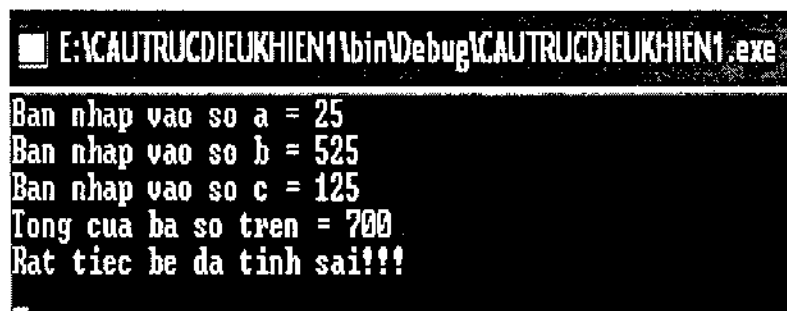


```
E:\CAUTRUCDIEUKHIEN1\bin\Debug\CAUTRUCDIEUKHIEN1.exe
Ban nhap vao so a = 25
Ban nhap vao so b = 525
Ban nhap vao so c = 125
Tong cua ba so tren = 675
Hoan ho be da tinh dung.
```

H18

◆ Vì $a + b + c = 675$, nên kết quả báo ra “Hoan ho be đa tinh dung”

◆ Nhấn phím **F5** chạy lại chương trình, nhập lại a, b, c như trên, nhưng nhập kết quả tổng của ba số trên là một số khác 675, giả sử chúng ta nhập kết quả là 700. Bạn sẽ thấy kết quả như hình sau:



```
E:\CAUTRUCDIEUKHIEN1\bin\Debug\CAUTRUCDIEUKHIEN1.exe
Ban nhap vao so a = 25
Ban nhap vao so b = 525
Ban nhap vao so c = 125
Tong cua ba so tren = 700
Rat tiếc be đa tinh sai!!!
```

H19

Bài trên bạn đã dùng phát biểu if với cấu trúc như sau:
if (điều kiện)

S;

Phát biểu trên sẽ xét điều kiện theo sau if. Nếu đúng thì thực hiện lệnh S. Nếu sai thì coi như phát biểu này đã thực hiện xong.

Ghi chú:

■ Nếu sau phát biểu if có một câu lệnh, bạn không dùng cặp dấu móc. Nếu có từ hai câu lệnh trở lên, bạn phải dùng cặp dấu móc bao quanh các câu lệnh này.

Ví dụ:

Cũng bài tập trên, nếu bạn sửa lại code ở phần cuối như sau:

if (kq==a+b+c)

Console.WriteLine("Hoan ho be đa tinh dung.");

if (kq!=a+b+c)

Console.WriteLine("Rat tiếc be đa tinh sai!!!");

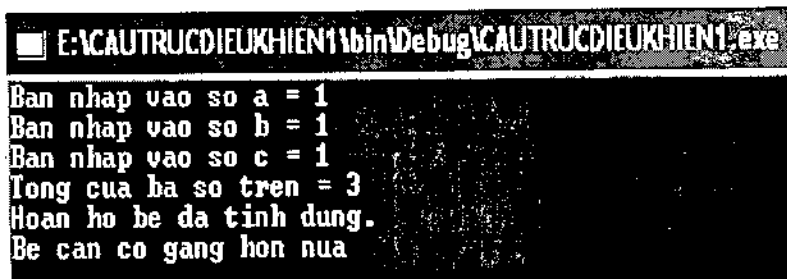
Console.WriteLine("Be can co gang nhiều hơn nữa");

Console.Read();

Bạn thêm câu lệnh **tô đậm** ở trên.

Nhấn phím **F5** chạy thử chương trình, giả sử nhập $a=1$, $b=1$, $c=1$. Nhập kết quả là 3. Đúng. Thay vì trên màn hình chỉ xuất hiện câu

"Hoan ho be da tinh dung", nếu viết như trên nó sẽ hiện thêm câu "Be can co gang nhieu hon nua". Bạn sẽ thấy như hình sau:



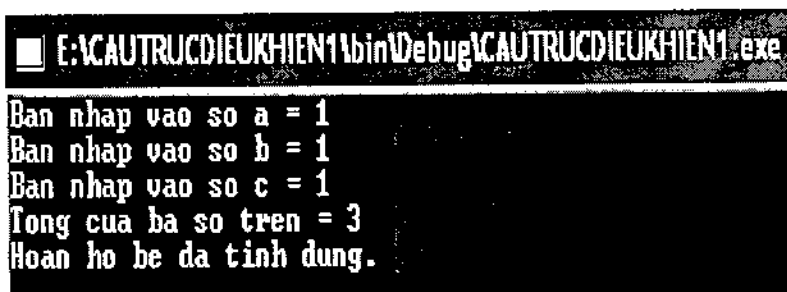
H20

Kết quả như vậy là sai, thực hiện lệch ý của chúng ta. Để không hiện câu "Be can co gang nhieu hon nua", Bạn phải thêm cặp dấu móc { } bao quanh hai câu lệnh của phát biểu if thứ hai như sau:

```
if (kq==a+b+c)
    Console.WriteLine("Hoan ho be da tinh dung.");
if (kq!=a+b+c)
{
    Console.WriteLine("Rat tiếc be da tinh sai!!!");
    Console.WriteLine("Be can co gang nhieu hon nua");
}
```

```
Console.Read();
```

Chạy thử chương trình, nhập kết quả như trên, bạn sẽ thấy như sau:



H21

Bạn cần chú ý thực hiện đúng như trên để tránh sai sót khi lập trình.

BÀI 3

CẤU TRÚC IF CÓ ELSE

Cấu trúc if else sẽ có tác dụng chuyển hướng tùy theo biểu thức điều kiện nhận giá trị đúng (True) hay sai (False).

Ví dụ:

```
if (kq==a+b+c)
    Console.WriteLine("Hoan ho be da tinh dung.");
if (kq!=a+b+c)
{
    Console.WriteLine("Rat tiec be da tinh sai!!!");
    Console.WriteLine("Be can co gang nhieu hon nua");
}
```

Chúng ta có thể sửa lại như sau:

```
if (kq==a+b+c)
    Console.WriteLine("Hoan ho be da tinh dung.");
else
{
    Console.WriteLine("Rat tiec be da tinh sai!!!");
    Console.WriteLine("Be can co gang nhieu hon nua");
}
```

Bạn thử vào cửa sổ viết code, sửa lại phát biểu có else như trên, lúc này đoạn code sẽ như sau:

```
int a, b, c;
long kq;
Console.Write("Ban nhap vao so a = ");
a=int.Parse(Console.ReadLine());
Console.Write("Ban nhap vao so b = ");
b=int.Parse(Console.ReadLine());
Console.Write("Ban nhap vao so c = ");
c=int.Parse(Console.ReadLine());
Console.Write("Tong cua ba so tren = ");
```

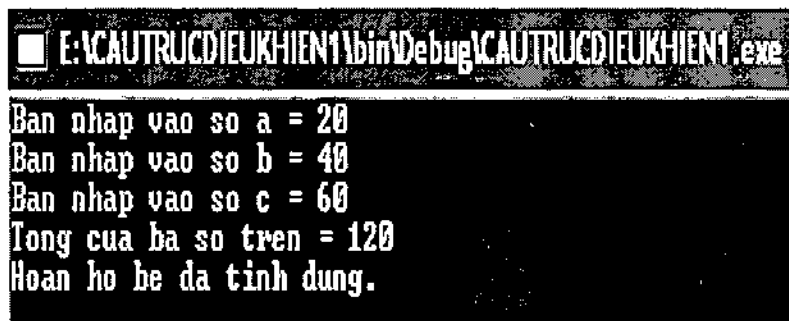
```

kq=long.Parse(Console.ReadLine());
if (kq==a+b+c)
    Console.WriteLine("Hoan ho be da tinh dung.");
else
{
    Console.WriteLine("Rat tiếc be da tinh sai!!!");
    Console.WriteLine("Be can cố gang hơn nữa");
}
Console.Read();

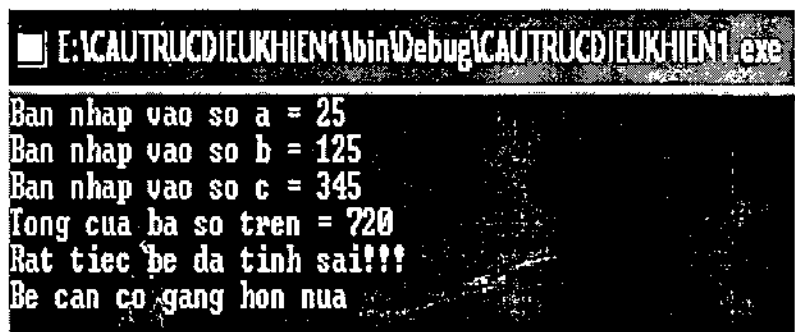
```

☐ Biên dịch và nhấn phím F5 để chạy thử chương trình, bạn sẽ có kết quả giống như dùng hai phát biểu if ở trên.

Bạn xem thử kết quả qua hai hình sau với các số mà chúng tôi nhập.



H22 Khi nhập a=20, b=40, c=60 và kết quả là 120



H23 Khi nhập kết quả tính tổng sai

Phát biểu if có else. Nếu điều kiện sau if đúng thì sẽ thực hiện câu lệnh sau if. Ngược lại sẽ thực hiện câu lệnh sau else.

if (điều kiện)

s1;

else

s2;

Ghi chú:

Bạn ngầm hiểu s1, s2 có thể là một hoặc nhiều phát biểu, nếu từ hai trở lên thì ta phải dùng cặp dấu móc.

BÀI 4

CẤU TRÚC IF ELSE LỒNG NHAU

Viết chương trình nhập vào điểm thi đại học của ba môn Toán, Lý và Hóa. Nếu tổng số điểm của ba môn lớn hơn hoặc 25 thì sẽ bảo là đậu vào đại học. Ngược lại, xét tiếp điều kiện, nếu tổng số điểm của 3 môn chỉ cần lớn hơn hoặc bằng 21 và mỗi môn phải lớn hơn hoặc bằng 7 thì cũng đậu. Các trường hợp còn lại sẽ không đủ tiêu chuẩn.

- * Vào **Microsoft Visual studio .NET**

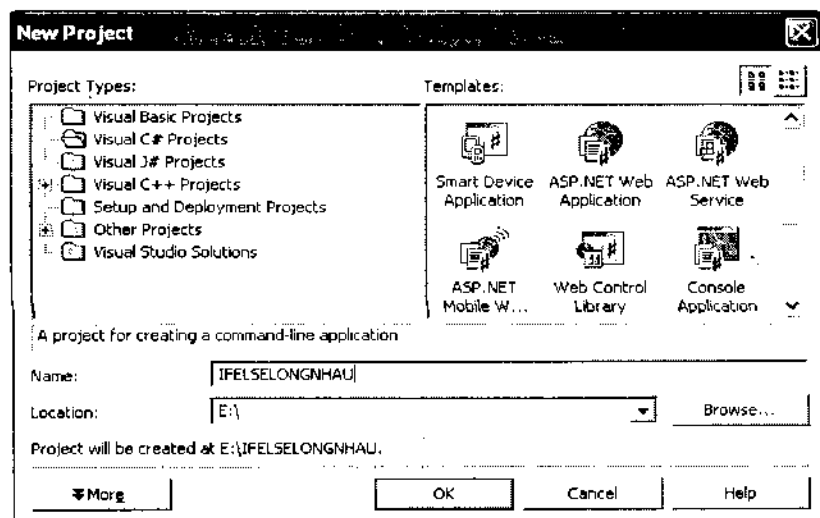
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**

- * Trong cửa sổ **New Project** hiện ra, bạn chọn:

- * **Visual C# Project** trong khung bên trái có tên **Project Types**.

- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.

- * Đặt tên trong khung **Name**, giả sử là tên **IFELSELONGNHAU** như hình sau:



* Click nút OK trên hộp thoại.

Viết code như sau vào sau dòng ghi chú (như các bài tập trước)

```
// TODO: Add code to start application here
float toan, ly, hoa, ketqua;
Console.WriteLine("Nhap diem Toan = ");
toan=float.Parse(Console.ReadLine());
Console.WriteLine("Nhap diem Ly = ");
ly=float.Parse(Console.ReadLine());
Console.WriteLine("Nhap diem hoa = ");
hoa=float.Parse(Console.ReadLine());
ketqua=toan+ly+hoa;
if (ketqua>=25)
{
    Console.WriteLine("Thi sinh nay duoc vao dai hoc.");
    Console.WriteLine("Khong can xet diem khong che");
}
else
{
    if ((ketqua>=21)&&(toan>=7)&&(ly>=7)&&(hoa>=7))
        Console.WriteLine("Duoc vao dai hoc vi cac mon tu 7 tro len");
    else
        Console.WriteLine("Khong du tieu chuan vao dai hoc");
}
Console.ReadLine();
```

Giải thích:

- Khai báo các biến toan, ly, hoa và ketqua có kiểu số thực (float)
- Nhập điểm Toán, Lý và Hóa, các phương thức nhập xuất và chuyển đổi kiểu xem giải thích ở các bài trước.
- Gán tổng số điểm của ba môn cho biến có tên ketqua.
- Nếu ketqua>=25. Báo hai phát biểu theo sau if.
- Ngược lại (tức ketqua<25), xét trường hợp nếu đồng thời ketqua>=21 và điểm của từng môn đều lớn hơn hoặc bằng 7 thì báo câu lệnh theo sau phát biểu if tương ứng. Ngược lại báo câu lệnh theo sau else cuối cùng.

Code đầy đủ của bài này như sau: (phần code còn lại là do phần mềm tạo sẵn)

```
using System;
namespace IFELSELONGNHAU
{
    /// <summary>
    /// Summary description for Class1.
    /// </summary>
    class Class1
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main(string[] args)
        {
            //
            // TODO: Add code to start application here
            float toan, ly, hoa, ketqua;
            Console.Write("Nhap diem Toan = ");
            toan=float.Parse(Console.ReadLine());
            Console.Write("Nhap diem Ly = ");
            ly=float.Parse(Console.ReadLine());
            Console.Write("Nhap diem hoa = ");
            hoa=float.Parse(Console.ReadLine());
            ketqua=toan+ly+hoa;
            if (ketqua>=25)
            {
                Console.WriteLine("Thi sinh nay duoc vao dai hoc.");
                Console.WriteLine("Khong can xet diem khong che");
            }
            else
            {
                if ((ketqua>=21)&&(toan>=7)&&(ly>=7)&&(hoa>=7))
```



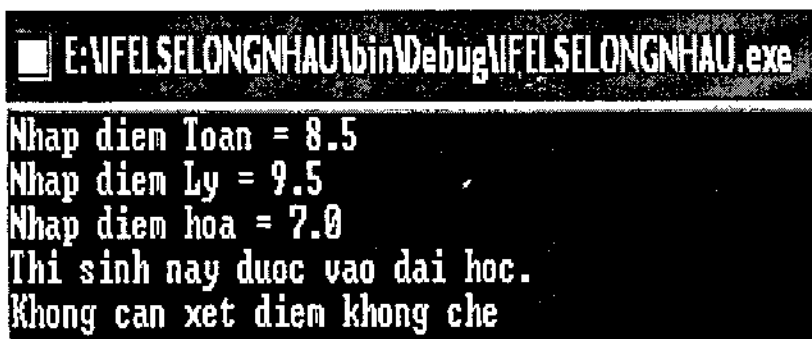
```

        Console.WriteLine("Duoc vao dai hoc vi cac mon tu 7 tro
                           len");

    else
        Console.WriteLine("Khong du tieu chuan vao dai hoc");
    }
    Console.ReadLine();
//
    }
}
}

```

* Biên dịch và nhấn phím F5 để chạy thử chương trình, nhập vào điểm của ba môn Toán, Lý và Hóa. Giả sử chúng ta nhập điểm của ba môn lần lượt như sau:



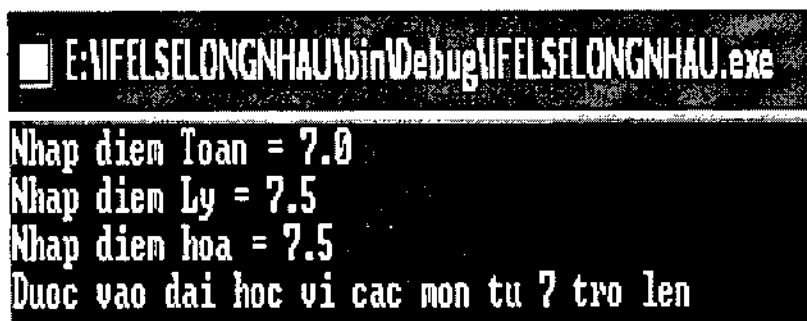
```

E:\IFELSELONGNHAU\bin\Debug\IFELSELONGNHAU.exe
Nhap diem Toan = 8.5
Nhap diem Ly = 9.5
Nhap diem hoa = 7.0
Thi sinh nay duoc vao dai hoc.
Khong can xet diem khong che

```

H25 Tổng điểm của ba môn là 25

Thỏa điều kiện của phát biểu if đầu tiên



```

E:\IFELSELONGNHAU\bin\Debug\IFELSELONGNHAU.exe
Nhap diem Toan = 7.0
Nhap diem Ly = 7.5
Nhap diem hoa = 7.5
Duoc vao dai hoc vi cac mon tu 7 tro len

```

H26 Tổng điểm của ba môn là 22, các môn đều ≥ 7

Không thỏa điều kiện của phát biểu if thứ nhất, thỏa điều kiện if thứ hai.

 E:\IFELSELONGNHAU\bin\Debug\IFELSELONGNHAU.exe

Nhap diem Toan = 8.5

Nhap diem Ly = 9.0

Nhap diem hoa = 6.5

Khong du tieu chuan vao dai hoc

H27 Tổng điểm của ba môn là 24

Không thỏa điều kiện của phát biểu if thứ nhất và thứ hai, mặc dầu tổng số điểm của ba môn ≥ 21 , nhưng có môn Hóa là 6.5, nên không đủ tiêu chuẩn vào đại học.

 E:\IFELSELONGNHAU\bin\Debug\IFELSELONGNHAU.exe

Nhap diem Toan = 10

Nhap diem Ly = 9

Nhap diem hoa = 6

Thi sinh nay duoc vao dai hoc.

Khong can xet diem khong che

H28 Tổng điểm của ba môn là 25

Mặc dầu một môn có điểm 6, nhưng vì tổng số điểm của ba môn bằng 25, thỏa điều kiện của đầu bài, nên thí sinh này đậu vào đại học.

BÀI 5

CẤU TRÚC CHỌN SWITCH CASE

Cấu trúc switch case có dạng như sau:

switch (biểu thức)

```
{  
    case giá trị 1 : s1; break;  
    case giá trị 2 : s2; break;  
    .....  
    .....  
    case giá trị n : sn; break;  
    [default : sm; break; ]  
}
```

▪ Đầu tiên các giá trị được xem xét và đối chiếu với trị của biểu thức, nếu trị đó nằm trong case nào, thì câu lệnh (*hoặc nhiều lệnh*) sau case đó sẽ được thực hiện, sau đó gặp lệnh break sẽ thoát khỏi phát biểu switch. Nó sẽ kiểm tra từ trên xuống, nếu không có giá trị của case nào thoả, nó sẽ thực hiện câu lệnh (*hoặc nhiều lệnh*) theo sau default (có thể có hoặc không), và kết thúc, nếu không có default, nó sẽ kết thúc và không làm gì cả (vì không có giá trị nào sau case thoả).

Ghi chú:

- Phải có lệnh break sau mục default.
- Biểu thức và các giá trị 1, giá trị 2, ..., giá trị n có thể là dữ liệu kiểu rời rạc, kiểu nguyên, kiểu kí tự, kiểu luận lý. Đặc biệt đối với C# thì những biểu thức này có thể là dữ liệu kiểu chuỗi (string).
- Chúng ta có thể dùng lệnh goto case để chuyển đến một nhánh khác trong cấu trúc này.

Nhận xét:

- Phát biểu **switch** là một dạng làm gọn của phát biểu if. Mọi phát biểu switch đều có thể viết lại thành phát biểu if. Bạn nên dùng

switch khi lựa chọn nhiều trường hợp, ví dụ như khi xếp loại học lực của học sinh... Bạn có thể xem ví dụ sau:

Viết chương trình nhập vào một số nguyên là số của tháng, chương trình sẽ báo ra bằng chữ tương ứng với số vừa nhập. Nếu nhập một số khác 1 đến 12, không thực hiện gì hết.

- * Vào **Microsoft Visual studio .NET**

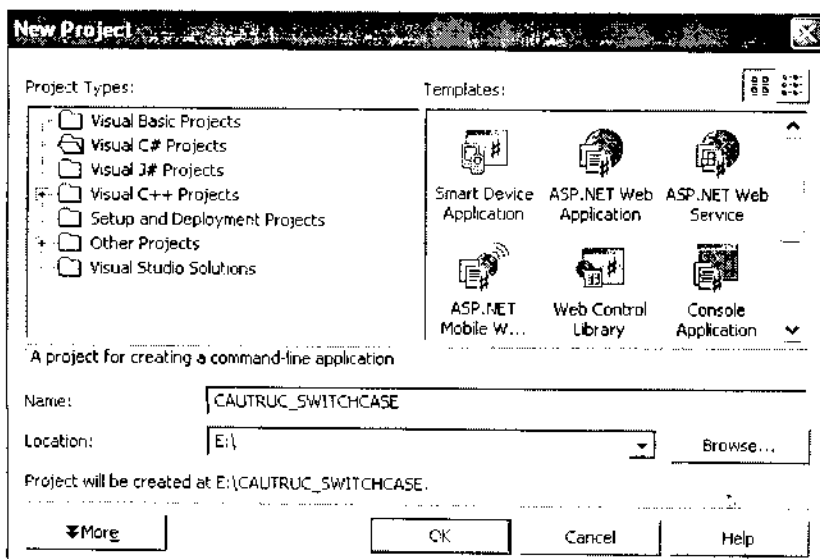
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**

- * Trong cửa sổ **New Project** hiện ra, bạn chọn:

- * **Visual C# Project** trong khung bên trái có tên **Project Types**.

- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.

- * Đặt tên **CAUTRUC_SWITCHCASE** trong khung **Name** như hình sau:



H29

- * Click nút **OK** trên hộp thoại.

Bạn viết đoạn code này vào sau câu ghi chú của chương trình.

```
// TODO: Add code to start application here  
byte so;
```

```

Console.Write("Nhap so cua thang : ");
so=byte.Parse(Console.ReadLine());
switch (so)
{
    case 1:
        Console.Write("Thang Gieng");
        break;
    case 2:
        Console.Write("Thang hai");
        break;
    case 3:
        Console.Write("Thang ba");
        break;
    case 4:
        Console.Write("Thang tu");
        break;
    case 5:
        Console.Write("Thang nam");
        break;
    case 6:
        Console.Write("Thang sau");
        break;
    case 7:
        Console.Write("Thang bay");
        break;
    case 8:
        Console.Write("Thang tam");
        break;
    case 9:
        Console.Write("Thang chin");
        break;
    case 10:
        Console.Write("Thang muoi");
        break;
    case 11:

```

```

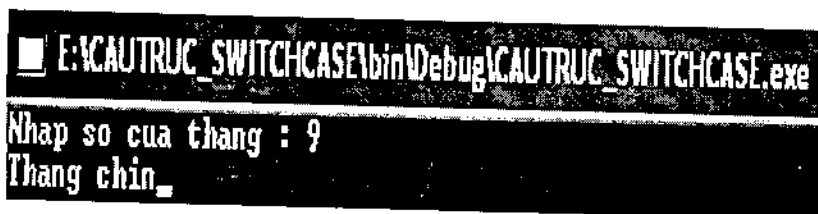
    Console.WriteLine("Thang muoi mot");
    break;
case 12:
    Console.WriteLine("Thang muoi hai");
    break;
}

```

Console.ReadLine();

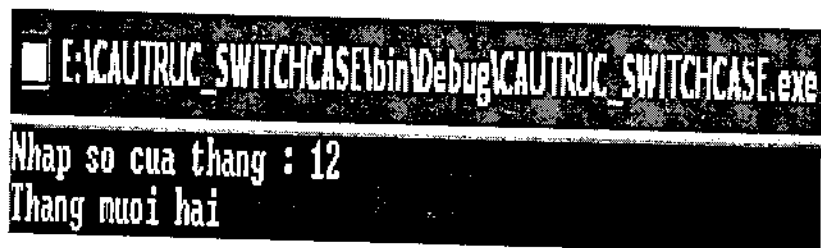
Giải thích:

- Khai báo biến so có kiểu nguyên (byte)
- Nhập vào một số, dùng phương thức Parse để đổi thành kiểu byte
- Biểu thức so được so sánh với các giá trị sau case, nếu giống, xuất ra màn hình câu lệnh sau case đó, lệnh break sẽ thoát khỏi cấu trúc switch case.
- ◆ Biên dịch và nhấn phím F5 chạy thử chương trình, giả sử nhập số 9. gõ phím Enter, bạn có kết quả như sau:



H30

- Chạy lại chương trình, giả sử nhập vào số 12, bạn sẽ thấy kết quả như hình sau:



H31

- Chạy lại chương trình, nhập vào số 15, bạn sẽ thấy kết quả như hình sau:

Nhap so cua thang : 15

H32

Không thực hiện.

Chúng ta thêm mục default như sau vào sau case cuối cùng:

default:

```
Console.WriteLine("Khong co thang nay");
```

```
break;
```

Cuối cùng ta có đoạn code như sau:

```
// TODO: Add code to start application here
```

```
byte so;
```

```
Console.WriteLine("Nhap so cua thang : ");
```

```
so=byte.Parse(Console.ReadLine());
```

```
switch (so)
```

```
{
```

```
    case 1:
```

```
        Console.WriteLine("Thang Gieng");
```

```
        break;
```

```
    case 2:
```

```
        Console.WriteLine("Thang hai");
```

```
        break;
```

```
    case 3:
```

```
        Console.WriteLine("Thang ba");
```

```
        break;
```

```
    case 4:
```

```
        Console.WriteLine("Thang tu");
```

```
        break;
```

```
    case 5:
```

```
        Console.WriteLine("Thang nam");
```

```
        break;
```

```
    case 6:
```

```
        Console.WriteLine("Thang sau");
```

```

        break;
    case 7:
        Console.WriteLine("Thang bay");
        break;
    case 8:
        Console.WriteLine("Thang tam");
        break;
    case 9:
        Console.WriteLine("Thang chin");
        break;
    case 10:
        Console.WriteLine("Thang muoi");
        break;
    case 11:
        Console.WriteLine("Thang muoi mot");
        break;
    case 12:
        Console.WriteLine("Thang muoi hai");
        break;
    default:
        Console.WriteLine("Khong co thang nay");
        break;
}

Console.ReadLine();

```

■ Nhấn phím F5 để chạy thử chương trình, nhập số 24, chúng ta sẽ có kết quả như hình sau:

 E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

Nhap so cua thang : 24
 Khong co thang nay

H33

Vì không có giá trị nào sau case là 24, nên sẽ thực hiện các câu lệnh nằm sau default. Do đó sẽ hiện câu "Khong co thang nay"

Nếu bạn quy ước nếu bạn nào nhập từ 1 đến 12, thì sẽ in ra tháng tương ứng. Nếu nhập số 15 thì sẽ quay lại nhánh case có giá trị là 12. còn lại sẽ thông báo câu lệnh sau default.

Bạn thử sửa code lại như sau:

```
// TODO: Add code to start application here
```

```
byte so;
```

```
Console.Write("Nhap so cua thang. : ");
```

```
so=byte.Parse(Console.ReadLine());
```

```
switch (so)
```

```
{
```

```
    case 1:
```

```
        Console.Write("Thang Gieng");
```

```
        break;
```

```
    case 2:
```

```
        Console.Write("Thang hai");
```

```
        break;
```

```
    case 3:
```

```
        Console.Write("Thang ba");
```

```
        break;
```

```
    case 4:
```

```
        Console.Write("Thang tu");
```

```
        break;
```

```
    case 5:
```

```
        Console.Write("Thang nam");
```

```
        break;
```

```
    case 6:
```

```
        Console.Write("Thang sau");
```

```
        break;
```

```
    case 7:
```

```
        Console.Write("Thang bay");
```

```
        break;
```

```
    case 8:
```

```
        Console.Write("Thang tam");
```

```
        break;
```

```
    case 9:
```

```

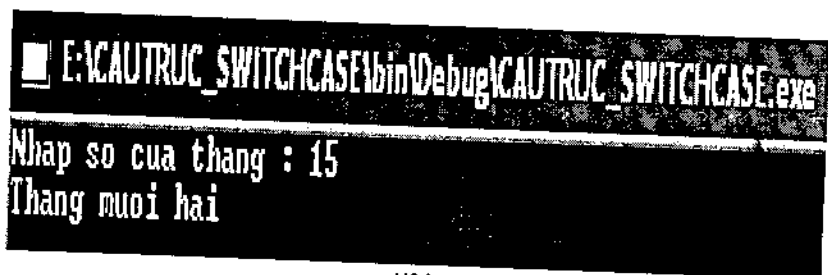
        Console.WriteLine("Thang chin");
        break;
    case 10:
        Console.WriteLine("Thang muoi");
        break;
    case 11:
        Console.WriteLine("Thang muoi mot");
        break;
    case 12:
        Console.WriteLine("Thang muoi hai");
        break;
    case 15:
        goto case 12;
    default:
        Console.WriteLine("Khong co thang nay");
        break;
}

Console.ReadLine();

```

Giải thích:

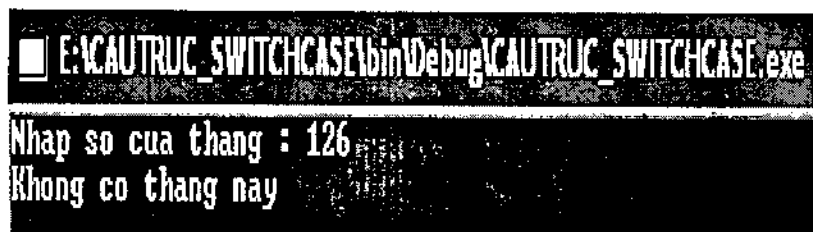
- Nếu nhập số **15**, thỏa với giá trị theo sau **case 15**: nên sẽ thực hiện câu lệnh **goto case 12**; sẽ chuyển đến nhánh **case** có giá trị **12**, do đó sẽ thực hiện câu lệnh nằm sau **case 12**.
- Nhấn phím **F5** để chạy thử chương trình, nhập số **15**, chúng ta sẽ có kết quả như hình sau:



H34

Bạn thấy sẽ thực hiện câu lệnh của case 12.

■ Chạy lại chương trình, giả sử nhập một số khác 1 đến 12 và số 15 thì sẽ thực hiện câu lệnh sau default. Giả sử nhập số 126, chúng ta sẽ có kết quả như hình sau:



H35

Như đã đề cập ở trên, trong C# những biểu thức điều kiện có thể là kiểu chuỗi (string)

Sửa lại chương trình nhập vào một tháng trong năm có kiểu chuỗi, nếu là:

Tháng giêng thì nhập **tháng1**

Tháng hai thì nhập là **tháng2**

Tháng ba thì nhập là **tháng3**

Tháng tư thì nhập là **tháng4**

Tháng năm thì nhập là **tháng5**

Tháng sáu thì nhập là **tháng6**

Tháng bảy thì nhập là **tháng7**

Tháng tám thì nhập là **tháng8**

Tháng chín thì nhập là **tháng9**

Tháng mười thì nhập là **tháng10**

Tháng mười một thì nhập là **tháng11**

Tháng mười hai thì nhập là **tháng12**

Chương trình sẽ báo ra một chuỗi là tên của tháng đó.

◆ Bạn vào cửa sổ viết code và sửa lại như sau:

```
string thang;  
Console.Write("Nhập số của tháng : ");  
thang=Console.ReadLine();  
switch (thang)  
{
```

```
case "thang1":
    Console.Write("Thang Gieng");
    break;
case "thang2":
    Console.Write("Thang hai");
    break;
case "thang3":
    Console.Write("Thang ba");
    break;
case "thang4":
    Console.Write("Thang tư");
    break;
case "thang5":
    Console.Write("Thang nam");
    break;
case "thang6":
    Console.Write("Thang sáu");
    break;
case "thang7":
    Console.Write("Thang bảy");
    break;
case "thang8":
    Console.Write("Thang tám");
    break;
case "thang9":
    Console.Write("Thang chín");
    break;
case "thang10":
    Console.Write("Thang mười");
    break;
case "thang11":
    Console.Write("Thang mười một");
    break;
case "thang12":
    Console.Write("Thang mười hai");
```

```

        break;
    default:
        Console.WriteLine("Khong co thang nay");
        break;
}
Console.ReadLine();

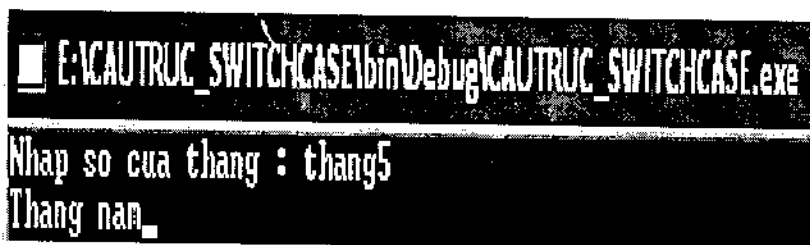
```

Giải thích:

- Khai báo biến thang có kiểu string
- Hiện ra câu "Nhap so cua thang :"
- Nhập vào tên của tháng, đó là (thang1, thang2, ..., thang12)
- Cấu trúc switch sẽ kiểm tra biến thang với các giá trị theo sau case. Nếu thỏa sẽ xuất hiện ra màn hình tên của tháng tương ứng.

Ghi chú:

- Giá trị sau case là chuỗi, các bạn nhớ có cặp dấu nháy kép.
- Biên dịch và chạy thử chương trình, giả sử nhập vào thang5. Bạn sẽ có kết quả như hình sau:



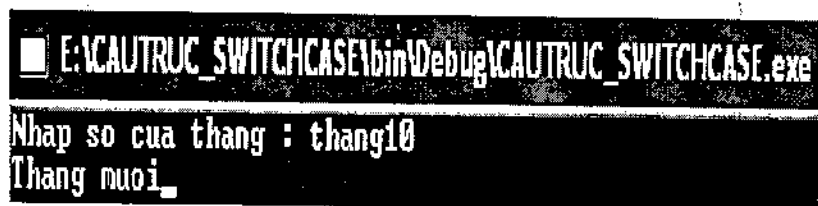
```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
Nhap so cua thang : thang5
Thang nam_

```

H36

- Nếu nhập thang10, chúng ta sẽ có kết quả thông báo như hình sau:



```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
Nhap so cua thang : thang10
Thang muoi_

```

H37

Viết chương trình nhập vào hai số a và b, nhập các phép toán cộng (+), trừ (-), nhân (*), chia (/) và tính số dư (%). Chương trình sẽ xuất hiện kết quả.

Giả sử bạn đã vào Microsoft Visual Studio .NET. Bạn sẽ viết code như sau vào sau câu ghi chú:

```
float a, b, ketqua;
Console.Write("Nhap so a = ");
a=float.Parse(Console.ReadLine());
Console.Write("Nhap so b = ");
b=float.Parse(Console.ReadLine());
string toantu;
Console.Write("Nhap toan tu : ");
toantu=Console.ReadLine();
switch (toantu)
{
    case "+":
        Console.Write("a + b = ");
        ketqua = a + b;
        Console.Write(ketqua);
        break;
    case "-":
        Console.Write("a - b = ");
        ketqua = a - b;
        Console.Write(ketqua);
        break;
    case "*":
        Console.Write("a * b = ");
        ketqua = a * b;
        Console.Write(ketqua);
        break;
    case "/":
        Console.Write("a / b = ");
        ketqua = a / b;
        Console.Write(ketqua);
```

```

        break;
    case "%":
        Console.WriteLine("So du cua a / b = ");
        ketqua = a % b;
        Console.WriteLine(ketqua);
        break;
    default:
        Console.WriteLine("Ban da nhap sai toan tu");
        break;
}
Console.ReadLine();

```

Giải thích:

- Khai báo các biến a, b và ketqua có kiểu số thực (float)
- Nhập vào số a, số b
- Khai báo biến toantu có kiểu chuỗi (string)
- Nhập vào toán tử: {+, -, *, / hoặc %}
- So sánh toán tử vừa nhập với các giá trị theo sau các case, nếu thỏa sẽ thực hiện các câu lệnh của case đó.
- Gán biến ketqua = a + b
- In ra biến ketqua "Console.WriteLine(ketqua)"

Biên dịch và chạy thử chương trình, giả sử nhập a = 2, b = 4. Nhập dấu +. Ta được kết quả như hình sau:

```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
Nhap so a = 2
Nhap so b = 4
Nhap toan tu : +
a + b = 6_

```

H38

- Chạy chương trình, giả sử nhập a = 2, b = 4. Nhập dấu -. Ta được kết quả như hình sau:

■ E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

Nhap so a = 2
Nhap so b = 4
Nhap toan tu : -
a - b = -2_

H39

■ Chạy chương trình, giả sử nhập a = 2, b = 4. Nhập dấu *. Ta được kết quả như hình sau:

■ E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

Nhap so a = 2
Nhap so b = 4
Nhap toan tu : *
a * b = 8

H40

■ Chạy chương trình, giả sử nhập a = 2, b = 4. Nhập dấu /. Ta được kết quả như hình sau:

■ E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

Nhap so a = 2
Nhap so b = 4
Nhap toan tu : /
a / b = 0.5_

H41

■ Chạy chương trình, giả sử nhập a = 5, b = 2. Nhập dấu %. Ta được kết quả như hình sau:


```
Nhap so a = 5
Nhap so b = 2
Nhap toan tu : %
So du cua a / b = 1
```

H42

Đôi khi người nhập thường nhầm phép nhân trong máy tính là x hoặc X (thay vì đúng là *). Bạn có thể sửa lại chương trình như sau:

```
float a, b, ketqua;
Console.Write("Nhap so a = ");
a=float.Parse(Console.ReadLine());
Console.Write("Nhap so b = ");
b=float.Parse(Console.ReadLine());
string toantu;
Console.Write("Nhap toan tu : ");
toantu=Console.ReadLine();
switch (toantu)
{
    case "+":
        Console.Write("a + b = ");
        ketqua = a + b;
        Console.Write(ketqua);
        break;
    case "-":
        Console.Write("a - b = ");
        ketqua = a - b;
        Console.Write(ketqua);
        break;
    case "**":
        Console.Write("a * b = ");
        ketqua = a * b;
```

```

        Console.Write(ketqua);
        break;
    case "/":
        Console.Write("a / b = ");
        ketqua = a / b;
        Console.Write(ketqua);
        break;
    case "%":
        Console.Write("So du cua a / b = ");
        ketqua = a % b;
        Console.Write(ketqua);
        break;
    case "x":
    case "X":
        goto case "**";
    default:
        Console.Write("Ban da nhap sai toan tu");
        break;
}
Console.ReadLine();

```

■ Bạn thêm các lệnh in đậm vào vị trí như trên, nếu toán tử nhập vào là "x" hoặc "X" thì sẽ thực hiện lệnh goto case "**"; trở về nhánh có case là "**" và thực hiện phép nhân.

Viết hai cách sau đều đúng. Bạn thử suy nghĩ tại sao chúng ta không viết:

```

case "x":
    goto case "**";
case "X":
    goto case "**";

```

mà viết:

```

case "x":
case "X":
    goto case "**";

```

Bạn thử suy nghĩ và tự giải thích!!!

BÀI 6

CẤU TRÚC LẬP FOR

Cấu trúc này thường được dùng khi xác định được số lần lặp.
for ([khởi động giá trị đầu]; [điều kiện]; [các lệnh trước khi lặp])

Ví dụ:

Viết chương trình xuất hiện ra 12 tháng trong năm.

* Vào Microsoft Visual Studio .NET

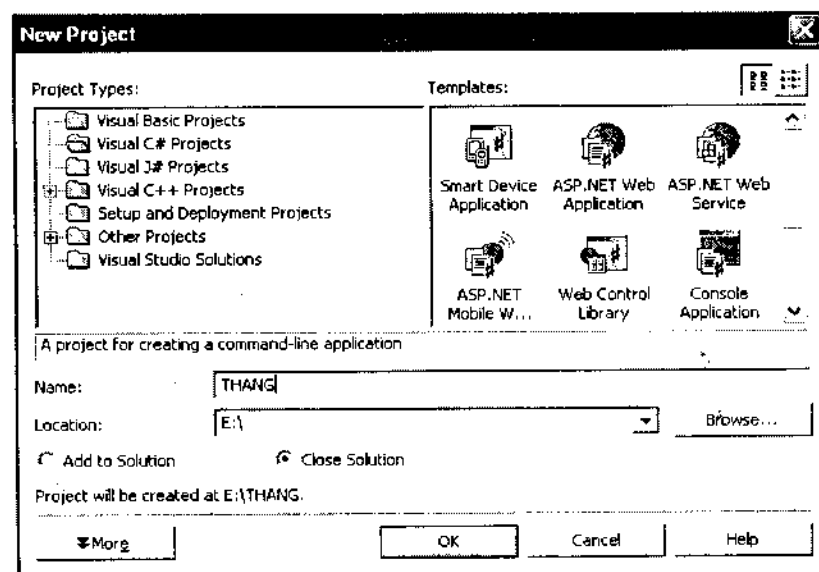
* Click chuột vào menu **File**, chọn **New**, chọn **Project**

* Trong cửa sổ **New Project** hiện ra, bạn chọn:

* **Visual C# Project** trong khung bên trái có tên **Project Types**.

* Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.

* Đặt tên **THANH** trong khung **Name** như hình sau:



H43

* Click nút **OK**.

Viết code của chương trình sau câu ghi chú như sau:

```
// TODO: Add code to start application here
```

```
byte x;
```

```
for (x=1; x<=12; x++)
```

```
{
```

```
    Console.Write("DAY LA THANG: ");
```

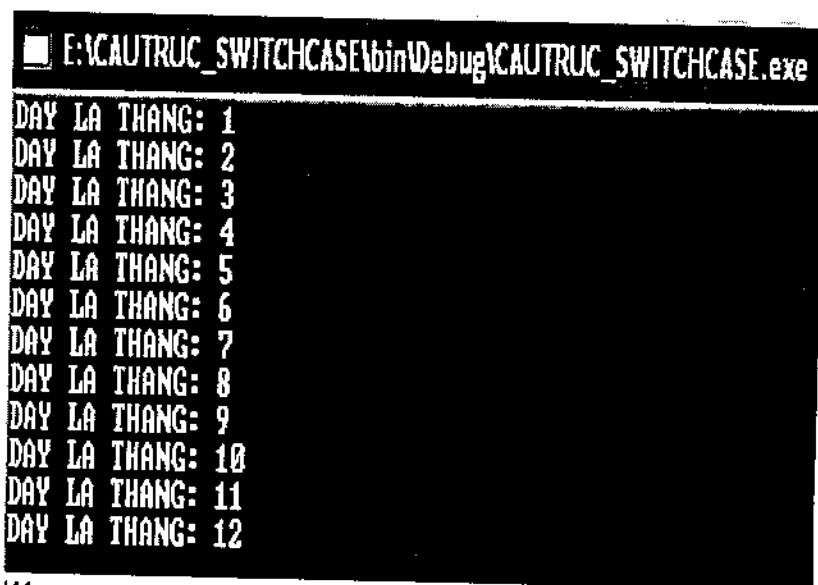
```
    Console.WriteLine(x);
```

```
}
```

```
Console.ReadLine();
```

Giải thích:

- Khai báo biến x có kiểu nguyên (byte).
- Vòng lặp for với giá trị ban đầu của x=1, thực hiện khi x còn nhỏ hơn hoặc bằng 12, tăng biến x lên 1 đơn vị sau mỗi lần lặp (x++ sẽ tăng lên 1 đơn vị so với giá trị của x trước đó)
- Biên dịch và chạy thử chương trình, bạn sẽ thấy kết quả như hình sau:



```
E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
DAY LA THANG: 1
DAY LA THANG: 2
DAY LA THANG: 3
DAY LA THANG: 4
DAY LA THANG: 5
DAY LA THANG: 6
DAY LA THANG: 7
DAY LA THANG: 8
DAY LA THANG: 9
DAY LA THANG: 10
DAY LA THANG: 11
DAY LA THANG: 12
```

H44

- Bạn có thể dùng vòng lặp for lặp từ số lớn đến số nhỏ.
- Vào cửa sổ viết code, sửa lại cấu trúc lặp for như sau:

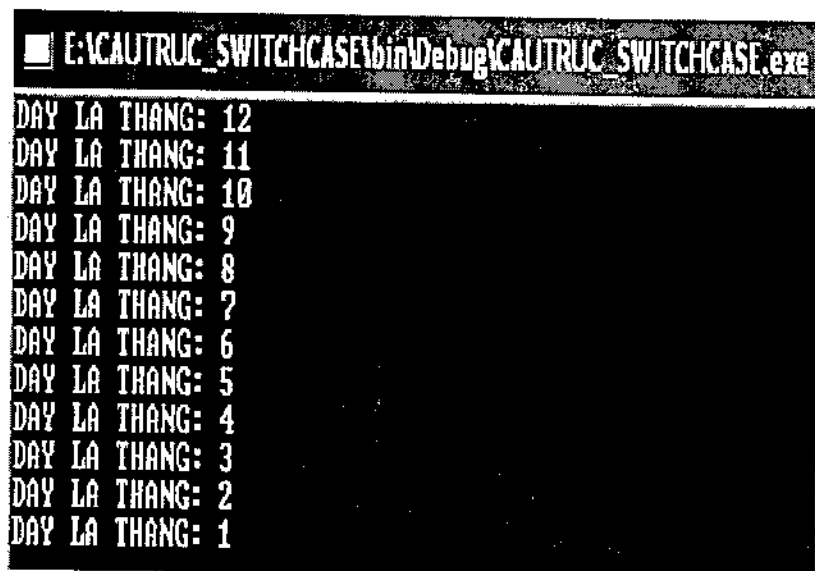
```

byte x;
for (x=12; x>=1; x--)
{
    Console.Write("DAY LA THANG: ");
    Console.WriteLine(x);
}
Console.ReadLine();

```

Giải thích:

- Vòng lặp for với giá trị ban đầu của x là 12, sau mỗi lần lặp sẽ giảm 1 đơn vị so với giá trị của x trước đó. (x -- sẽ giảm 1 đơn vị sau mỗi lần lặp)
- Nhấn F5 chạy thử chương trình, bạn sẽ thấy kết quả như hình sau:



```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
DAY LA THANG: 12
DAY LA THANG: 11
DAY LA THANG: 10
DAY LA THANG: 9
DAY LA THANG: 8
DAY LA THANG: 7
DAY LA THANG: 6
DAY LA THANG: 5
DAY LA THANG: 4
DAY LA THANG: 3
DAY LA THANG: 2
DAY LA THANG: 1

```

H45

Ví dụ:

Viết chương trình tính tổng các số lẻ từ 1 đến 2006

Bạn tạo một Project mới với một tên gì đó, viết code như sau:

```

int x;
long tong=0;

```

```
for (x=1; x<2006; x=x+2)
```

```
tong=tong+x;
```

```
Console.WriteLine("TONG CAC SO LE TU 1 DEN 2006 LA: ");
```

```
Console.WriteLine(tong);
```

```
Console.ReadLine();
```

Giải thích:

- Khai báo biến `x` có kiểu nguyên (`int`)
- Khai báo biến `tong` có kiểu số nguyên (`long`) và gán cho biến này có giá trị là 0.
- Vòng lặp `for` có giá trị ban đầu `x = 1`, thực hiện khi `x` còn nhỏ hơn 2006, tăng lên 2 đơn vị sau mỗi lần lặp so với giá trị trước đó của `x`. Đó chính là các số lẻ từ 1 đến 2005 (1, 3, 5, 7, ..., 2003, 2005).
- Cứ mỗi lần lặp của vòng lặp `for` sẽ lấy biến `tong` cộng với giá trị của `x`.
- Báo ra màn hình lời ghi chú và giá trị của biến `tong` tương ứng với lần lặp cuối cùng tức là `x = 2005` (tăng thêm 2 đơn vị nữa, lúc này `x` sẽ là 2007. Không thỏa điều kiện `x < 2006`. Vòng lặp kết thúc).
- Biên dịch và chạy thử chương trình, chúng ta sẽ thấy kết quả như sau:



H46

Ví dụ:

Nhập và tính tổng của 5 số thực được nhập vào.

Bạn viết code như sau:

```
int x;
```

```
float so,tong=0;
```

```
for (x=1; x<=5; x++)
```

```
{  
    Console.WriteLine("BAN NHAP SO THU ");  
    Console.Write(x);
```

```

Console.Write(" = ");
so=float.Parse(Console.ReadLine());
tong=tong+so;
}
Console.Write("TONG 5 SO THUC LA: ");
Console.WriteLine(tong);
Console.ReadLine();

```

Giải thích:

- Khai báo biến x có kiểu số nguyên (int)
- Khai báo biến so và tong có kiểu số thực (float)
- Vòng lặp for chạy từ 1 đến 5 (tức là lặp 5 lần)
- Hiện ra câu nhắc để nhập số thực
- Đổi biến so có kiểu số thực
- Cộng số thực vừa nhập vào với biến tong là số trước đó
- Báo ra tổng của 5 số thực vừa nhập vào

Biên dịch và chạy thử chương trình, giả sử nhập vào 5 số thực lần lượt như trong hình, đó là 12.5, 20.4, 10.1, 34.6, 235.8. Chúng ta sẽ có kết quả như hình dưới

■ E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

```

BAN NHAP SO THU 1 = 12.5
BAN NHAP SO THU 2 = 20.4
BAN NHAP SO THU 3 = 10.1
BAN NHAP SO THU 4 = 34.6
BAN NHAP SO THU 5 = 235.8
TONG 5 SO THUC LA: 313.4

```

H47

BÀI 7

CẤU TRÚC LẶP WHILE/DO WHILE

Các vòng lặp này dừng khi biểu-thức [Điều kiện] nhận giá trị sai.

while (Điều kiện)

```
{  
    S1;  
    S2;  
    .....  
    Sn  
}
```

Ví dụ:

Chúng tôi lấy bài tập vừa thực hiện ở vòng lặp for ở trên, đó là nhập vào 5 số thực, tính tổng của 5 số thực đó. Nhưng yêu cầu các bạn phải dùng vòng lặp while.

- * Vào Microsoft Visual Studio .NET
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**
- * Trong cửa sổ **New Project** hiện ra, bạn chọn:
- * **Visual C# Project** trong khung bên trái có tên **Project Types**.
- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console**

Application.

- * Đặt tên cho project trong khung **Name**

Viết code vào sau dòng ghi chú của chương trình như sau:

```
int x=1;  
float so,tong=0;  
while (x<6)  
{  
    Console.Write("BAN NHAP SO THU ");  
    Console.Write(x);  
    Console.Write(" = ");
```



```

so=float.Parse(Console.ReadLine());
tong=tong+so;
x=x+1;
}
Console.Write("TONG 5 SO THUC LA: ");
Console.WriteLine(tong);
Console.ReadLine();

```

Giải thích:

- Khai báo biến x có kiểu số nguyên (int)
- Khai báo biến so và tong có kiểu số thực (float)
- Vòng lặp while thực hiện từ x=1 đến khi x< 6 (tức là đến x=5)
- Hiện ra câu nhắc để nhập số thực
- Đổi biến so có kiểu số thực
- Cộng số thực vừa nhập vào với biến tong là số trước đó
- Tăng biến x lên một đơn vị
- Báo ra tổng của 5 số thực vừa nhập vào

Biên dịch và chạy thử chương trình, giả sử nhập vào 5 số thực lần lượt là: 5.1, 5.2, 5.3, 5.4, 5.5. Bạn sẽ thấy kết quả như hình sau:

```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
BAN NHAP SO THU 1 = 5.1
BAN NHAP SO THU 2 = 5.2
BAN NHAP SO THU 3 = 5.3
BAN NHAP SO THU 4 = 5.4
BAN NHAP SO THU 5 = 5.5
TONG 5 SO THUC LA: 26.5

```

H48

Bạn thấy kết quả cũng như vòng lặp for ở bài trước.

đo

```

{
    S1;
    S2;
    .....
    Sn;
} while (Điều kiện);

```

Cũng thực hiện nội dung của bài tập trên, nhưng dùng phát biểu lặp do ... while vừa nêu.

```

int x=1;
float so,tong=0;
do
{
    Console.Write("BAN NHAP SO THU ");
    Console.Write(x);
    Console.Write(" = ");
    so=float.Parse(Console.ReadLine());
    tong=tong+so;
    x=x+1;
} while (x<6);
Console.Write("TONG 5 SO THUC LA: ");
Console.WriteLine(tong);
Console.ReadLine();

```

Giải thích:

Bạn tự tìm hiểu đoạn code trên.

☐ Phát biểu **break**

Phát biểu này dùng để ngưng các vòng lặp, mặc dù điều kiện của vòng lặp còn đúng. Các bạn đã biết phát biểu này trong cấu trúc switch case.

Bạn thử thêm phát biểu break trong code của bài tập trên như sau:

```

int x=1;
float so,tong=0;
do
{

```

```

Console.Write("BAN NHAP SO THU ");
Console.Write(x);
Console.Write(" = ");
so=float.Parse(Console.ReadLine());
tong=tong+so;
x=x+1;
if (x==3)
    break;
} while (x<6);
Console.Write("TONG 5 SO THUC LA: ");
Console.WriteLine(tong);
Console.ReadLine();

```

■ Bạn thêm phát biểu if, nếu $x = 3$ thì sẽ thực hiện phát biểu break. Lúc này $x = 3$ vẫn còn thỏa điều kiện trong vòng lặp do ... while ở trên, nhưng nó vẫn thoát ra khỏi vòng lặp.

■ Chạy thử chương trình nhập 5 số thực. Bạn chỉ nhập được hai số thì vòng lặp kết thúc, vì lúc này biến $x = 3$. Các câu lệnh phía sau vòng lặp sẽ được thực hiện. Bạn xem hình sau:

 E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

```

BAN NHAP SO THU 1 = 2.5
BAN NHAP SO THU 2 = 7.5
TONG 5 SO THUC LA: 10

```

H49

Chúng tôi nhập số thứ nhất là 2.5, số thứ hai là 7.5, lúc này $x = 3$ nên thực hiện phát biểu break thoát ra khỏi vòng lặp và thực hiện các câu lệnh sau vòng lặp.

▣ Phát biểu goto

Bạn đã biết phát biểu này trong cấu trúc switch case. Bạn xem ví dụ sau:

```

char nhayden;
int x=1;

```

```

float so,tong=0;
do
{
    Console.Write("BAN NHAP SO THU ");
    Console.Write(x);
    Console.Write(" = ");
    so=float.Parse(Console.ReadLine());
    tong=tong+so;
    x=x+1;
    if (x==3)
        goto nhayden;
} while (x<6);
Console.Write("TONG 5 SO THUC LA: ");
Console.WriteLine(tong);
nhayden::
if (x==3)
    Console.Write("TAI SAO KHONG BAO RA TONG CUA 5 SO THUC!!!");
Console.ReadLine();

```

Giải thích:

- Khai báo thêm biến nhayden có kiểu char
- Khi biến x = 3 thì thực hiện phát biểu goto nhayden; lệnh này nhảy đến nhãn nhayden; vì x = 3 nên thực hiện câu lệnh của phát biểu if đó là xuất hiện câu "TAI SAO KHONG BAO RA TONG CUA 5 SO THUC!!!"
- Chạy thử chương trình, bạn sẽ thấy kết quả như sau:

 E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

```

BAN NHAP SO THU 1 = 123.7
BAN NHAP SO THU 2 = 3456.5
BAN BIET TAI SAO KHONG BAO RA TONG CUA 5 SO THUC!!!_

```

BÀI 8

MẢNG TRONG C#

Để khai báo nhiều biến có cùng tên nhưng phân biệt nhau bởi chỉ số, chúng ta dùng dữ liệu mảng. Mảng trong C# quy ước khởi đầu với chỉ số 0.

☞ Mảng một chiều

Kiểu của mảng[] tên mảng;

Ví dụ:

Int [] cam;

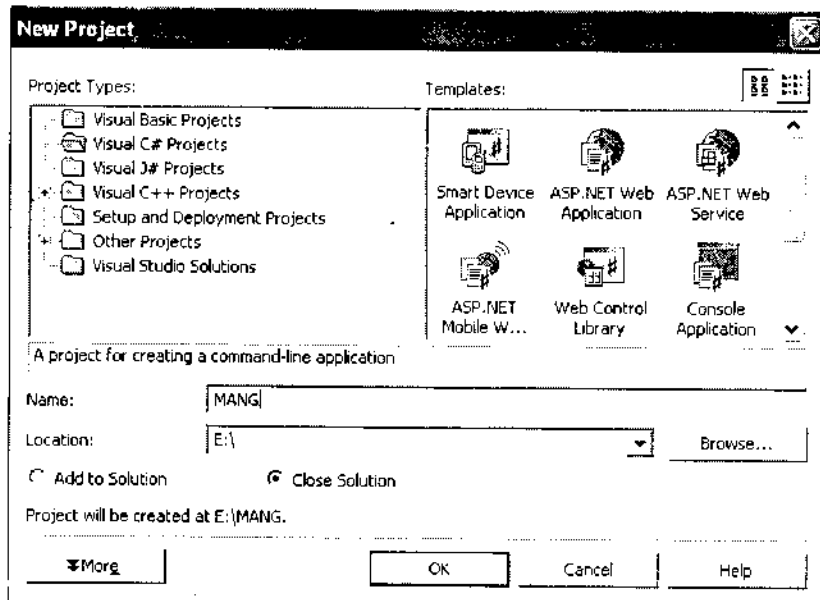
Float [] so;

Mảng trong C# được cấp phát bộ nhớ bằng toán tử **new**. Bạn sẽ thấy trong bài tập thực hành.

Một đặc điểm của C# không giống với các ngôn ngữ truyền thống C và C++ đó là người lập trình không cần quan tâm đến việc giải phóng bộ nhớ, hệ thống sẽ thực hiện việc này nhờ cơ chế nhặt rác. Bạn xem một bài tập về mảng như sau:

Viết chương trình nhập vào chỉ số mảng bất kỳ, nhập các phần tử đó vào, xuất các phần tử của mảng vừa nhập ra màn hình.

- * Vào Microsoft Visual Studio .NET
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**
- * Trong cửa sổ **New Project** hiện ra, bạn chọn:
- * **Visual C# Project** trong khung bên trái có tên **Project Types**.
- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.
- * Đặt tên **MANG** trong khung **Name** như hình sau:



H51

* Click nút OK

Viết code tại vị trí ghi chú của chương trình như sau:

```
float[] mang;
int i, sophantu;
Console.WriteLine("BAN NHAP SO PHAN TU CUA MANG: ");
sophantu=int.Parse(Console.ReadLine());
mang=new float[sophantu];
// Nhập các phần tử của mảng
for (i=0; i<sophantu; i++)
{
    Console.WriteLine("GIA TRI PHAN TU THU ");
    Console.WriteLine(i);
    Console.WriteLine(" LA : ");
    mang[i]=float.Parse(Console.ReadLine());
}
// Xuất các phần tử của mảng ra màn hình
for (i=0; i<sophantu; i++)
```

```

{
    Console.Write("PHAN TU THU ");
    Console.Write(i);
    Console.Write(" LA : ");
    Console.WriteLine(mang[i]);
}
Console.ReadLine();

```

Giải thích:

- Khai báo một mảng kiểu số thực (float) có tên là mang.
- Khai báo biến i và biến sophantu có kiểu nguyên. Biến i dùng trong vòng lặp for, biến sophantu là số phần tử của mảng.
- Nhập vào số phần tử của mảng.
- Cấp phát bộ nhớ cho mảng
- Vòng lặp for thứ nhất dùng để nhập các phần tử của mảng
- Vòng lặp thứ hai để xuất ra các phần tử của mảng
- Biên dịch và chạy thử chương trình, giả sử bạn nhập mảng có 6 phần tử, sau đó nhập giá trị của 6 phần tử này. Nhập xong, bạn sẽ thấy kết quả các số vừa nhập hiện ra.

Nhìn vào hình sau để đối chiếu với những lời chúng tôi vừa nói ở trên.

```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
BAN NHAP SO PHAN TU CUA MANG: 6
GIA TRI PHAN TU THU 0 LA : 123.6
GIA TRI PHAN TU THU 1 LA : 564.5
GIA TRI PHAN TU THU 2 LA : 12.3
GIA TRI PHAN TU THU 3 LA : 45.9
GIA TRI PHAN TU THU 4 LA : 78.9
GIA TRI PHAN TU THU 5 LA : 456.8

```

■ Khi nhập số 6 chỉ số phần tử có trong mảng và giá trị của 6 phần tử như hình trên. Nếu gõ phím Enter, bạn sẽ thấy kết quả như hình sau:

```
E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
BAN NHAP SO PHAN TU CUA MANG: 6
GIA TRI PHAN TU THU 0 LA : 123.6
GIA TRI PHAN TU THU 1 LA : 564.5
GIA TRI PHAN TU THU 2 LA : 12.3
GIA TRI PHAN TU THU 3 LA : 45.9
GIA TRI PHAN TU THU 4 LA : 78.9
GIA TRI PHAN TU THU 5 LA : 456.8
PHAN TU THU 0 LA : 123.6
PHAN TU THU 1 LA : 564.5
PHAN TU THU 2 LA : 12.3
PHAN TU THU 3 LA : 45.9
PHAN TU THU 4 LA : 78.9
PHAN TU THU 5 LA : 456.8
```

H53

Bạn xem hình sau và tự tìm hiểu:

```
E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
BAN NHAP SO PHAN TU CUA MANG: 10
GIA TRI PHAN TU THU 0 LA : 1
GIA TRI PHAN TU THU 1 LA : 2
GIA TRI PHAN TU THU 2 LA : 3
GIA TRI PHAN TU THU 3 LA : 4
GIA TRI PHAN TU THU 4 LA : 5
GIA TRI PHAN TU THU 5 LA : 6
GIA TRI PHAN TU THU 6 LA : 7
GIA TRI PHAN TU THU 7 LA : 8
GIA TRI PHAN TU THU 8 LA : 9
GIA TRI PHAN TU THU 9 LA : 10
PHAN TU THU 0 LA : 1
PHAN TU THU 1 LA : 2
PHAN TU THU 2 LA : 3
PHAN TU THU 3 LA : 4
PHAN TU THU 4 LA : 5
PHAN TU THU 5 LA : 6
PHAN TU THU 6 LA : 7
PHAN TU THU 7 LA : 8
PHAN TU THU 8 LA : 9
PHAN TU THU 9 LA : 10
```

H54

Các bạn có thể khai báo một mảng với các phần tử có sẵn bên trong, giả sử khai báo một mảng có 5 phần tử là các số nguyên như sau:

```
int[] so = new int[]{5, 25, 46, 57, 12};
```

Ví dụ:

Bạn viết chương trình xuất ra màn hình 5 phần tử của một mảng chứa 5 phần tử là các số nguyên lần lượt là: 2, 5, 89, 567, 123.

Bạn viết code như sau:

```
int[] mang=new int[]{2, 5, 89, 567, 123};
int i;
for (i=0; i<5; i++)
{
    Console.Write("PHAN TU THU ");
    Console.Write(i+1);
    Console.Write(" LA : ");
    Console.WriteLine(mang[i]);
}
Console.ReadLine();
```

Giải thích:

- Khai báo mảng một chiều tên mang có kiểu là int và có 5 phần tử là 2, 5, 89, 567 và 123
- Khai báo biến i để dùng trong vòng lặp for
- Vòng lặp thực hiện 5 lần
- Biến i + 1 để xuất ra phần tử đầu tiên là phần tử 1
- Biên dịch và chạy thử chương trình, bạn sẽ có kết quả như hình sau:



```
E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
PHAN TU THU 1 LA : 2
PHAN TU THU 2 LA : 5
PHAN TU THU 3 LA : 89
PHAN TU THU 4 LA : 567
PHAN TU THU 5 LA : 123
```

H55

Qua bài tập này, các bạn thấy ví khai báo các phần tử có trong mảng, nên chúng ta bỏ qua vòng lặp for để nhập chúng.

➤ Mảng hai chiều

Kiểu của mảng[,] tên mảng;

Ví dụ:

Int [,] so;

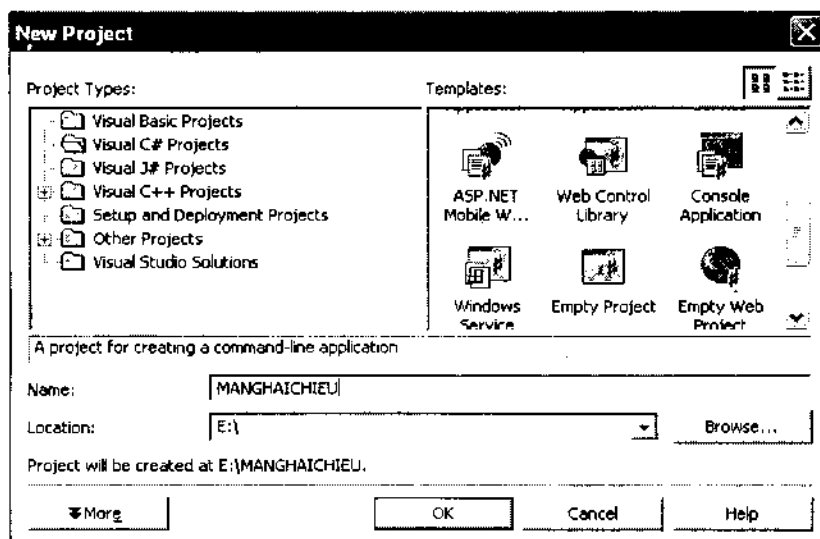
float [,] cacso;

Tương tự như mảng một chiều, ở mảng hai chiều, các bạn cũng có thể gán hoặc truy xuất các phần tử của mảng. Bạn có thể tham khảo các ví dụ sau:

Ví dụ:

Viết chương trình khai báo mảng hai chiều mang[3,3]. Nhập các phần tử của mảng vào, in ra màn hình các phần tử của mảng hai chiều này. Bạn thực hiện như sau:

- * Vào Microsoft Visual Studio .NET
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**
- * Trong cửa sổ **New Project** hiện ra, bạn chọn:
- * **Visual C# Project** trong khung bên trái có tên **Project Types**.
- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.
- * Đặt tên **MANGHAICHIEU** trong khung **Name** như hình sau:



* Click nút OK.

Viết code cho chương trình sau câu ghi chú là:

```
int[ , ] mang;  
int i, j;  
mang=new int[3,3];  
for (i=0; i<3; i++)  
    for (j=0; j<3; j++)  
    {  
        Console.WriteLine("BAN NHAP PHAN TU THU MANG[");  
        Console.WriteLine(i+1);  
        Console.WriteLine(",");  
        Console.WriteLine(j+1);  
        Console.WriteLine("] LA :");  
        mang[i,j]=int.Parse(Console.ReadLine());  
    }  
Console.WriteLine("CAC PHAN TU CUA MANG LA: ");  
for (i=0; i<3; i++)  
    for (j=0; j<3; j++)  
    {  
        Console.WriteLine(mang[i,j]);  
        Console.WriteLine(" ");  
    }  
Console.ReadLine();
```

Giải thích:

- Khai báo mảng hai chiều kiểu nguyên int có tên là mang
- Khai báo biến i và j có kiểu nguyên
- Cấp phát bộ nhớ cho mảng hai chiều có 3 hàng 3 cột
- Hai vòng lặp for lồng nhau dùng để nhập vào 9 phần tử của mảng
 - i + 1 và j + 1 trong các câu lệnh của hai vòng lặp for để xuất hiện đúng phần tử thứ 1, 1, phần tử thứ 1, 2, phần tử thứ 1, 3, phần tử thứ 2, 1... cho đến phần tử thứ 3, 3.
 - Trong hai vòng lặp for lồng nhau ở trên, đầu tiên biến i = 0, các biến j của vòng for bên trong sẽ chạy từ 0 đến 2. Sau đó biến i sẽ

tăng lên một tức là $i = 1$, các biến j của vòng for bên trong sẽ chạy từ 0 đến 2, sau đó biến i tăng lên một tức là $i = 2$, biến j sẽ chạy từ 0 đến 2. Biến i tăng lên một đơn vị, tức là $i = 3$, không thỏa điều kiện của ($i < 3$) nên kết thúc vòng lặp. Ta nhập tất cả là 9 phần tử.

- Xuất hiện trên màn hình câu "CAC PHAN TU CUA MANG LA: ". Nếu bạn để câu này trong vòng lặp for, nó sẽ xuất hiện lặp đi lặp lại nhiều lần.

- Hai vòng lặp for tiếp theo sẽ xuất 9 phần tử của mảng hai chiều ra màn hình.

- Biên dịch và chạy thử chương trình, bạn nhập 9 phần tử vào, giả sử chúng tôi nhập 9 phần tử bất kỳ, các bạn sẽ có kết quả như hình sau:

```
E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
BAN NHAP PHAN TU THU MANG[1,1] LA :67
BAN NHAP PHAN TU THU MANG[1,2] LA :123
BAN NHAP PHAN TU THU MANG[1,3] LA :345
BAN NHAP PHAN TU THU MANG[2,1] LA :642
BAN NHAP PHAN TU THU MANG[2,2] LA :12356
BAN NHAP PHAN TU THU MANG[2,3] LA :34567
BAN NHAP PHAN TU THU MANG[3,1] LA :6788
BAN NHAP PHAN TU THU MANG[3,2] LA :89765
BAN NHAP PHAN TU THU MANG[3,3] LA :34567
CAC PHAN TU CUA MANG LA:
67 123 345 642 12356 34567 6788 89765 34567
```

H57

Khi chạy chương trình, bạn sẽ thấy trên màn hình hiện câu:
BAN NHAP PHAN TU THU MANG[1, 1] LA :

Chúng ta nhập 67 và gõ phím Enter, câu thứ hai hiện ra, chúng ta nhập 123, gõ phím Enter nhập tiếp tục số thứ ba, thực hiện tương tự cho đến khi nhập đến phần tử cuối cùng. Gõ phím Enter, bạn sẽ thấy xuất hiện

CÁC PHÂN TỬ CỦA MẢNG LÀ:

67 123 345 642 12356 34567 6788 89765 34567

Gõ phím Enter, nhấn F5 chạy lại chương trình, bạn nhập thử một kết quả khác để kiểm chứng lại.

Bài trên chúng ta tự cho mảng hai chiều [3, 3] ngay từ trong chương trình. Nếu người sử dụng yêu cầu nhập các số này sẽ được nhập từ bàn phím khi chạy chương trình, bạn sửa code lại như sau:

```
int[ , ] mang;  
int i,j,hang,cot;  
Console.Write("BAN NHAP SO HANG CUA MANG: ");  
hang=int.Parse(Console.ReadLine());  
Console.Write("BAN NHAP SO COT CUA MANG: ");  
cot=int.Parse(Console.ReadLine());  
mang=new int[hang,cot];  
for (i=0; i<hang; i++)  
    for (j=0; j<cot; j++)  
    {  
        Console.Write("BAN NHAP PHAN TU THU MANG[");  
        Console.Write(i+1);  
        Console.Write(",");  
        Console.Write(j+1);  
        Console.Write("] LA :");  
        mang[i,j]=int.Parse(Console.ReadLine());  
    }  
Console.WriteLine("CAC PHAN TU CUA MANG LA: ");  
for (i=0; i<hang; i++)  
    for (j=0; j<cot; j++)  
    {  
        Console.Write(mang[i,j]);  
        Console.Write(" ");  
    }  
Console.ReadLine();
```

Giải thích:

- Khai báo mảng hai chiều kiểu nguyên int có tên là mang

- Khai báo biến i, j, hang, cot có kiểu nguyên int. Biến i và j dùng cho vòng lặp for. Biến hang là số hàng của ma trận, biến cot là số cột của ma trận. Hai biến hang và cot sẽ được nhập vào.

- Cấp phát bộ nhớ cho mảng hai chiều này.

- Phần tiếp theo giống như bài trên.

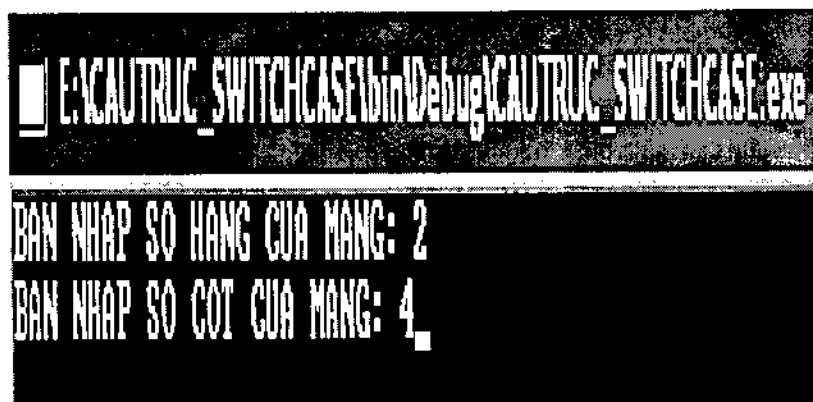
Ghi chú:

Bạn không được cấp phát bộ nhớ cho mảng hai chiều trên khi chưa có giá trị của biến hang và biến cot.

Nếu bạn cấp phát trước, chương trình sẽ báo lỗi. Bạn xem đoạn code sau:

```
int[ , ] mang;  
int i,j,hang,cot;  
mang=new int[hang,cot];  
Console.Write("BAN NHAP SO HANG CUA MANG: ");  
hang=int.Parse(Console.ReadLine());  
Console.Write("BAN NHAP SO COT CUA MANG: ");  
cot=int.Parse(Console.ReadLine());  
for (i=0; i<hang; i++)  
    for (j=0; j<cot; j++)  
    {  
        Console.Write("BAN NHAP PHAN TU THU MANG[");  
        Console.Write(i+1);  
        Console.Write(",");  
        Console.Write(j+1);  
        Console.Write("] LA :");  
        mang[i,j]=int.Parse(Console.ReadLine());  
    }  
Console.WriteLine("CAC PHAN TU CUA MANG LA: ");  
for (i=0; i<hang; i++)  
    for (j=0; j<cot; j++)  
    {  
        Console.Write(mang[i,j]);  
        Console.Write(" ");  
    }  
Console.ReadLine();
```

- Nếu bạn thay đổi vị trí câu lệnh cấp phát bộ nhớ cho mảng hai chiều lên bên trên (câu lệnh đậm). Bạn sẽ bị chương trình báo lỗi.
- Biên dịch và chạy thử chương trình, giả sử nhập hàng=2, cot=4, bạn sẽ thấy như hình sau:



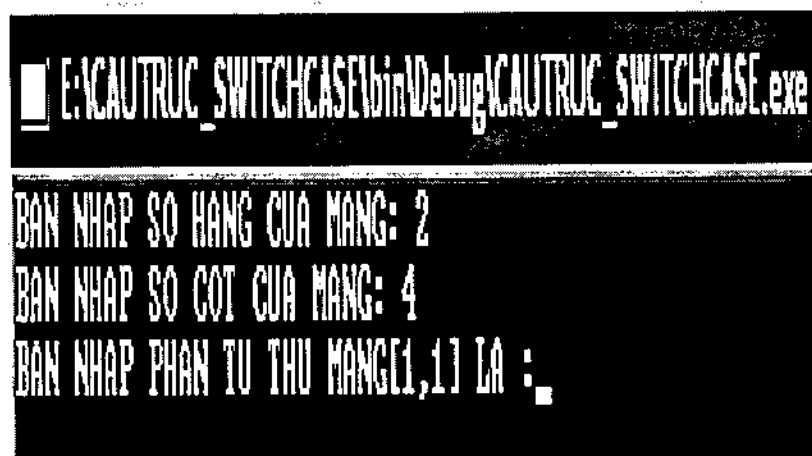
```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
BAN NHAP SO HANG CUA MANG: 2
BAN NHAP SO COT CUA MANG: 4

```

H58

Gõ phím Enter, bạn thấy hình sau:



```

E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe
BAN NHAP SO HANG CUA MANG: 2
BAN NHAP SO COT CUA MANG: 4
BAN NHAP PHAN TU THU MANG[1,1] LA :

```

H59

- Bạn nhập giá trị thứ nhất của mảng vào, gõ phím Enter, và tiếp tục như vậy cho đến khi nhập xong 8 phần tử. Giả sử chúng tôi nhập 8 số nguyên bất kỳ, sau đó gõ phím Enter, các bạn xem kết quả như hình sau:

■ E:\CAUTRUC_SWITCHCASE\bin\Debug\CAUTRUC_SWITCHCASE.exe

```
BAN NHAP SO HANG CUA MANG: 2
BAN NHAP SO COT CUA MANG: 4
BAN NHAP PHAN TU THU MANG[1,1] LA :1
BAN NHAP PHAN TU THU MANG[1,2] LA :3
BAN NHAP PHAN TU THU MANG[1,3] LA :5
BAN NHAP PHAN TU THU MANG[1,4] LA :7
BAN NHAP PHAN TU THU MANG[2,1] LA :9
BAN NHAP PHAN TU THU MANG[2,2] LA :11
BAN NHAP PHAN TU THU MANG[2,3] LA :13
BAN NHAP PHAN TU THU MANG[2,4] LA :15
CAC PHAN TU CUA MANG LA:
1 3 5 7 9 11 13 15
```

H60

BÀI 9

LỚP VÀ ĐỐI TƯỢNG TRONG C#

Khái niệm lớp (**class**) trong các ngôn ngữ lập trình hướng đối tượng dùng để mô tả dữ liệu và các thao tác của các đối tượng trong thế giới thực. Các thành phần dữ liệu cũng được gọi các trường hay các thuộc tính. Các thao tác còn được gọi là hàm hay các phương thức. Chẳng hạn, để mô tả phân số thì thuộc tính sẽ là tử số và mẫu số còn các phương thức là các phép toán về phân số. Để mô tả một hình học thì các thuộc tính là các số đo kích thước còn các phương thức sẽ là các hàm tính diện tích, chu vi ...

Bạn xem các ví dụ sau để hiểu rõ hơn về lớp. Tiêu chí của chúng tôi là dùng những bài tập giản đơn nhất, với mục đích cuối cùng là giúp cho người đọc cảm thấy nhẹ nhàng, thoải mái và dễ thực hiện, dễ tiếp thu từ đó tạo sự thích thú của người đọc với ngôn ngữ lập trình, yêu thích lập trình và tìm tòi nghiên cứu thêm khi cần giải quyết những sự việc.

Bạn xem một lớp thể hiện cách tính diện tích và chu vi của một hình tròn.

```
class vongtron
{
    static double PI=3.1416;
    double R;
    public vongtron(double bk)
    {
        R=bk;
    }
    public double dientich()
    {
        return PI*R*R;
    }
    public double chuvi()
```

```

{
    return 2*PI*R;
}

```

```
};
```

Giải thích:

- Để tạo lớp, chúng ta dùng từ khóa **class** kế đến là tên của lớp cần tạo. Trong đoạn code trên, tên lớp là vongtron.

- Vì trong lớp sẽ có nhiều câu lệnh, chúng ta dùng dấu móc "{" để làm dấu mở bao các câu lệnh, sau các câu lệnh sẽ là dấu móc và dấu chấm phẩy ";", để đóng các câu lệnh, bạn hiểu một cách nôm na như vậy.

- Trong vòng tròn, bạn cần khai báo các thuộc tính cần thiết để có thể tính được diện tích và chu vi của vòng tròn. Bạn thấy có số π và bán kính R . Vậy chúng ta phải khai báo π có kiểu số thực, R có kiểu số thực.

```
static double PI=3.1416;
```

```
double R;
```

- Trong lớp vongtron, khai báo một phương thức (ham) có tên vongtron với một tham số đó là bán kính. Từ khóa Public để có thể gọi được từ bên ngoài lớp. Gán R bằng giá trị trong phương thức. **Đây là phương thức khởi tạo dữ liệu (constructor).**

```
public vongtron(double bk)
```

```
{
```

```
    R=bk;
```

```
}
```

- Trong lớp vongtron, khai báo một phương thức (ham) có tên dientich(). Từ khóa Public để có thể gọi được từ bên ngoài lớp. Trả về giá trị là diện tích của hình tròn là $\pi * R^2$.

```
public double dientich()
```

```
{
```

```
    return PI*R*R;
```

```
}
```

- Trong lớp vongtron, khai báo một phương thức (ham) có tên chuvi(). Từ khóa Public để có thể gọi được từ bên ngoài lớp. Trả về giá trị là chu vi của hình tròn là $2 * \pi * R$.

Ghi chú:

Trong các phương thức ta thấy có phương thức `vongtron` trùng với tên của lớp (phương thức `vongtron(double bk)` ứng với lớp `vongtron`). Đây là phương thức tạo lập (**constructor**) có mục đích khởi tạo dữ liệu, những phương thức này tự động chạy ngay khi đối tượng của lớp được tạo lập. Cơ chế này nhằm khắc phục tình trạng các biến không được khởi tạo giá trị mặc nhiên (nhận giá trị rác). Sau khi xây dựng các lớp như trên, có thể sử dụng bằng cách khai báo các đối tượng của chúng như sau:

Bạn sẽ viết đoạn chương trình chính, trong đó sẽ gọi đến các phương thức của lớp vừa tạo ở trên.

```
class MainProgram
{
    static void Main()
    {
        vongtron vt=new vongtron(3);
        Console.WriteLine("Hình tron :");
        Console.WriteLine(" DT={0},CV={1}",vt.dientich(), vt.chuvi());
        Console.ReadLine();
    }
}
```

Giải thích:

- Đầu tiên là một lớp chương trình chính với hàm `void Main()`.
- Khai báo biến `vt` có kiểu `vongtron` và cấp phát bộ nhớ với bán kính bằng 3.
- Xuất hiện ra màn hình diện tích và chu vi của hình tròn có bán kính là 3.

```
Console.WriteLine(" DT={0},CV={1}",vt.dientich(), vt.chuvi());
```

- Câu lệnh trên trong phương thức `WriteLine`, Giá trị của biến `dientich` và `chuvi` được nhúng vào chuỗi dữ liệu định dạng, hai đối số `vt.dientich()`, `vt.chuvi()` được đánh chỉ số từ 0 đến 1. Trong dòng lệnh trên là kiểu định dạng mặc nhiên đơn giản nhất, ta chỉ việc ghi chỉ số biến kẹp trong hai dấu móc { và }, đó là {0}, {1}, {0} sẽ được thay thế bởi `vt.dientich()` và {1} được thay thế bởi `vt.chuvi()`. Vì `vt.dientich()` có

chỉ số là 0 trong danh sách các đối số sau chuỗi định dạng và `vt.chuoi()` có chỉ số là 1 trong danh sách các đối số sau chuỗi định dạng.

■ Sau đây là những trường hợp định dạng thường dùng trong chuỗi định dạng của phương thức `Console.WriteLine()`

C (hay c, currency): định dạng tiền, số in ra có dấu \$, số âm để trong ngoặc đơn.

Ví dụ:

`Console.WriteLine("{0:C4}", x)` cho kết quả \$1.2000 nếu x là 1,2 và (\$1.2000) nếu x có giá trị là -1,2 (số 4 nghĩa là in 4 chữ số lẻ).

E (hay e): định dạng kiểu số mũ

Ví dụ:

`Console.WriteLine("{0:E}", 250)` cho kết quả là 2.500000E+002

F (hay f): dạng số dấu chấm động

Ví dụ:

`Console.WriteLine("{0:F2}", 25)` cho kết quả là 25.00

N (hay n): định dạng số nói chung, có dấu ngăn cách hàng nghìn và dấu ngăn cách phần nguyên và phần lẻ (dấu phẩy, dấu chấm)

X (hay x): dạng thập lục phân

Ngoài ra Ký tự 0 (1 chữ số hay là chữ số 0 nếu không có chữ số tại vị trí tương ứng), ký tự # (1 chữ số hay là không có gì nếu không có chữ số tại vị trí tương ứng), dấu - (dấu ngắt khi viết viết số điện thoại), dấu phẩy và dấu chấm.

Ví dụ:

`Console.WriteLine("{0(0)(##)###.###.###}", 564567341)` cho kết quả là (0)(56)4-567-341

Ghi chú:

Phương thức `string.Format()` cũng tương tự như các hàm in, nhưng kết quả được đặt vào biến chuỗi thay vì in ra màn hình.

Ví dụ:

Một lệnh khai báo như sau:

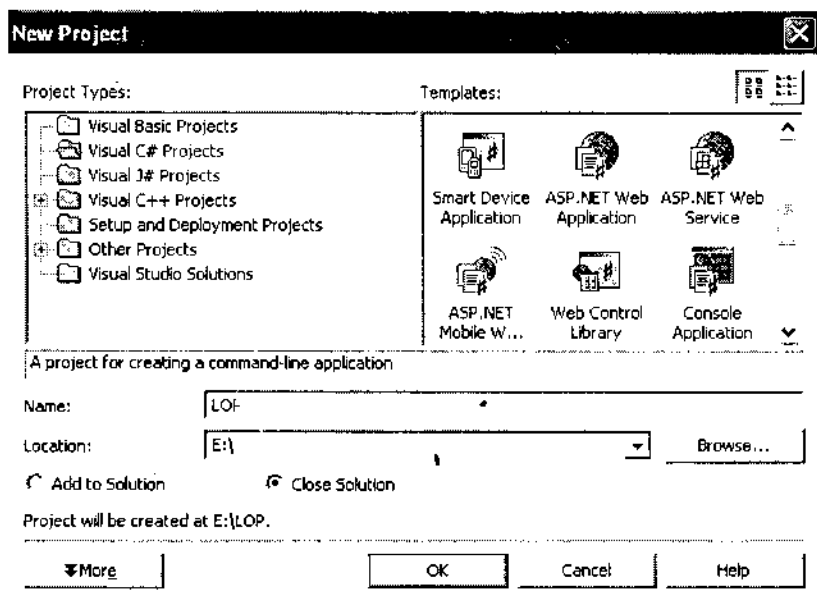
`String chuoi = string.Format("0x{0:X}", 27);`

Thì biến `chuoi` sẽ nhận giá trị là "0x1B".

Để thử chương trình trên, bạn thực hiện như sau:

✱ Vào Microsoft Visual Studio .NET

- * Click chuột vào menu **File**, chọn **New**, chọn **Project**
- * Trong cửa sổ **New Project** hiện ra, bạn chọn:
- * **Visual C# Project** trong khung bên trái có tên **Project Types**.
- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.
- * Đặt tên LOP trong khung **Name** như hình sau:



H61

- * Click nút **OK**

Vào cửa sổ viết code, xóa hết các dòng lệnh có sẵn trong chương trình. Viết lại các câu lệnh như sau:

```
using System;
class vongtron
{
    static double PI=3.1416;
    double R;
    public vongtron(double bk)
    {
```

```

        R=bk;
    }
    public double dientich()
    {
        return PI*R*R;
    }
    public double chuvi()
    {
        return 2*PI*R;
    }
};
class MainProgram
{
    static void Main()
    {
        vongtron vt=new vongtron(3);
        Console.WriteLine("Hinh tron :");
        Console.WriteLine(" DT={0},CV={1}",vt.dientich(), vt.chuvi());
        Console.ReadLine();
    }
}

```

Giải thích:

- Câu lệnh using System dùng cho phương thức Console
- Viết một lớp có tên vongtron, với các khai báo và các phương thức bên trong để tính diện tích và chu vi của hình tròn, chúng tôi đã giải thích ở phần trên.
- Viết một lớp chương trình chính để in ra diện tích và chu vi của hình tròn với bán kính R=3.
- Biên dịch và chạy thử chương trình, chúng ta sẽ thấy kết quả như hình sau:



```

E:\VLOP\bin\Debug\VLOP.exe
Hinh tron :
DT=28.2744,CV=18.8496

```

Thử lại bạn thấy:

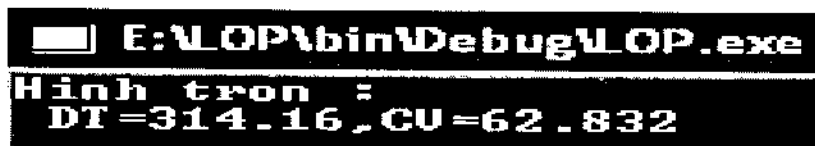
$$DT = \pi * R * R = 3.1416 * 3 * 3 = 28.2744$$

$$CV = 2 * \pi * R = 2 * 3.1416 * 3 = 18.8496$$

Bạn vào code của chương trình sửa số 3 trong cấp phát bộ nhớ cho phương thức vongtron(3) phần lớp chính của chương trình, thành số 10 (nghĩa là bán kính bằng 10). Bạn sẽ thấy đoạn code như sau:

```
using System;
class vongtron
{
    static double Pi=3.1416;
    double R;
    public vongtron(double bk)
    {
        R=bk;
    }
    public double dientich()
    {
        return Pi*R*R;
    }
    public double chuvi()
    {
        return 2*Pi*R;
    }
};
class MainProgram
{
    static void Main()
    {
        vongtron vt=new vongtron(10);
        Console.WriteLine("Hinh tron :");
        Console.WriteLine(" DT={0},CV={1}",vt.dientich(), vt.chuvi());
        Console.ReadLine();
    }
}
```

Chạy thử chương trình, bạn sẽ thấy kết quả như hình sau:



H63

Bạn tự kiểm tra lại kết quả khi bán kính $R=10$

Nhận xét:

■ Nếu viết chương trình như trên, muốn tính diện tích và chu vi của vòng tròn với giá trị bán kính là bao nhiêu thì chúng ta phải vào code của chương trình để sửa lại giá trị của bán kính, vậy rất là bất tiện, chúng ta sẽ viết sao để người sử dụng sẽ nhập giá trị của R khi chạy chương trình. Bạn viết thêm code như sau:

using System;

class vongtron

{

static double PI=3.1416;

double R;

public vongtron(double bk)

{

R=bk;

}

public double dientich()

{

return PI*R*R;

}

public double chuvi()

{

return 2*PI*R;

}

};

class MainProgram

{

static void Main()

{

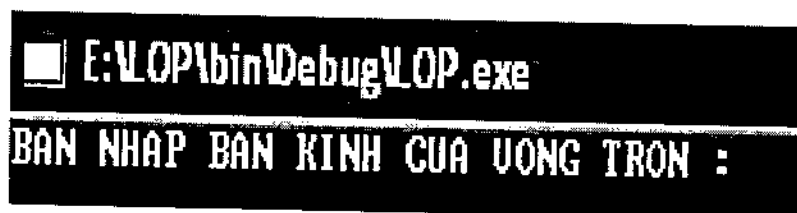

```

float bankinh;
Console.Write("BAN NHAP BAN KINH CUA VONG TRON :");
bankinh=float.Parse(Console.ReadLine());
vongtron vt=new vongtron(bankinh);
Console.WriteLine("Hinh tron :");
Console.WriteLine(" DT={0},CV={1}",vt.dientich(), vt.chuvi());
Console.ReadLine();
}
}

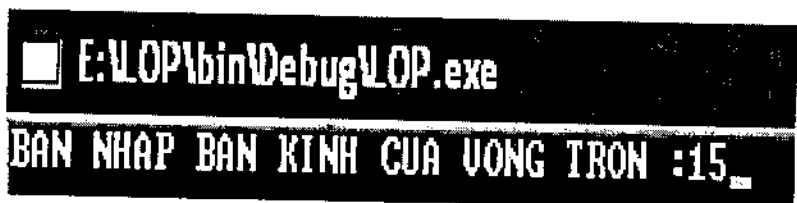
```

Giải thích:

- Bạn thêm vào 3 dòng lệnh đậm tại vị trí ở trên.
- Xóa số 10 trong phương thức vongtron(10), nhập vào biến bankinh thành vongtron(bankinh).
- Khai báo biến bankinh có kiểu số thực float.
- In ra màn hình câu hỏi "BAN NHAP BAN KINH CUA VONG TRON".
- Nhập vào giá trị của bán kính.
- Biên dịch và chạy thử chương trình, bạn sẽ có thấy như hình sau:



- Giả sử bạn nhập R=15 như hình sau:



Nhấn phím Enter, chúng ta có kết quả

E:\LOP\bin\Debug\LOP.exe

BAN NHAP BAN KINH CUA VONG TRON :15
Hinh tron :
DT=706.86,CU=94.248

H66

- Bạn tự kiểm tra kết quả.

Nếu bạn sửa một chút như sau:

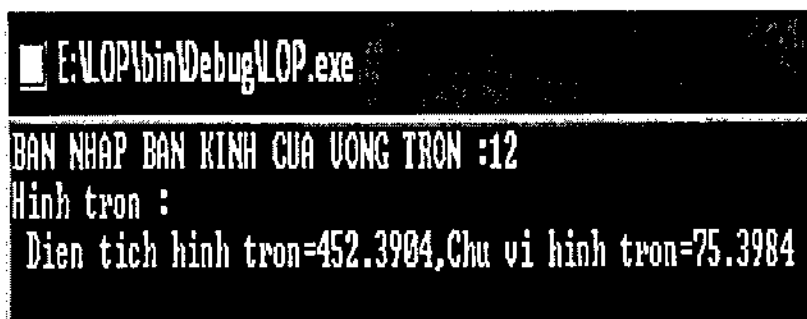
```
using System;
class vongtron
{
    static double PI=3.1416;
    double R;
    public vongtron(double bk)
    {
        R=bk;
    }
    public double dientich()
    {
        return PI*R*R;
    }
    public double chuvi()
    {
        return 2*PI*R;
    }
};
class MainProgram
{
    static void Main()
    {
        float bankinh;
        Console.WriteLine("BAN NHAP BAN KINH CUA VONG TRON :");
```

```

bankinh=float.Parse(Console.ReadLine());
vongtron vt=new vongtron(bankinh);
Console.WriteLine("Hình tron :");
Console.WriteLine(" Diện tích hình tron={0},Chu vi hình
tron={1}",vt.dientich(), vt.chuvi());
Console.ReadLine();
}
}

```

■ Thay đổi chuỗi thể hiện trong câu lệnh đậm ở trên, bạn sẽ thấy kết quả như hình sau, nếu bạn nhập giá trị bán kính là 12.



H67

Để hiểu rõ hơn về lớp, bạn xem ví dụ sau:

Viết một lớp thực hiện tính diện tích và chu vi của một hình chữ nhật.

Phân tích:

Trong hình chữ nhật sẽ có các thuộc tính chiều dài và chiều rộng, có các phương thức (hàm) để tính diện tích và chu vi qua các thuộc tính. Bạn sẽ viết một lớp để tính như sau:

```

class hinhchunhat
{
    double dai, rong;
    public hinhchunhat(double d, double r)
    {
        dai=d;
        rong=r;
    }
}

```

```

public double dientich()
{
    return dai*rong;
}
public double chuvi()
{
    return 2*(dai+rong);
}
};

```

Giải thích:

- Nếu đọc kỹ bài trên, bạn thấy bài này rất là dễ hiểu. Khai báo lớp bằng từ khóa class có tên là hìnhchunhat. Khai báo biến dai, rong có kiểu số thực double.

- Viết phương thức hìnhchunhat với hai đối có kiểu thực là d và r (trong vongtron thì chỉ có một đối đó là bán kính)

- Gán giá trị của hai đối cho các biến tương ứng.

- Viết phương thức tính diện tích

- Viết phương thức tính chu vi

- Để hoàn thành chương trình, bạn viết code như sau:

```

using System;
class hìnhchunhat
{
    double dai, rong;
    public hìnhchunhat(double d, double r)
    {
        dai=d;
        rong=r;
    }
    public double dientich()
    {
        return dai*rong;
    }
    public double chuvi()
    {

```

```

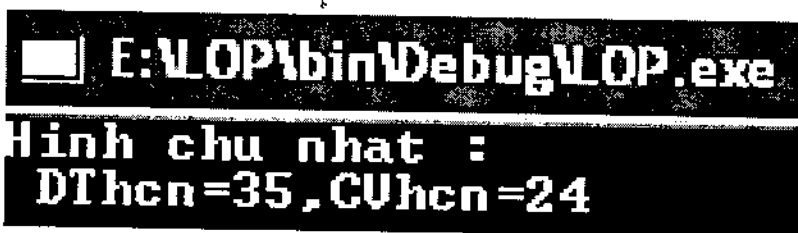
        return 2*(dai+rong);
    }
};
class MainProgram
{
    static void Main()
    {
        hinhchunhat hcn=new hinhchunhat(7,5);
        Console.WriteLine("Hình chu nhật :");
        Console.WriteLine(" DThcn={0},CVhcn={1}",hcn.dientich(),
                           hcn.chuvi());
        Console.ReadLine();
    }
}

```

Giải thích:

- Trong lớp chính chúng ta khai báo biến hcn có kiểu hinhchunhat và cấp phát bộ nhớ cho nó với chiều dài bằng 7cm và chiều rộng bằng 5cm. Hiện ra kết quả diện tích và chu vi của hình chữ nhật với chiều dài và rộng như trên.

- Biên dịch và chạy thử chương trình, bạn thấy kết quả như hình sau:



H68

Thử lại kết quả:

$DThcn = dai * rong = 7 * 5 = 35$

$CVhcn = 2 (dai + rong) = 2(7+5) = 24$

Bạn thấy kết quả trên không có đơn vị sau giá trị của diện tích và chu vi. Để xuất hiện, bạn định dạng trong code như sau:

```

using System;
class hinhchunhat

```

```

{
    double dai, rong;
    public hinhchunhat(double d, double r)
    {
        dai=d;
        rong=r;
    }
    public double dientich()
    {
        return dai*rong;
    }
    public double chuvi()
    {
        return 2*(dai+rong);
    }
};

class MainProgram
{
    static void Main()
    {
        hinhchunhat hcn=new hinhchunhat(7.5);
        Console.WriteLine("Hình chu nhật :");
        Console.WriteLine(" DThcn={0:#cm2}, CVhcn={1:#cm}",
                           hcn.dientich(), hcn.chuvi());
        Console.ReadLine();
    }
}

```

Giải thích:

- Bạn sửa lại chuỗi định dạng trong câu lệnh tô đậm ở trên. Dấu # thay thế cho số, cm^2 , cm điền đơn vị sau số.
- Biên dịch và chạy thử chương trình, chúng ta sẽ thấy kết quả như hình sau:

Hình chu nhật :
 $DT_{hcn}=35cm^2, CV_{hcn}=24cm$

H69

Để nhập vào giá trị của chiều dài và chiều rộng bất kỳ khi chạy chương trình, bạn thêm vào đoạn code như sau:

```
using System;
class hìnhchunhat
{
    double dai, rong;
    public hìnhchunhat(double d, double r)
    {
        dai=d;
        rong=r;
    }
    public double dientich()
    {
        return dai*rong;
    }
    public double chuvi()
    {
        return 2*(dai+rong);
    }
};
class MainProgram
{
    static void Main()
    {
        float chieudai, chieurong;
        Console.Write("BAN NHAP CHIEU DAI :");
        chieudai=float.Parse(Console.ReadLine());
```

```

Console.Write("BAN NHAP CHIEU RONG :");
chieurong=float.Parse(Console.ReadLine());
hinhchunhat hcn=new hinhchunhat(chieudai,chieurong);
Console.WriteLine("Hinh chu nhhat :");
Console.WriteLine(" DThcn={0:#cm2},CVhcn={1:#cm}",
                  hcn.dientich(), hcn.chuvi());
Console.ReadLine();
}
}

```

Giải thích:

- Bạn nhập và sửa lại các dòng lệnh đậm ở trên.
- Khai báo hai biến chieudai và chieurong có kiểu thực
- Nhập vào chiều dài, chiều rộng
- Thay hai đối của phương thức hinhchunhat là chieudai và chieurong, hinhchunhat(chieudai, chieurong)
- Xuất ra màn hình diện tích và chu vi của hình chữ nhật.
- Biên dịch và chạy thử chương trình, giả sử chúng ta nhập chiều dài là 10 và chiều rộng bằng 8, ta có kết quả như hình sau:



H70

- Chạy lại chương trình, giả sử chúng ta nhập sai (chiều dài có giá trị nhỏ hơn chiều rộng) như sau:
- Chiều dài bằng 10, chiều rộng bằng 15. Bạn sẽ thấy:



H71

Chương trình vẫn tính, vì trong phương thức `hinhchunhat` có hai đối là hai số thực, nhận được giá trị là nó sẽ thực hiện, không xét điều kiện chiều dài phải lớn hơn chiều rộng.

Để chương trình không thực hiện khi người sử dụng nhập chiều dài có giá trị nhỏ hơn chiều rộng, có nhiều cách, giả sử chúng ta thực hiện một cách là dùng lệnh `goto` như sau:

```
using System;
class hinhchunhat
{
    double dai, rong;
    public hinhchunhat(double d, double r)
    {
        dai=d;
        rong=r;
    }
    public double dientich()
    {
        return dai*rong;
    }
    public double chuvi()
    {
        return 2*(dai+rong);
    }
};
class MainProgram
{
    static void Main()
    {
        char hihi;
        float chieudai, chieurong;
        Console.Write("BAN NHAP CHIEU DAI :");
        chieudai=float.Parse(Console.ReadLine());
        Console.Write("BAN NHAP CHIEU RONG :");
        chieurong=float.Parse(Console.ReadLine());
        if (chieurong>=chieudai)
```

```

    goto hihi;
    hinhchunhat hcn=new hinhchunhat(chieudai,chieurong);
    Console.WriteLine("Hình chu nhật :");
    Console.WriteLine(" DThcn={0:#cm2},CVhcn={1:#cm}",
        hcn.dientich(), hcn.chuvi());

    hihi;
    if (chieurong>=chieudai)
        Console.WriteLine("BAN DA NHAP SAI, CHIEU DAI PHAI
            LON HON CHIEU RONG");

    Console.ReadLine();
}
}

```

Giải thích:

- Khai báo biến hihi có kiểu char
- Thêm phát biểu if với điều kiện khi chiều rộng lớn hơn hoặc bằng chiều dài thì sẽ thực hiện lệnh goto hihi để nhảy đến nhãn có tên hihi để bỏ qua các câu lệnh tính toán, và báo ra lỗi sai cho người sử dụng biết.

Qua hai bài tập trên, các bạn đã hiểu rõ khá nhiều về lớp. Bây giờ các bạn có thể gộp hai lớp trên vào chung một chương trình. Người sử dụng sẽ nhập bán kính của vòng tròn để tính diện tích và chu vi. Sau đó nhập tiếp chiều dài và chiều rộng để tính diện tích và chu vi của hình chữ nhật. Bạn viết code lại như sau:

```

using System;
class vongtron
{
    static double PI=3.1416;
    double R;
    public vongtron(double bk)
    {
        R=bk;
    }
    public double dientich()
    {
        return PI*R*R;
    }
}

```

```

    }
    public double chuvi()
    {
        return 2*PI*R;
    }
};
class hinhchunhat
{
    double dai, rong;
    public hinhchunhat(double d, double r)
    {
        dai=d;
        rong=r;
    }
    public double dientich()
    {
        return dai*rong;
    }
    public double chuvi()
    {
        return 2*(dai+rong);
    }
};
class MainProgram
{
    static void Main()
    {
        float bankinh;
        Console.Write("BAN NHAP BAN KINH CUA VONG TRON :");
        bankinh=float.Parse(Console.ReadLine());
        vongtron vt=new vongtron(bankinh);
        Console.WriteLine("Hinh tron :");
        Console.WriteLine(" Diên tích hình tron={0},Chu vi hình
                           tron={1}",vt.dientich(), vt.chuvi());
        Console.ReadLine();
    }
}

```

```

char hihi;
float chieudai, chieurong;
Console.Write("BAN NHAP CHIEU DAI :");
chieudai=float.Parse(Console.ReadLine());
Console.Write("BAN NHAP CHIEU RONG :");
chieurong=float.Parse(Console.ReadLine());
if (chieurong>=chieudai)
    goto hihi;
hinhchunhat hcn=new hinhchunhat(chieudai,chieurong);
Console.WriteLine("Hinh chu nhat :");
Console.WriteLine(" DThcn={0:#cm2},CVhcn={1:#cm}",
    hcn.dientich(), hcn.chuvi());
hihi;
if (chieurong<chieudai)
    Console.WriteLine("BAN DA NHAP SAI, CHIEU DAI PHAI
        LON HON CHIEU RONG");
Console.ReadLine();
}
}

```

Giải thích:

- Code của chương trình không có gì là thay đổi, chúng ta chỉ sắp xếp lại: trước tiên là lớp vongtron, sau đó là lớp hinhchunhat, sau đó là lớp chính cho chúng ta nhập vào giá trị bán kính và xuất hiện kết quả diện tích và chu vi hình tròn, sau đó nhập kết quả chiều dài, chiều rộng của hình chữ nhật, sẽ xuất diện tích và chu vi của hình chữ nhật.

- Biên dịch và chạy thử chương trình, bạn thấy tuần tự thực hiện của chương trình như sau:

 E:\LOP\bin\Debug\LOP.exe

BAN NHAP BAN KINH CUA VONG TRON :

■ Bạn nhập vào giá trị của bán kính. Giả sử nhập 7, gõ Enter ta được

```
E:\LOP\bin\Debug\LOP.exe
BAN NHAP BAN KINH CUA VONG TRON :7
Hinh tron :
Dien tích hình tron=153.9384,Chu vi hình tron=43.9824
```

H73

Ghi chú:

■ Nhờ có lệnh Console.ReadLine() phần này (lệnh đậm), chương trình sẽ dừng lại. Gõ phím Enter, bạn thấy hình như sau:

```
E:\LOP\bin\Debug\LOP.exe
BAN NHAP BAN KINH CUA VONG TRON :7
Hinh tron :
Dien tích hình tron=153.9384,Chu vi hình tron=43.9824
BAN NHAP CHIEU DAI :_
```

H74

Giả sử bạn nhập chiều dài là 10. gõ Enter

```
E:\LOP\bin\Debug\LOP.exe
BAN NHAP BAN KINH CUA VONG TRON :7
Hinh tron :
Dien tích hình tron=153.9384,Chu vi hình tron=43.9824
BAN NHAP CHIEU DAI :10
BAN NHAP CHIEU RONG :
```

H75

■ Giả sử bạn nhập chiều rộng là 8. Gõ Enter bạn có kết quả như sau:

```
■ E:\LOP\bin\Debug\LOP.exe
BAN NHAP BAN KINH CUA UONG TRON :7
Hinh tron :
  Dien tích hình tron=153.9384,Chu vi hình tron=43.9824

BAN NHAP CHIEU DAI :10
BAN NHAP CHIEU RONG :8
Hinh chu nhật :
  DThcn=80cm2, CVhcn=36cm
```

H76

Nếu nhập sai về kích thước của hình chữ nhật, tức chiều dài nhỏ hơn hoặc bằng chiều rộng, bạn sẽ thấy kết quả tính toán của hình chữ nhật không thực hiện và báo ra lỗi sai. Bạn tham khảo hai hình sau:

```
■ E:\LOP\bin\Debug\LOP.exe
BAN NHAP BAN KINH CUA UONG TRON :4
Hinh tron :
  Diện tích hình tron=50.2656,Chu vi hình tron=25.1328

BAN NHAP CHIEU DAI :4
BAN NHAP CHIEU RONG :4
BAN DA NHAP SAI, CHIEU DAI PHAI LON HON CHIEU RONG
```

H77 Khi chiều dài bằng chiều rộng

```
■ E:\LOP\bin\Debug\LOP.exe
BAN NHAP BAN KINH CUA UONG TRON :4
Hinh tron :
  Diện tích hình tron=50.2656,Chu vi hình tron=25.1328

BAN NHAP CHIEU DAI :6
BAN NHAP CHIEU RONG :9
BAN DA NHAP SAI, CHIEU DAI PHAI LON HON CHIEU RONG
```

H78 Khi chiều rộng lớn hơn chiều dài

Bạn thử làm thêm một bài tập nữa về lớp như sau:

Viết chương trình nhập vào hai số a và b. Tính $a+b$, $a-b$, $a*b$, a/b và $a\%b$ dùng lớp.

- * Vào Microsoft Visual Studio .NET

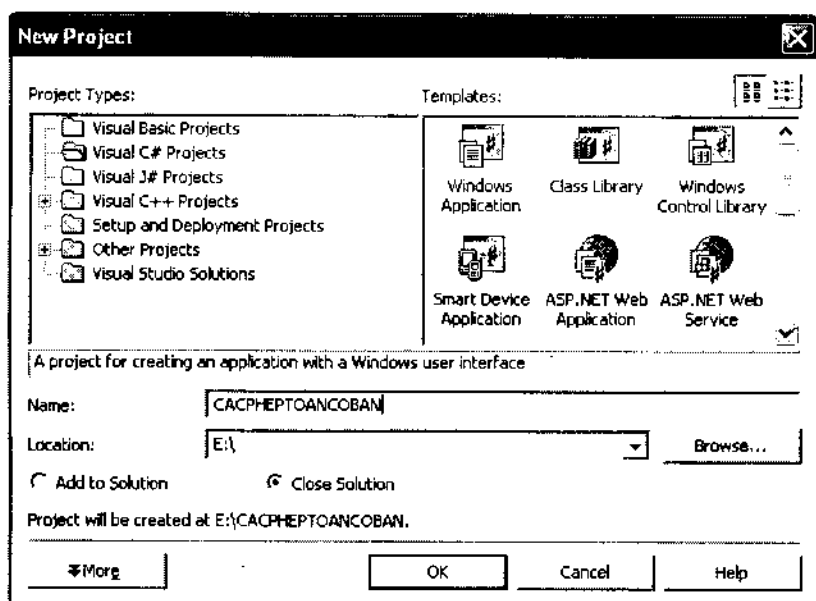
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**

- * Trong cửa sổ **New Project** hiện ra, bạn chọn:

- * **Visual C# Project** trong khung bên trái có tên **Project Types**.

- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.

- * Đặt tên **CACPHEPTOANCOBAN** trong khung **Name** như hình sau:



H79

- * Click nút OK

Vào cửa sổ viết code, xóa hết các dòng lệnh có sẵn, viết code như sau:

```
using System;
class tinhtoan
{
    double soa, sob;
```

```

public tinhtoan(double a, double b)
{
    soa=a;
    sob=b;
}
public double cong()
{
    return soa+sob;
}
public double tru()
{
    return soa-sob;
}
public double nhan()
{
    return soa*sob;
}
public double chia()
{
    return soa/sob;
}
public double sodu()
{
    return soa%sob;
}
};

```

```

class MainProgram
{
    static void Main()
    {
        double x, y;
        Console.Write("BAN NHAP SO a :");
        x=double.Parse(Console.ReadLine());
        Console.Write("BAN NHAP SO b :");
    }
}

```



```

y=double.Parse(Console.ReadLine());
tinhtoan tt=new tinhtoan(x,y);
Console.WriteLine("KET QUA TINH TOAN");
Console.WriteLine(" a+b={0},a-b={1},a*b={2}, a/b={3},a mod
    b={4} ",tt.cong(), tt.tru(), tt.nhan(), tt.chia(),tt.sodu());
Console.ReadLine();
}

```

Giải thích:

- Thiết lập một lớp có tên tinhtoan
- Để tính toán $a+b$, $a-b$, $a*b$, a/b và $a\%b$ ta thấy phải có hai số a và b thì mới tính được, vậy bạn phải khai báo hai biến này, ở đây chúng tôi báo hai biến có kiểu số thực.
- Khai báo một phương thức có tên trùng với tên của lớp đó là tinhtoan, có hai đối là số thực. Gán biến soa cho đối thứ nhất, sob cho đối thứ hai. Phương thức này là public để có thể gọi được ở ngoài lớp. Phương thức này dùng để khởi tạo dữ liệu (constructor).
- Khai báo phương thức cong có kiểu thực dạng **public** để có thể gọi ở ngoài lớp, với trị trả về là $a + b$.
- Khai báo phương thức tru có kiểu thực dạng public để có thể gọi ở ngoài lớp, với trị trả về là $a - b$.
- Khai báo phương thức nhan có kiểu thực dạng public để có thể gọi ở ngoài lớp, với trị trả về là $a * b$.
- Khai báo phương thức chia có kiểu thực dạng public để có thể gọi ở ngoài lớp, với trị trả về là a / b .
- Khai báo phương thức sodu có kiểu thực dạng public để có thể gọi ở ngoài lớp, với trị trả về là $a \% b$.
- Thiết lập một lớp chính của chương trình, khai báo hai biến x, y có kiểu thực.
- Nhập hai số thông qua hai biến này.
- Khai báo biến tt có kiểu tinhtoan và cấp phát bộ nhớ cho biến này với 2 đối là hai số vừa nhập.
- Xác lập các giá trị trong chuỗi của phương thức Write là: {0}, {1}, {2}, {3}, {4} tương ứng với các biến là các phương thức cong, tru, nhan, chia và sodu là con của lớp tinhtoan thông qua tên biến tt.

■ Biên dịch và chạy thử chương trình. Giả sử bạn nhập số a là 10, số b là 5. Gõ phím Enter, bạn thấy kết quả như hình sau:

```
E:\LOP\bin\Debug\LOP.exe
BAN NHAP SO a :10
BAN NHAP SO b :5
KET QUA TINH TOAN
a+b=15,a-b=5,a*b=50, a/b=2,a mod b=0
```

H80

Bạn chạy lại chương trình, nhập giá trị a và b bất kỳ thử kiểm tra kết quả xem có đúng hay không.

BÀI 10

Kiểu dữ liệu của người lập trình

Ngoài những kiểu dữ liệu đơn có sẵn, người lập trình có thể định nghĩa thêm kiểu dữ liệu mới cùng với các phép toán trên kiểu dữ liệu này. Chẳng hạn, để tạo kiểu dữ liệu tinhtuan cho hai số qua các phép toán cơ bản cộng, trừ, nhân, chia, tính số dư và in kết quả ra màn hình. Bạn có thể dùng kiểu cấu trúc struct

```
struct tenkieudulieu
```

```
{
```

```
.....
```

```
.....
```

```
.....
```

```
};
```

Bạn xem đoạn chương trình sau:

```
using System;
```

```
struct tinhtuan
```

```
{
```

```
double soa, sob;
```

```
public tinhtuan(double a, double b)
```

```
{
```

```
    soa=a;
```

```
    sob=b;
```

```
}
```

```
public double cong()
```

```
{
```

```
    return soa+sob;
```

```
}
```

```
public double tru()
```

```
{
```

```
    return soa-sob;
```

```
}
```

```

public double nhan()
{
    return soa*sob;
}
public double chia()
{
    return soa/sob;
}
public double sodu()
{
    return soa%sob;
}
};

class MainProgram
{
    static void Main()
    {
        double x, y;
        Console.Write("BAN NHAP SO a :");
        x=double.Parse(Console.ReadLine());
        Console.Write("BAN NHAP SO b :");
        y=double.Parse(Console.ReadLine());
        tinhtoan tt=new tinhtoan(x,y);
        Console.WriteLine("KET QUA TINH TOAN");
        Console.WriteLine(" a+b={0},a-b={1},a*b={2}, a/b={3},a mod  

            b={4} ",tt.cong(), tt.tru(), tt.nhan(), tt.chia(),tt.sodu());
        Console.ReadLine();
    }
}

```

■ Bạn thấy thay vì thiết lập một lớp với từ khóa class, bạn có thể dùng cấu trúc struct. Đối chiếu với chương trình tính toán các phép toán đơn giản dùng lớp ở bài trước, bạn thấy không có gì thay đổi nhiều, chỉ có một thay đổi nhỏ duy nhất đó là từ khóa class được thay thế bằng từ khóa struct.

■ Bạn vào lại chương trình cũ, sửa **class** thành **struct** và chạy thử chương trình, nhập vào hai số a và b. Giả sử chúng ta nhập a=34, b=3. Gõ phím Enter, bạn thấy kết quả như hình sau:

```
■ E:\LOP\bin\Debug\LOP.exe
BAN NHAP SO a :34
BAN NHAP SO b :3
KET QUẢ TÍNH TOÁN
a+b=37,a-b=31,a*b=102, a/b=11.33333333333333,a mod b=1
```

H81

■ Tương tự như vậy, bạn có thể vào lại project để tính diện tích và chu vi của vòng tròn và hình chữ nhật. Thay đổi từ khóa class của hai lớp vongtron và hinhchunhat thành từ khóa struct. Bạn thấy chương trình cũng thực hiện giống như khi dùng class. Vậy trong C# từ khóa class và từ khóa struct là tương đương với nhau (trừ một vài trường hợp đặc biệt, chúng tôi sẽ đề cập sau). Bạn xem lại đoạn code của chương trình trên.

```
using System;
struct vongtron
{
    static double PI=3.1416;
    double R;
    public vongtron(double bk)
    {
        R=bk;
    }
    public double dientich()
    {
        return PI*R*R;
    }
}
```

```

    public double chuvi()
    {
        return 2*PI*R;
    }
};

struct hinhchunhat
{
    double dai, rong;
    public hinhchunhat(double d, double r)
    {
        dai=d;
        rong=r;
    }
    public double dientich()
    {
        return dai*rong;
    }
    public double chuvi()
    {
        return 2*(dai+rong);
    }
};

class MainProgram
{
    static void Main()
    {
        float bankinh;
        Console.Write("BAN NHAP BAN KINH CUA VONG TRON :");
        bankinh=float.Parse(Console.ReadLine());
        vongtron vt=new vongtron(bankinh);
        Console.WriteLine("Hinh tron :");
        Console.WriteLine(" Diên tích hình tron={0},Chu vi hình
                           tron={1}",vt.dientich(), vt.chuvi());
        Console.ReadLine();
        char hihi;
    }
}

```

```

float chieudai, chieurong;
Console.Write("BAN NHAP CHIEU DAI :");
chieudai=float.Parse(Console.ReadLine());
Console.Write("BAN NHAP CHIEU RONG :");
chieurong=float.Parse(Console.ReadLine());
if (chieurong>=chieudai)
    goto hihi;
hinhchunhat hcn=new hinhchunhat(chieudai,chieurong);
Console.WriteLine("Hình chu nhật:");
Console.WriteLine(" DThcn={0:#cm2},CVhcn={1:#cm}",
    hcn.dientich(), hcn.chuvi());

hihi;
if (chieurong>=chieudai)
    Console.WriteLine("BAN DA NHAP SAI, CHIEU DAI PHAI
        LON HON CHIEU RONG");

Console.ReadLine();
}
}

```

■ Biên dịch và chạy thử chương trình, bạn nhập bán kính R, bạn sẽ thấy kết quả diện tích và chu vi của vòng tròn ứng với giá trị của bán kính vừa nhập. Gõ phím Enter, nhập chiều dài, chiều rộng của hình chữ nhật, bạn sẽ thấy kết quả diện tích và chu vi của hình chữ nhật ứng với các giá trị vừa nhập.

■ Bạn thử xem kết quả khi chúng tôi nhập R=20, chiều dài = 15 và chiều rộng = 10.

■ E:\LOP\bin\Debug\LOP.exe

BAN NHAP BAN KINH CUA VONG TRON :20

Hình tron :

Diện tích hình tron=1256.64,Chu vi hình tron=125.664

BAN NHAP CHIEU DAI :15

BAN NHAP CHIEU RONG :10

Hình chu nhật :

DThcn=150cm²,CVhcn=50cm

BÀI 11

CÁC PHƯƠNG THỨC TRÙNG TÊN (Chống phương thức)

Một ưu điểm của các ngôn ngữ lập trình hướng đối tượng là việc cho phép các phương thức có trùng tên và phân biệt nhau nhờ vào số lượng các tham số, kiểu của tham số và giá trị trả về của phương thức. Khái niệm này được gọi là **method overloading** (chống phương thức). Cơ chế này cho phép khai thác các phương thức một cách linh động hơn. Bạn xem lại chương trình dưới đây, chúng tôi lấy lại bài tạo lớp tinhtoan để tính các phép toán cơ bản, và thay đổi lại một chút.

```
using System;
class tinhtoan
{
    double soa, sob, soc;
    public tinhtoan(double a, double b)
    {
        soa=a;
        sob=b;
    }
    public tinhtoan(double a, double b, double c)
    {
        soa=a;
        sob=b;
        soc=c;
    }
    public double conghai()
    {
        return soa+sob;
    }
}
```



```

    public double truhai()
    {
        return soa-sob;
    }
    public double nhanhai()
    {
        return soa*sob;
    }
    public double chiahai()
    {
        return soa/sob;
    }
    public double sodu()
    {
        return soa%sob;
    }
    public double congba()
    {
        return soa+sob+soc;
    }
    public double truba()
    {
        return soa-sob-soc;
    }
    public double nhanba()
    {
        return soa*sob*soc;
    }
    public double chiaba()
    {
        return soa/sob/soc;
    }
};

```

```

class MainProgram

```

```

{
    static void Main()
    {
        double x, y, z;
        Console.Write("BAN NHAP SO a :");
        x=double.Parse(Console.ReadLine());
        Console.Write("BAN NHAP SO.b :");
        y=double.Parse(Console.ReadLine());
        Console.Write("BAN NHAP SO c :");
        z=double.Parse(Console.ReadLine());
        tinhtoan tt2=new tinhtoan(x,y);
        Console.WriteLine("KET QUA TINH TOAN TREN HAI SO a,b");
        Console.WriteLine(" a+b={0},a-b={1},a*b={2}, a/b={3},a mod
                           b={4} ",tt2.conghai(), tt2.truhai(), tt2.nhanhai(),
                           tt2.chiahai(),tt2.sodu());
        tinhtoan tt3=new tinhtoan(x,y,z);
        Console.WriteLine("KET QUA TINH TOAN TREN BA SO a,b,c");
        Console.WriteLine(" a+b+c={0},a-b-c={1},a*b*c={2},
                           a/b/c={3}",tt3.congba(), tt3.truba(), tt3.nhanba(),
                           tt3.chiaba());
        Console.ReadLine();
    }
}

```

Giải thích:

■ Bài này cũng giống như bài tập dùng lớp ở trên. Một cái đặc biệt là bạn thấy chúng tôi khai báo hai phương thức có cùng tên tinhtoan, nhưng hai phương thức này có số tham số khác nhau, phương thức tinhtoan thứ nhất có hai tham số, phương thức tinhtoan thứ hai có ba tham số. Điều này cho phép được trùng tên với nhau.

- Thiết lập một lớp có tên là tinhtoan
- Khai báo ba biến có kiểu số thực (double)
- Thiết lập phương thức có tên tinhtoan có hai đối, đây là phương thức khởi tạo dữ liệu
- Thiết lập phương thức có tên tinhtoan có ba đối, đây là phương thức khởi tạo dữ liệu

- Thiết lập các phương thức conghai, truhai, nhanhai, chiahai, sodu để thực hiện các phép toán tương ứng trên hai số.
- Thiết lập các phương thức congba, truba, nhanba, chiaaba để thực hiện các phép toán tương ứng trên ba số.
- Tạo một lớp chính để thể hiện việc tính toán ra màn hình.
- Khai báo ba biến là số thực để nhận ba số nhập vào
- Khai báo biến tt2 có kiểu tinhtoan có hai đối và cấp phát bộ nhớ cho nó.
- Xuất kết quả các phép toán ra màn hình
- Khai báo biến tt3 có kiểu tinhtoan có ba đối và cấp phát bộ nhớ cho nó.
- Xuất kết quả tính toán ra màn hình.
- Biên dịch và chạy thử chương trình, giả sử nhập $a=12$, $b=4$, $c=2$.
Gõ phím Enter, bạn thấy kết quả như hình sau:

 E:\LOP\bin\Debug\LOP.exe

BAN NHAP SO a :12

BAN NHAP SO b :4

BAN NHAP SO c :2

KET QUA TINH TOAN TREN HAI SO a,b

$a+b=16$, $a-b=8$, $a*b=48$, $a/b=3$, $a \bmod b=0$

KET QUA TINH TOAN TREN BA SO a,b,c

$a+b+c=18$, $a-b-c=6$, $a*b*c=96$, $a/b/c=1.5$

H83

BÀI 12

TRUYỀN THAM SỐ

Một cải tiến đáng chú ý của C# so với các ngôn ngữ lập trình khác là việc truyền tham số của các phương thức được quy định rất chặt chẽ, tạo điều kiện để giảm bớt các lỗi lập trình xảy ra do việc truyền tham số. Tuy nhiên, người lập trình cũng cần phải nắm vững các quy định này thì mới vận dụng có hiệu quả.

Tham số giá trị

Giá trị của tham số được truyền cho hàm gọi, các thao tác trên tham số hình thức sẽ không ảnh hưởng đối với biến truyền đến.

Ví dụ:

Bạn xem đoạn chương trình sau:

```
using System;
struct sophuc
{
    public double thuc, ao;
    public sophuc(double a, double b)
    {
        thuc=a;
        ao=b;
    }
};
class MainProgram
{
    static void tang(int i)
    {
        i++;
    }
}
```

```

static void giam(int j)
{
    j--;
}
static void tang(sophuc c)
{
    c.thuc++;
}
static void giam(sophuc d)
{
    d.thuc--;
}
static void addString(string chuoi)
{
    chuoi=chuoi+" are you";
}
static void Main()
{
    int n=25;
    sophuc sp =new sophuc(4,9);
    string ch="How";
    tang(n);
    Console.WriteLine("1/ gia tri cua n la: 0",n);
    giam(n);
    Console.WriteLine("2/ gia tri cua n la: 0",n);
    tang(sp);
    Console.WriteLine("3/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
    giam(sp);
    Console.WriteLine("4/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
    addString(ch);
    Console.WriteLine("5/ {0}",ch);
    Console.ReadLine();
}
};

```

Giải thích:

- Khai báo một cấu trúc có tên sophuc.
 - Khai báo hai biến thuc và ao có kiểu số thực double, dạng public để có thể gọi khi ở ngoài cấu trúc.
 - Thiết lập một phương thức khởi tạo có tên sophuc (cùng tên với cấu trúc). Phương thức này có hai đối số a, b kiểu số thực.
 - Khởi tạo biến thuc và biến ao.
 - Tạo một hàm tang có một đối số i kiểu số nguyên
 - Tăng biến i lên 1
 - Tạo một hàm giam có một đối số j kiểu số nguyên
 - Giảm biến j xuống 1
 - Tạo một hàm tang có một đối số c kiểu số phức
 - Tăng biến thực của số phức lên 1
 - Tạo một hàm giam có một đối số d kiểu số phức
 - Giảm biến thực của số phức xuống 1
 - Tạo một hàm addString có một đối số chuỗi kiểu string
 - Gán chuỗi bằng chuỗi + " are you"
 - Vào hàm chính
 - Khai báo biến n có kiểu int và gán giá trị 25 cho nó
 - Khai báo biến sp có kiểu số phức có giá trị thực là 4, ảo là 9 và cấp phát bộ nhớ cho nó
 - Khai báo biến ch có kiểu string và gán chuỗi "How" cho nó
 - Gọi hàm tang với đối số là n (n=25)
 - In giá trị của biến n sau khi thực hiện hàm tang
 - Gọi hàm giam với đối số là n (n=25)
 - In giá trị của biến n sau khi thực hiện hàm giam
 - Gọi hàm tang với đối số là sp
 - In giá trị của số phức sau khi thực hiện hàm tang
 - Gọi hàm giam với đối số là sp
 - In giá trị của số phức sau khi thực hiện hàm giam
 - Gọi hàm nối chuỗi addString
 - In ra chuỗi sau khi thực hiện hàm nối chuỗi addString
- Biên dịch và chạy thử chương trình, bạn thấy kết quả như hình sau:

```
E:\TRUYEN_THAM_SO\bin\Debug\TRUYEN_THAM_SO.exe
```

```
1/ gia tri cua n la: 25
2/ gia tri cua n la: 25
3/ So phuc: 4+9i
4/ So phuc: 4+9i
5/ How
```

H84

Bạn thấy kết quả vẫn giữ nguyên giá trị cũ. $n=25$, thành phần thực của số phức là 4, ảo là 9 (đây là các giá trị được gán trong chương trình khi khai báo biến).

Trường hợp đặc biệt

Khi tham số là **đối tượng** thì tình huống xảy ra không giống như kiểu **struct** bởi vì các đối tượng của **C#** là kiểu dữ liệu **tham chiếu**.

Vẫn lấy ví dụ trên nhưng thay từ khóa **struct** bằng từ khóa **class** như sau (từ tô đậm):

```
using System;
class sophuc
{
    public double thuc, ao;
    public sophuc(double a, double b)
    {
        thuc=a;
        ao=b;
    }
};
class MainProgram
{
    static void tang(int i)
    {
        i++;
    }
    static void giam(int j)
    {
        j--;
    }
}
```

```

static void tang(sophuc c)
{
    c.thuc++;
}
static void giam(sophuc d)
{
    d.thuc--;
}
static void addString(string chuoi)
{
    chuoi=chuoi+" are you";
}
static void Main()
{
    int n=25;
    sophuc sp =new sophuc(4,9);
    string ch="How";
    tang(n);
    Console.WriteLine("1/ gia tri cua n la: {0}",n);
    giam(n);
    Console.WriteLine("2/ gia tri cua n la: {0}",n);
    tang(sp);
    Console.WriteLine("3/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
    giam(sp);
    Console.WriteLine("4/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
    addString(ch);
    Console.WriteLine("5/ {0}",ch);
    Console.ReadLine();
}
};

```

Chạy thử chương trình, bạn thấy kết quả như hình sau:


```
1/ gia tri cua n la: 25
2/ gia tri cua n la: 25
3/ So phuc: 5+9i
4/ So phuc: 4+9i
5/ How
```

H85

Hàm tang và giam viết cho kiểu sophuc trong (class) và (struct) là giống nhau, nhưng kết quả in ra cho trường hợp này là phần thực của số phức tăng lên 1 đơn vị (từ 4 lên 5) khi thực hiện hàm tang(sp). Sau đó sẽ giảm 1 đơn vị khi gặp hàm giam(sp) (từ 5 xuống 4). Số n và biến chuỗi không phải là đối tượng nên không thay đổi.

Nếu bạn sửa lại code của chương trình từ giam(sp) thành tang(sp) như sau:

```
using System;
class sophuc
{
    public double thuc, ao;
    public sophuc(double a, double b)
    {
        thuc=a;
        ao=b;
    }
};
class MainProgram
{
    static void tang(int i)
    {
        i++;
    }
    static void giam(int j)
    {
        j--;
    }
    static void tang(sophuc c)
    {
        c.thuc++;
    }
}
```

```

}
static void giam(sophuc d)
{
    d.thuc--;
}
static void addString(string chuoai)
{
    chuoai=chuoai+" are you";
}
static void Main()
{
    int n=25;
    sophuc sp =new sophuc(4,9);
    string ch="How";
    tang(n);
    Console.WriteLine("1/ gia tri cua n la: {0}",n);
    giam(n);
    Console.WriteLine("2/ gia tri cua n la: {0}",n);
    tang(sp);
    Console.WriteLine("3/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
    tang(sp);
    Console.WriteLine("4/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
    addString(ch);
    Console.WriteLine("5/ {0}",ch);
    Console.ReadLine();
}
};

```

Chạy thử chương trình, bạn sẽ thấy kết quả như sau:

```

E:\TRUYEN_THAM_SO\bin\Debug\TRUYEN_THAM_SO.exe
1/ gia tri cua n la: 25
2/ gia tri cua n la: 25
3/ So phuc: 5+9i
4/ So phuc: 6+9i
5/ How

```

Hàm tang(sp) thứ nhất tăng phần thực của số phức lên 1 (4 lên 5), hàm tang(sp) thứ hai sẽ tăng phần thực lên 1 đơn vị nữa (5 lên 6).

Bạn xem thêm một ví dụ nữa qua đoạn code sau:

```
using System;
struct phanso
{
    public int tuso, mauso;
    public phanso(int tu, int mau)
    {
        tuso=tu;
        mauso=mau;
    }
};
class MainProgram
{
    static void tang(phanso ps)
    {
        ps.tuso+=ps.mauso;
    }
    static void Main()
    {
        phanso huuti=new phanso(3,11);
        tang(huuti);
        Console.WriteLine("{0}/{1}",huuti.tuso,huuti.mauso);
        Console.ReadLine();
    }
};
```

Giải thích:

- Khai báo cấu trúc struct có tên phanso
- Khai báo hai biến tuso và mauso có kiểu nguyên int dạng public
- Thiết lập phương thức phanso có hai đối là số nguyên (phương thức khởi tạo dữ liệu)
- Khai báo hàm tang có một đối ps có kiểu phanso để tăng phần số (phần số + 1)

■ Khai báo biến huuti có kiểu phanso có tử số là 3, mẫu số là 11 và cấp phát bộ nhớ cho nó.

- Thực hiện hàm tang(huuti)
- In ra màn hình phân số
- Chạy thử chương trình, bạn thấy kết quả như sau:



H87

Kết quả in ra trên màn hình phân số 3/11 (giống như giá trị khai báo)

Sửa lại đoạn code trên như sau:

```
using System;
class phanso
{
    public int tuso, mauso;
    public phanso(int tu, int mau)
    {
        tuso=tu;
        mauso=mau;
    }
};
class MainProgram
{
    static void tang(phanso ps)
    {
        ps.tuso+=ps.mauso;
    }
    static void Main()
    {
        phanso huuti=new phanso(3,11);
        tang(huuti);
        Console.WriteLine("{0}/{1}",huuti.tuso,huuti.mauso);
    }
}
```

```
Console.ReadLine();
```

- Chạy lại chương trình, bạn thấy kết quả như sau:



```
E:\TRUYEN_THAM_SO\bin\Debug\TRUYEN_THAM_SO.exe
```

```
14/11
```

H88

Kết quả in ra là 14/11 tức là bằng phân số khai báo (3//11) cộng với 1 ($3/11 + 11/11 = 14/11$)

Bạn suy nghĩ kết quả của hai trường hợp trên.

Tham số tham chiếu

Các biến đưa tới khi gọi hàm sẽ thay đổi nếu có các thao tác trên tham số hình thức tương ứng.

```
using System;
```

```
struct sophuc
```

```
{  
    public double thuc, ao;  
    public sophuc(double a, double b)  
    {  
        thuc=a;  
        ao=b;  
    }  
};
```

```
class MainProgram  
{
```

```
    static void tang(ref int i)
```

```
    {  
        i++;  
    }
```

```
    static void giam(ref int j)
```

```
    {
```

```

        j--;
    }
    static void tang(ref sophuc c)
    {
        c.thuc++;
    }
    static void giam(ref sophuc d)
    {
        d.thuc--;
    }
    static void addString(ref string chuoai)
    {
        chuoai=chuoai+" are you";
    }
    static void Main()
    {
        int n=25;
        sophuc sp =new sophuc(4,9);
        string ch="How";
        tang(ref n);
        Console.WriteLine("1/ gia tri cua n la: {0}",n);
        giam(ref n);
        Console.WriteLine("2/ gia tri cua n la: {0}",n);
        tang(ref sp);
        Console.WriteLine("3/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
        giam(ref sp);
        Console.WriteLine("4/ So phuc: {0}+{1}i",sp.thuc, sp.ao);
        addString(ref ch);
        Console.WriteLine("5/ {0}",ch);
        Console.ReadLine();
    }
};

```

Giải thích:

■ Đoạn code này giống như chương trình trên dùng từ khóa struct chỉ thay đổi bằng cách thêm từ khóa ref trước các tham số.

- Các tham số tham chiếu được khai báo tường minh bằng từ khóa `ref` kể cả lúc viết hàm và gọi hàm.

- Biên dịch và chạy thử chương trình, chúng ta có kết quả như hình sau:

```
E:\TRUYEN_THAM_SO\bin\Debug\TRUYEN_THAM_SO.exe
1/ gia tri cua n la: 26
2/ gia tri cua n la: 25
3/ So phuc: 5+9i
4/ So phuc: 4+9i
5/ How are you
```

H89

Bây giờ thì giá trị các biến đưa đến bị thay đổi so với trước khi gọi hàm.

Tham số kết quả

C# hỗ trợ thêm kiểu truyền tham số kết quả để lưu kết quả tính toán sau khi gọi hàm. Tham số này được khai báo với từ khóa `out` có cú pháp lập trình hoàn toàn giống như từ khóa `ref` (lúc cài đặt và gọi hàm), ý nghĩa gần giống tham số tham chiếu nhưng cần chú ý các đặc điểm khác biệt như sau:

Không cần khởi tạo biến sẽ được gửi đến tham số kết quả của hàm (ngược lại đối với tham số giá trị và tham số tham chiếu thì C# bắt buộc phải khởi tạo các biến gửi đến)

Bên trong mỗi phương thức phải có ít nhất một lệnh gán giá trị cho mỗi tham số kết quả của hàm.

BÀI 13

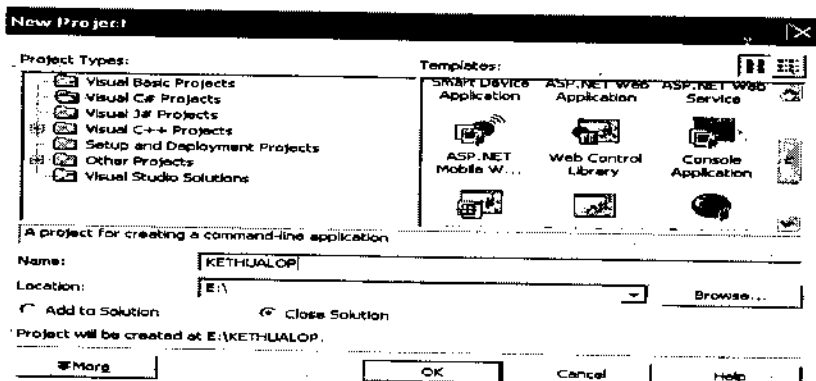
KẾ THỪ LỚP ĐỐI TƯỢNG

Một trong những đặc điểm nổi bật của ngôn ngữ lập trình hướng đối tượng là việc kế thừa mã nguồn đã có từ các lớp có sẵn. Ý tưởng kế thừa được dựa trên quan hệ rất tự nhiên của thế giới thực.

Ví dụ:

Bởi vì hình vuông là trường hợp đặc biệt của hình chữ nhật nên nó cũng có đầy đủ các tính chất của hình chữ nhật. Nên ta có thể tạo một lớp mới kế thừa lớp hình chữ nhật đã có. Khi lớp B kế thừa từ lớp A, ta nói A là lớp cơ sở (cũng gọi là lớp cha) còn B là lớp kế thừa (cũng gọi là lớp dẫn xuất hay là lớp con). Bạn thiết kế chương trình như sau:

- * Vào Microsoft Visual Studio .NET
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**
- * Trong cửa sổ **New Project** hiện ra, bạn chọn:
- * **Visual C# Project** trong khung bên trái có tên **Project Types**.
- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.
- * Đặt tên **KETHUALOP** trong khung **Name** như hình sau:



* Click nút OK

Xóa hết code có sẵn trong khung, nhập vào đoạn code như sau:

```
using System;
class hcn
{
    protected float dai, rong;
    public hcn(float d, float r)
    {
        dai=d;
        rong=r;
    }
    public float dientich()
    {
        return dai*rong;
    }
    public float chuvi()
    {
        return 2*(dai+rong);
    }
};
class hv:hcn
{
    public hv(float kt):base(kt,kt)
    {
    }
};
class MainProgram
{
    static void Main()
    {
        hcn cn=new hcn(8,6);
        hv v=new hv(8);
        Console.WriteLine("- Diện tích hình chu nhật  
= {0}",cn.dientich());
    }
}
```

```

Console.WriteLine("- Chu vi hình chữ nhật ={0}",cn.chuvi());
Console.WriteLine("- Diện tích hình vuông ={0}",v.dientich());
Console.WriteLine("- chu vi hình vuông ={0}",v.chuvi());
Console.ReadLine();
}
}

```

Giải thích:

- Thiết lập lớp **hcn** (hình chữ nhật) bao gồm một constructor và hai phương thức tính diện tích và chu vi.
 - Khai báo hai biến dài và rộng có kiểu số thực float. Từ khóa **protected** (lớp **hcn**) là để che dấu hai trường dài và rộng trong lớp **hcn** nhưng vẫn cho truy xuất được trong lớp kế thừa **hv** (hình vuông) (khác với trường **private** là không được truy xuất ngay cả đối với lớp kế thừa).
 - Các kiểu truy xuất đối với thuộc tính và phương thức của lớp:
 - ◆ Khai báo **private**: được hiển thị mặc nhiên nếu không quy định từ khóa này. Chỉ dùng trong phạm vi lớp, không thể truy xuất từ bên ngoài hay từ lớp kế thừa.
 - ◆ Khai báo **protected**: được truy xuất từ lớp kế thừa
 - ◆ Khai báo **public**: không giới hạn phạm vi, được truy xuất từ bên ngoài lớp.
 - Thiết lập lớp **hv** (hình vuông) thừa kế lớp **hcn** (hình chữ nhật), bạn thấy khai báo **class hv:hcn**
- class tên lớp con : tên lớp cha hoặc**
class tên lớp cơ sở : tên lớp dẫn xuất
- Từ khóa **base** trong câu lệnh **public hv(float kt):base(kt,kt)** dùng để gọi lại constructor của lớp cơ sở trong quan hệ kế thừa, tức là constructor của lớp **hcn**.
 - Trong đoạn chương trình chính khai báo biến **cn** có kiểu **hcn** có hai đối dài =8 và rộng = 6 đồng thời cấp bộ nhớ cho nó.
 - Khai báo biến **v** có kiểu **hv** với đối là 8.
 - In ra màn hình diện tích hình chữ nhật.
 - In ra màn hình chu vi hình chữ nhật.
 - In ra màn hình diện tích hình vuông, sử dụng phương thức **dientich** của **hcn** (kế thừa)

- In ra màn hình chu vi hình vuông, sử dụng phương thức chuvi của hcn (kế thừa)
- Biên dịch và chạy thử chương trình, bạn thấy kết quả như hình sau:

 E:\KETHUA\bin\Debug\KETHUA.exe

```
- Diện tích hình chữ nhật =48
- Chu vi hình chữ nhật =28
- Diện tích hình vuông =64
- chu vi hình vuông =32
```

H91

Nếu bạn thích nhập độ dài các cạnh của hình chữ nhật, hình vuông bằng một số bất kỳ, bạn sửa lại code của chương trình như sau:

```
using System;
class hcn
{
    protected float dai, rong;
    public hcn(float d, float r)
    {
        dai=d;
        rong=r;
    }
    public float dientich()
    {
        return dai*rong;
    }
    public float chuvi()
    {
```

```

        return 2*(dai+rong);
    }
};
class hv:hcn
{
    public hv(float kt):base(kt,kt)
    {
    }
};
class MainProgram
{
    static void Main()
    {
        float dai, rong, canh;
        Console.Write("Ban nhap chieu dai cua hinh chu nhac: ");
        dai=float.Parse(Console.ReadLine());
        Console.Write("Ban nhap chieu rong cua hinh chu nhac: ");
        rong=float.Parse(Console.ReadLine());
        Console.Write("Ban nhap canh cua hinh vuong: ");
        canh=float.Parse(Console.ReadLine());
        hcn cn=new hcn(dai,rong);
        hv v=new hv(canh);
        Console.WriteLine("- Dien tich hinh chu nhac
                           ={0}",cn.dientich());
        Console.WriteLine("- Chu vi hinh chu nhac ={0}",cn.chuvi());
        Console.WriteLine("- Dien tich hinh vuong ={0}",v.dientich());
        Console.WriteLine("- chu vi hinh vuong ={0}",v.chuvi());
        Console.ReadLine();
    }
}

```

Giải thích:

- Bạn thêm các dòng lệnh đậm ở phía trên. Đầu tiên là khai báo các biến dai, rong, canh có kiểu số thực float.
- Nhập giá trị chiều dài của hình chữ nhật
- Nhập giá trị chiều rộng của hình chữ nhật

- Nhập giá trị cạnh hình vuông
- Cấp phát bộ nhớ cho biến cn có kiểu hcn và đối là hai biến dai và rong vừa nhập
- Cấp phát bộ nhớ cho biến v có kiểu hv và đối là biến canh vừa nhập.
- Các lệnh khác đã giải thích ở phần trên
- Biên dịch và chạy thử chương trình, giả sử nhập chiều dài = 12, chiều rộng = 8, cạnh hình vuông = 15. Ta được kết quả như hình sau:

 E:\KETHUA\bin\Debug\KETHUA.exe

```
Ban nhap chieu dai cua hinh chu nhat: 12
Ban nhap chieu rong cua hinh chu nhat: 8
Ban nhap canh cua hinh vuong: 15
- Dien tich hinh chu nhat =96
- Chu vi hinh chu nhat =40
- Dien tich hinh vuong =225
- chu vi hinh vuong =60
```

H92

Bạn tự định dạng để có đơn vị của từng lời giải, nếu quên, xem lại các bài tập ở trên.

BÀI 14

TƯỚNG ỨNG BỘI (POLYMORPHISM)

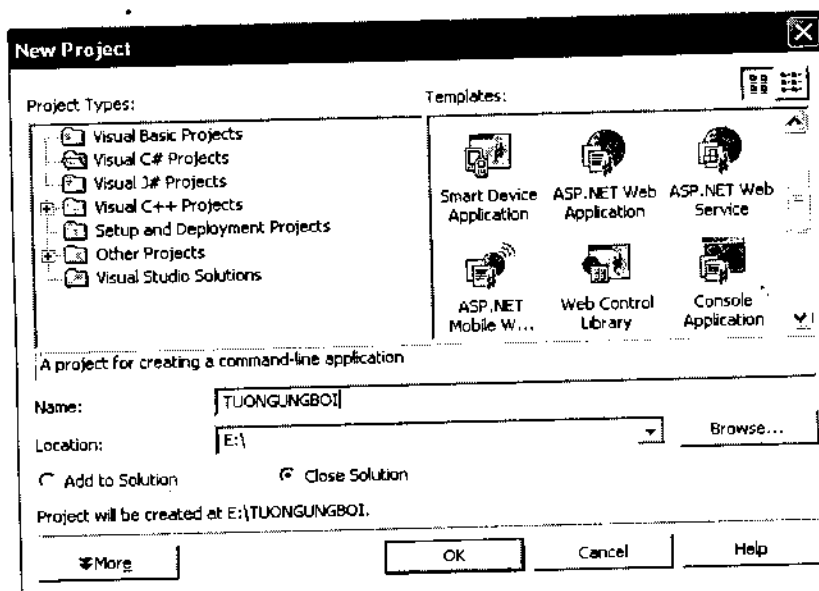
Đây là cơ chế đặc sắc của các ngôn ngữ lập trình hướng đối tượng. Khi có nhiều đối tượng khác kiểu mà có phương thức trùng tên, cơ chế này cho phép một biến tham chiếu đến đối tượng gọi đúng phương thức cài đặt cho lớp của chính đối tượng đó.

Bạn xem ví dụ sau:

- * Vào Microsoft Visual Studio .NET
- * Click chuột vào menu **File**, chọn **New**, chọn **Project**
- * Trong cửa sổ **New Project** hiện ra, bạn chọn:
- * **Visual C# Project** trong khung bên trái có tên **Project Types**.
- * Di chuyển con chạy bên khung bên phải, tìm chọn **Console Application**.

Application.

- * Đặt tên **TUONGUNGBOI** trong khung **Name** như hình sau:



* Click nút OK

Chúng tôi viết code như sau:

```
using System;
class chung
{
    public virtual float dientich()
    {
        return 0;
    }
};
class vongtron:chung
{
    float pi=3.1416F;
    float bk;
    public vongtron (float r)
    {
        bk=r;
    }
    public override float dientich()
    {
        return pi*bk*bk;
    }
};
class hinhchunhat:chung
{
    float dai, rong;
    public hinhchunhat(float d, float r)
    {
        dai=d;
        rong=r;
    }
    public override float dientich()
    {
        return dai*rong;
    }
};
```

```

    }
};
class MainProgram
{
    static void Main()
    {
        chung goi;
        vongtron vt=new vongtron(5);
        hinhchunhat hcn=new hinhchunhat(4,3);
        goi = vt;
        Console.WriteLine(goi.dientich());
        goi = hcn;
        Console.WriteLine(goi.dientich());
        Console.ReadLine();
    }
}

```

Giải thích:

- Lớp có tên chung khai báo phương thức `dientich()` với từ khóa `virtual` (phương thức ảo). Hai phương thức `vongtron` và `hinhchunhat` kế thừa từ lớp chung.

- Cài đặt cụ thể phương thức `dientich()` để tính diện tích hình tròn và hình chữ nhật. Nhờ khai báo `virtual` nên có cơ chế tương ứng bội.

- Trong hàm `Main()`, khai báo biến `goi` có kiểu chung, khai báo biến `vt` có kiểu `vongtron` có bán kính bằng 5 và cấp phát bộ nhớ cho nó, khai báo biến `hcn` có kiểu `hinhchunhat` có hai đối chiều dài bằng 4 và chiều rộng bằng 3.

- Khi biến `goi` trỏ đến `vt` (kiểu `vongtron`) thì lệnh `goi.dientich()` sẽ gọi phương thức `dientich()` của lớp `vongtron`, nếu biến `goi` trỏ đến `hcn` (kiểu `hinhchunhat`) thì cũng là lệnh `goi.dientich()` nhưng lại gọi đến phương thức của lớp `hinhchunhat`.

- Biên dịch và chạy thử chương trình, bạn thấy kết quả như hình sau:

78.54

12

H94

- Kiểm tra lại ta thấy:

Diện tích hình tròn: $3.1416 * 25 = 78.54$

Diện tích hình chữ nhật: $4 * 3 = 12$

Bây giờ nếu muốn tính thêm chu vi hình tròn và chu vi hình chữ nhật. Bạn phải thêm phương thức chuvi với từ khóa virtual, cài đặt thêm hai phương thức tính chu vi để tính chu vi hình tròn và chu vi hình chữ nhật. Bạn xem đoạn code sửa lại như sau:

```
using System;
class chung
{
    public virtual float dientich()
    {
        return 0;
    }
    public virtual float chuvi()
    {
        return 0;
    }
};
class vongtron:chung
{
    float pi=3.1416F;
    float bk;
    public vongtron (float r)
    {
        bk=r;
    }
}
```

```

    public override float dientich()
    {
        return pi*bk*bk;
    }
    public override float chuvi()
    {
        return 2*pi*bk;
    }
};
class hinhchunhat:chung
{
    float dai, rong;
    public hinhchunhat(float d, float r)
    {
        dai=d;
        rong=r;
    }
    public override float dientich()
    {
        return dai*rong;
    }
    public override float chuvi()
    {
        return 2*(dai+rong);
    }
};
class MainProgram
{
    static void Main()
    {
        chung goi;
        vongtron vt=new vongtron(5);
        hinhchunhat hcn=new hinhchunhat(4,3);
        goi = vt;
        Console.WriteLine(goi.dientich());
    }
}

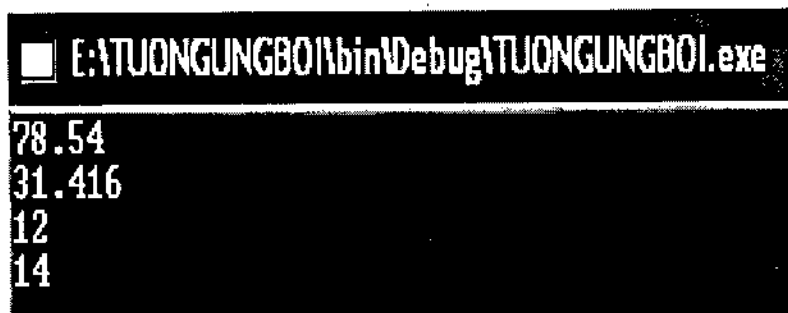
```

```

Console.WriteLine(goi.chuvi());
goi = hcn;
Console.WriteLine(goi.dientich());
Console.WriteLine(goi.chuvi());
Console.ReadLine();
}
}

```

- Biên dịch và chạy thử chương trình, bạn có kết quả như hình sau:



H95

Bạn tính và thấy kết quả chính xác phải không. Bạn thử thay đổi code để có thể nhập vào bán kính, chiều dài, chiều rộng và khi kết quả in ra sẽ có đơn vị nằm ở phía sau (giả sử đơn vị là cm). Bạn viết code như sau:

```

using System;
class chung
{
    public virtual float dientich()
    {
        return 0;
    }
    public virtual float chuvi()
    {
        return 0;
    }
};
class vongtron:chung

```

```

{
    float pi=3.1416F;
    float bk;
    public vongtron (float r)
    {
        bk=r;
    }
    public override float dientich()
    {
        return pi*bk*bk;
    }
    public override float chuvi()
    {
        return 2*pi*bk;
    }
};
class hinhchunhat:chung
{
    float dai, rong;
    public hinhchunhat(float d, float r)
    {
        dai=d;
        rong=r;
    }
    public override float dientich()
    {
        return dai*rong;
    }
    public override float chuvi()
    {
        return 2*(dai+rong);
    }
};
class MainProgram
{

```

```

static void Main()
{
    chung goi;
    float bankinh, cd, cr;
    Console.Write("Ban nhap vao gia tri cua ban kinh: ");
    bankinh=float.Parse(Console.ReadLine());
    Console.Write("Ban nhap vao chieu dai cua hinh chu nhac: ");
    cd=float.Parse(Console.ReadLine());
    Console.Write("Ban nhap vao chieu rong cua hinh chu nhac: ");
    cr=float.Parse(Console.ReadLine());
    vongtron vt=new vongtron(bankinh);
    hinhchunhat hcn=new hinhchunhat(cd,cr);
    goi = vt;
    Console.WriteLine("Dien tich vong tron ={0}cm2",goi.dientich());
    Console.WriteLine("Chu vi cua vong tron ={0}cm",goi.chuvi());
    goi = hcn;
    Console.WriteLine("Dien tich hinh chu nhac
                        ={0}cm2",goi.dientich());
    Console.WriteLine("Chu vi hinh chu nhac ={0}cm",goi.chuvi());
    Console.ReadLine();
}
}

```

Giải thích:

■ Phần này chúng tôi đã giải thích ở trên, nếu quên các bạn có thể xem lại ở những bài trước.

Phần 3

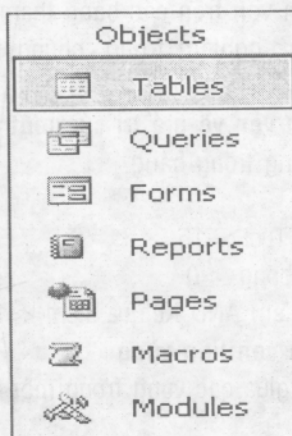
LẬP TRÌNH VISUAL C# DÙNG CƠ SỞ DỮ LIỆU ACCESS 2003

VÀI NÉT CƠ BẢN VỀ CSDL ACCESS

I/ Cách tạo DataBase (CSDL), Table (bảng) và mối liên hệ giữa các Table trong Access

Trước khi bắt tay vào viết các chương trình về quản lý, đầu tiên các bạn phải biết về **cơ sở dữ liệu**. Vậy cơ sở dữ liệu là gì, bạn cứ hiểu một cách nôm na nó là **một kho chứa những thông tin liên quan đến một chủ đề hay một mục đích quản lý nào đó**. Có nhiều loại cơ sở dữ liệu, trong cuốn sách này, chúng tôi chỉ đề cập đến **cơ sở dữ liệu quan hệ**. Đây là cơ sở dữ liệu phổ biến nhất hiện nay.

Trong Access 2003, một DataBase gồm có **7 loại vật dụng** đó là **các Table (bảng)**, **các Query (bảng truy vấn)**, **các Form (biểu mẫu)**, **các Report (báo biểu)**, **các Macro (những lệnh ngầm)**, các Papes và những Module (đơn thể lập trình).



H96

Trong 7 loại vật dụng trên, chúng ta sẽ phải luôn luôn **dùng các table**. Vì tất cả các dữ liệu sẽ được chứa trong các table. Các table này là **tập hợp các record** hay còn gọi là **các mẫu tin**, mỗi **record** gồm có **nhiều cột** gọi là **các field** hoặc còn gọi là các **trường** hay là những **vùng**.

Một cơ sở dữ liệu quan hệ cho phép ta lấy về các tập hợp dữ liệu con từ các table bằng các câu lệnh **SQL** mà chúng tôi sẽ giới thiệu cho các bạn trong phần kế tiếp.

Trong cơ sở dữ liệu cho phép ta liên kết các table với nhau nhằm mục đích cho ta truy cập các mẫu tin liên quan chứa trong các table khác nhau. Việc liên kết phải được thiết lập các ràng buộc toàn vẹn.

◆ **Ràng buộc toàn vẹn**

Ràng buộc toàn vẹn là điều kiện ràng buộc trên dữ liệu trong các bảng tham chiếu của CSDL và tất cả các ràng buộc toàn vẹn phải được thoả mãn ở bất kỳ thời điểm nào. Việc kiểm tra các ràng buộc toàn vẹn thường được tiến hành sau các thao tác cập nhật dữ liệu (thêm, sửa, xoá) và có thể được thông qua các thủ tục kiểm tra do ta thiết kế, hoặc do Visual Basic sẽ tự động kiểm tra các ràng buộc toàn vẹn dựa trên những xác lập của chúng ta khi thiết kế CSDL.

Thông thường người ta thường ràng buộc toàn vẹn trên vùng và trên mẫu tin. Trong đó ràng buộc vùng sẽ được kiểm tra trước các ràng buộc trên mẫu tin.

Việc ràng buộc toàn vẹn trên các bảng tham chiếu nếu thiếu sót sẽ gây ra những hiệu quả nghiêm trọng, chúng ta thử nghiên cứu các ràng buộc toàn vẹn.

■ **Ràng buộc toàn vẹn về giá trị của một vùng**

Chỉ dựa vào một vùng trong bảng.

Ví dụ:

Vandongvien.Tuoi >=15

Sinhvien.Hocbang >=0

0<= Ketqua.diem AND Ketqua.diem <=10

■ **Ràng buộc toàn vẹn liên vùng**

Mô tả mối quan hệ giữa các vùng trong một bảng.

Ví dụ:

Dondathang.Ngaydathang <= Dondathang.Ngayxuathang

Thanh tien = Soluong * Dongia

Ghi chú:

Chúng ta nên hạn chế việc một vùng được tính toán từ các vùng khác.

■ **Ràng buộc toàn vẹn giữa các mẫu tin trong một bảng**

Loại ví dụ này cũng thường thấy trong thực tế, ví dụ một vận động viên đến đăng ký thi thể thao phải ghi đầy đủ 5 chi tiết.

Một dạng đặc biệt của ràng buộc toàn vẹn loại này là các ràng buộc về khóa gồm các khái niệm: *siêu khóa, khóa nội và khóa chính*

⇒ **Siêu khóa (super key) – Khóa chính (primary key)**

Trong một bảng, các mẫu tin có thể phân biệt với nhau về lưu trữ thông qua số thứ tự của mẫu tin. Tuy nhiên, trong quá trình khai thác một bảng, sự phân biệt này không đóng vai trò quan trọng, nhất là trong trường hợp hai mẫu tin có số thứ tự khác nhau vẫn có thể giống nhau hoàn toàn về dữ liệu. Do đó cần thiết phải có một khái niệm khác có thể dùng để phân biệt về mặt dữ liệu giữa các mẫu tin trong một bảng.

Một siêu khóa SK của bảng B là một tập hợp gồm một số vùng (hoặc toàn bộ vùng) của bảng B mà hai mẫu tin bất kỳ trong bảng B sẽ không lấy cùng giá trị trên các vùng này. Nói cách khác, một siêu khóa của một bảng dùng để phân biệt hai bộ bất kỳ trong bảng.

Ví dụ:

(MSVDV, Hoten, Nuoc) là một siêu khóa của bảng Vandongvien. Với giả thiết một bảng B không chứa những mẫu tin hoàn toàn trùng nhau, ta nói bảng B luôn có ít nhất một siêu khóa chứa tất cả các vùng của bảng B và có thể có nhiều siêu khóa.

Ví dụ:

(MSVDV, Hoten, Nuoc), (MSVDV, MaMT), ... là những siêu khóa của bảng Vandongvien.

Vậy siêu khóa dùng để nhận diện các mẫu tin trong bảng, nghĩa là, khi cần truy tìm một mẫu tin x nào đó, ta chỉ cần biết các giá trị của các vùng siêu khóa của x là đủ để dò tìm. Vậy nếu một bảng có nhiều siêu khóa, ta nên chọn một siêu khóa nhỏ nhất (chứa ít vùng nhất), để

giảm thời gian dò tìm. Một siêu khóa nhỏ nhất được gọi là một khóa hay khóa nội của một bảng.

Ví dụ: MSVDV là khóa nội của bảng Vandongvien.

Đối với các thực thể tồn tại khách quan như sinh viên, học sinh, nhân viên, vận động viên, hàng hoá ... ta hay dùng mã số để làm khóa, đối với khóa nội của các bảng biểu diễn cho trừu tượng hóa (kết quả thi ...) thường là một tập hợp của hai hay nhiều vùng của bảng. Khi có nhiều khóa nội, ta chọn một trong các khóa nội để điều khiển chính cho việc truy cập các mẫu tin. Khóa nội này được gọi là khóa chính. Khóa chính chỉ có ý nghĩa khi truy tìm, không có vai trò gì khác so với các khóa nội còn lại.

Khi nhập mẫu tin mới có giá trị khóa chính trùng với khóa chính của một vận động viên đã có, VB sẽ báo lỗi.

■ Ràng buộc toàn vẹn về khóa ngoại

Ràng buộc toàn vẹn về khóa ngoại còn được gọi là ràng buộc về phụ thuộc tồn tại. Cũng giống như ràng buộc toàn vẹn về khóa, ràng buộc khóa ngoại rất phổ biến trong CSDL.

Ta xem ví dụ một sinh viên V đang đang theo học khoa Tin học. Mẫu tin tương ứng với sinh viên này trong bảng Sinhvien sẽ có giá trị vùng makh là "TINHOC". Sự tồn tại của mẫu tin này trong bảng Sinhvien hoàn toàn phụ thuộc vào sự tồn tại của một mẫu tin trong bảng Khoa có giá trị vùng makhoa là "TINHOC". Nếu trong bảng Khoa không có mẫu tin nào có giá trị vùng makhoa là "TINHOC" thì mẫu tin của sinh viên V trong bảng Sinhvien không thể tồn tại. Điều đó có nghĩa là khoa Tin học phải có trước khi sinh viên V đăng ký theo học. Trong trường hợp này, ta nhận thấy rằng khóa makhoa của bảng chính(Khoa) là có mối liên kết với vùng makh của bảng phụ thuộc(Sinhvien). Vùng makh còn được gọi là khóa ngoại.

■ Ràng buộc toàn vẹn giữa các mẫu tin của nhiều bảng

Một ví dụ của loại ràng buộc toàn vẹn này: mỗi chi tiết đặt hàng hoặc không được đáp ứng hoặc được giao tối đa một lần.

■ Ràng buộc toàn vẹn giữa các vùng của nhiều bảng

Ví dụ: Ngày giao hàng cho một đơn đặt hàng phải sau ngày đặt hàng.

■ Ràng buộc toàn vẹn phức tạp trong nhiều bảng

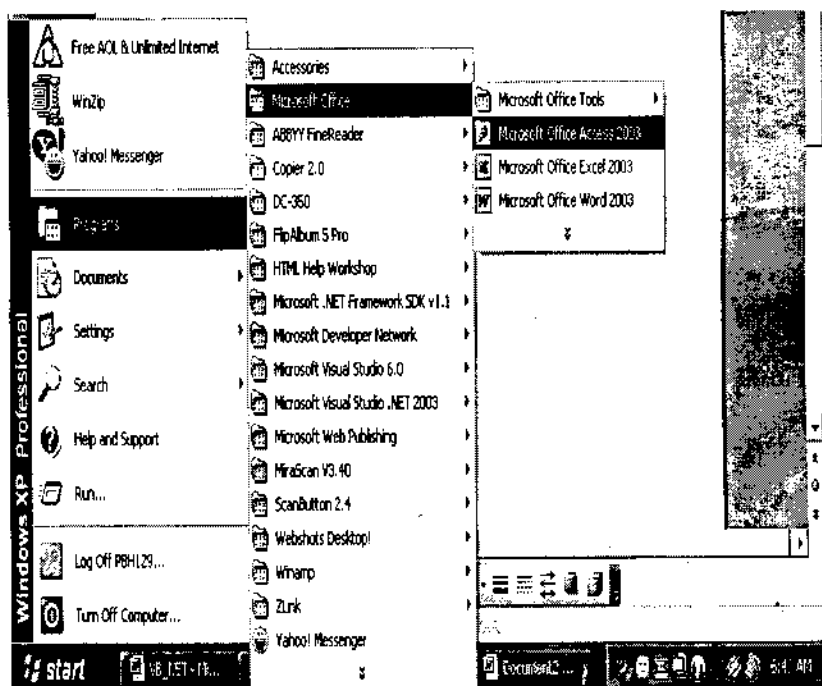
Ví dụ: Chỉ giao những mặt hàng mà khách đã đặt với số lượng không vượt quá số lượng đặt hàng.

Trong tập sách này, chúng tôi dùng **cơ sở dữ liệu Access**. Vậy trước tiên chúng tôi sẽ giới thiệu đến các bạn về **cách tạo DataBase trong Access**.

II- Cách tạo DataBase trong Access 2003

Muốn tạo **DataBase** trong **Access**, bạn sẽ tiến hành các bước như sau:

■ Từ menu **Start** chọn **Program**, chọn **Microsoft Office**, chọn **Microsoft Access 2003**.



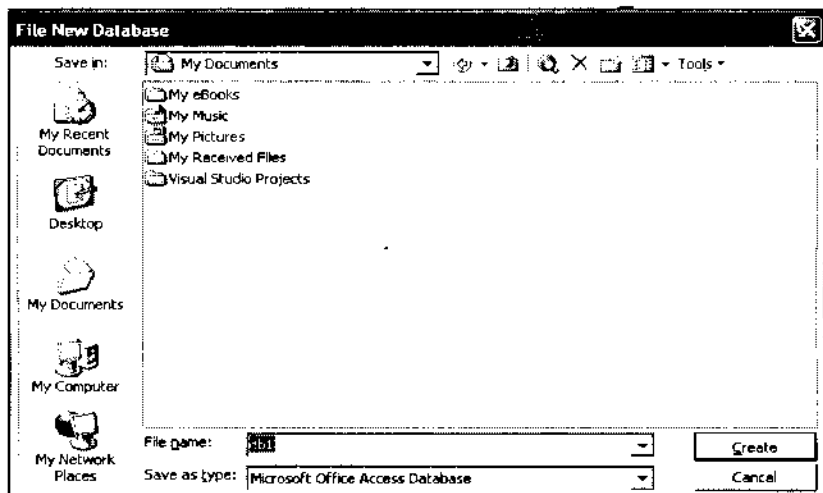
H97

Chọn xong, cửa sổ **Microsoft Access** hiện ra, các bạn chọn một trong những cách sau:

☐ **Create new file**

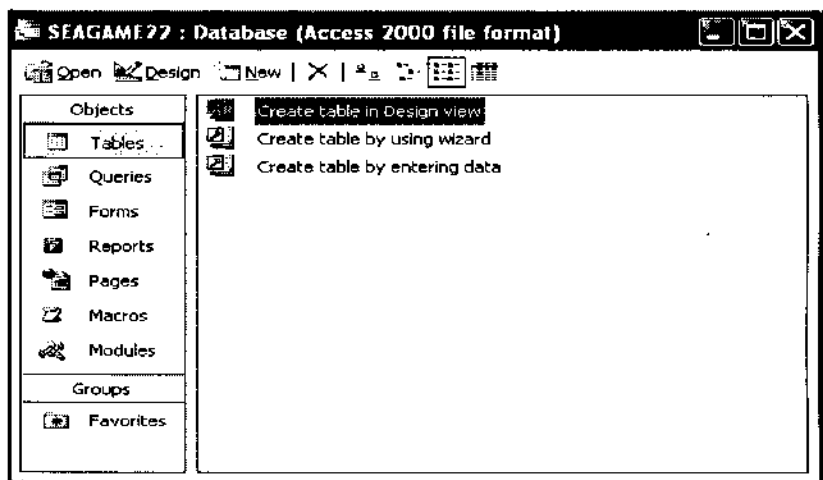
☐ Hoặc vào menu **File**, chọn **New**

☐ Hoặc click vào icon có hình tờ giấy trắng trên thanh công cụ.



H100

Trong hộp Textbox có tên **Save in**, bạn chọn ổ đĩa và thư mục cần chứa DataBase. Trong textBox có tên **file name**, bạn gõ tên của DataBase mà bạn cần tạo, giả sử bạn gõ vào SEAGAMES23 sau đó click chuột vào nút **Create**, bạn sẽ thấy cửa sổ hiện ra có tên là **SEAGAMES23 DataBase** như ở bên dưới và DataBase này sẽ có phần mở rộng là **.MDB**.



H101 Cửa sổ của DataBase mà bạn vừa tạo

Nhìn vào cửa sổ bên trái, bạn thấy có 7 tab ghi 7 loại vật dụng mà chúng tôi đã đề cập ở trên. Hiện thời tab **tables** đang được chọn, bởi vì chúng mới tạo nên chưa có table nào cả, **bạn nhớ phải có table thì mới tạo được các query, form và các report được.**

Vì chúng ta dùng tiếng Việt, nên trước khi tạo các table bạn vào menu **Tools**, chọn **Options**, hộp thoại Option xuất hiện.

■ Chọn tab **DataSheet**

■ Trong mục **Default Font**, bạn chọn kiểu tiếng Việt, giả sử bạn chọn **VNI-Helve-Condense** và cỡ chữ là **12** chẳng hạn. Bạn sẽ thấy hộp thoại có dạng như sau:

Options [?] [X]

Advanced | International | Error Checking | Spelling | Tables/Queries
View | General | Edit/Find | Keyboard | Datasheet | Forms/Reports | Pages

Default colors

Font: [Black] v
Background: [White] v
Gridlines: [Silver] v

Default gridlines showing

☒ Horizontal
☒ Vertical
Default column width: [1"]

Default font

Font: [VNI-Helve-Condense] v
Weight: [Normal] v Size: [12] v
☐ Underline ☐ Italic

Default cell effect

☒ Flat
☐ Raised
☐ Sunken
☒ Show animations
☒ Show Smart Tags on Datasheets

[OK] [Cancel] [Apply]

H102 Hộp thoại Option cho bạn chọn font

■ Chọn **Apply** rồi **OK**.

Vậy là bạn đã báo cho máy biết các **table** trong **DataBase** này sẽ dùng **Font VNI-Helve-Condense** và có **size** là **12**.

Tiếp theo chúng ta sẽ tạo các table (bảng). Như đã nói thành phần cơ bản của cơ sở dữ liệu trong Access là các table, đây là đối tượng quan trọng nhất, dùng để ghi nhận và chứa các dữ liệu cơ sở, các hiện tượng phát sinh, các biến động...

III- Cách tạo một Table trong Access 2003

Để tạo một table, đầu tiên bạn phải phác họa **cấu trúc** của nó trước ngoài nháp, trong table có các **field** gì, các field này có **kiểu** là gì, **chiều dài** (kích thước) của nó, **field** nào sẽ là **field khóa**.

Ý nghĩa của **field Khóa** trong một table là với **bất kỳ hai mẫu tin trong table** thì **field khóa của hai mẫu tin này sẽ luôn luôn có giá trị khác nhau**. Bạn có thể hình dung như là số chứng minh nhân dân của mỗi người công dân Việt Nam đến tuổi được cấp thì sẽ khác nhau. Trong Access khi bạn đã chọn field nào đó là field khóa, khi bạn vô tình **nhập giá trị trùng với giá trị đã có**, thì máy sẽ báo là **bạn đã vi phạm và đề nghị bạn nhập lại**. Field khóa này sẽ giúp bạn tìm kiếm một nhân vật hoặc một mặt hàng một cách dễ dàng và chính xác khi bạn cần tìm.

Khi khai báo cấu trúc của một table nó gồm có ba cột khai báo là **Field Name**, **Data type** và **Description**:

Cột Field Name: Dùng để khai báo tên của vùng này, tên vùng bạn nên chọn ngắn gọn, tên gợi nhớ và không nên dùng dấu.

Cột Data Type: Cột này dùng để mô tả kiểu của vùng, nó có thể là Text, Number, Date/Time, Memo, Currency, Yes/No... Khi khai báo kiểu, bạn có thể khai báo chiều dài của nó ở ô Textbox bên dưới, có tên là Field Size, nếu không sửa lại thì bạn sẽ chọn giá trị mặc nhiên của nó.

Cột Description: Cột này dùng để mô tả, ghi chú lại tên của vùng đó. Các bạn có thể để trống cột này nếu không thích ghi chú gì cả.

Giả sử chúng tôi có hai table VANDONGVIEN và GIAITHUONG, cấu trúc của hai table này như sau:

Table VANDONGVIEN

Field Name	Data Type	Description
MSVDV	Text (5)	Mã số vận động viên
Hoten	Text (40)	Họ tên
Nam	Yes/No	Nam/Nữ
Tuoi	Integer	Tuổi
Nuoc	Text (30)	Nước
Monthi	Text(20)	Môn thi

Table GIAITHUONG

Field Name	Data Type	Description
MSGT	Text (5)	Mã số giải thưởng
Diem	Integer	Điểm
Huychuong	Text (10)	Huy chương
VDVMSVDV	Text(5)	Mã số vận động viên

Ở hai table trên, trong cột **Field Type** bạn thấy chúng tôi khai báo tên các vùng là tên gọi nhớ, được viết liền nhau và không dùng dấu.

Trong cột **Data Type** chúng tôi khai báo kiểu tương ứng của các vùng, bên cạnh chúng tôi có chú thích kích thước của chúng, trong thực tế, các bạn sẽ khai báo trong hàng bên dưới ở dòng có tên là **Field Size**. Nếu không thay đổi nó sẽ lấy giá trị mặc nhiên.

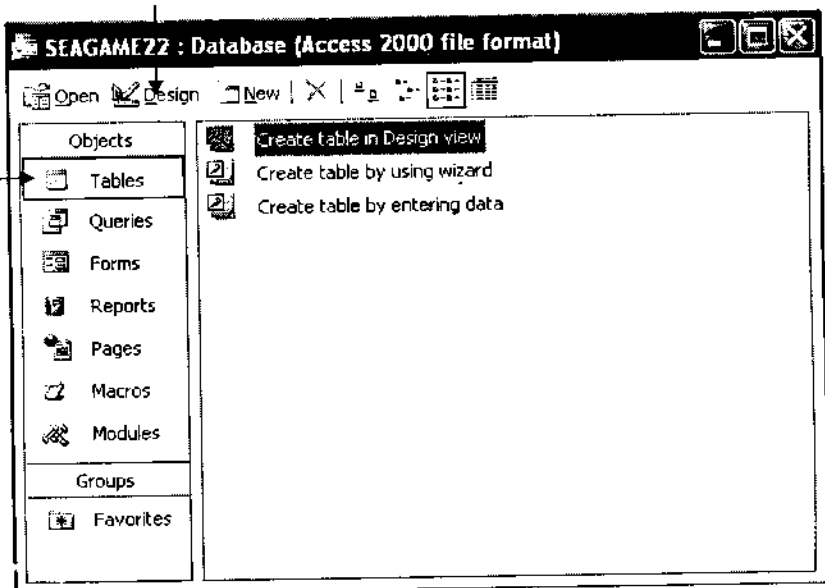
Trong cột **Description** chúng tôi ghi chú tên gọi của vùng, nếu không ghi cũng chẳng sao.

Trong bảng VANDONGVIEN khóa chính là **MSVDV**, trong bảng GIAITHUONG, khóa chính là **STT** và khóa ngoại là **VDVMSVDV**. Chúng tôi sẽ đề cập đến khóa ngoại sau.

Có nhiều cách tạo bảng trong Access, ở đây chúng tôi giới thiệu với bạn cách thông dụng mà chúng tôi hay dùng đó là **Design**.

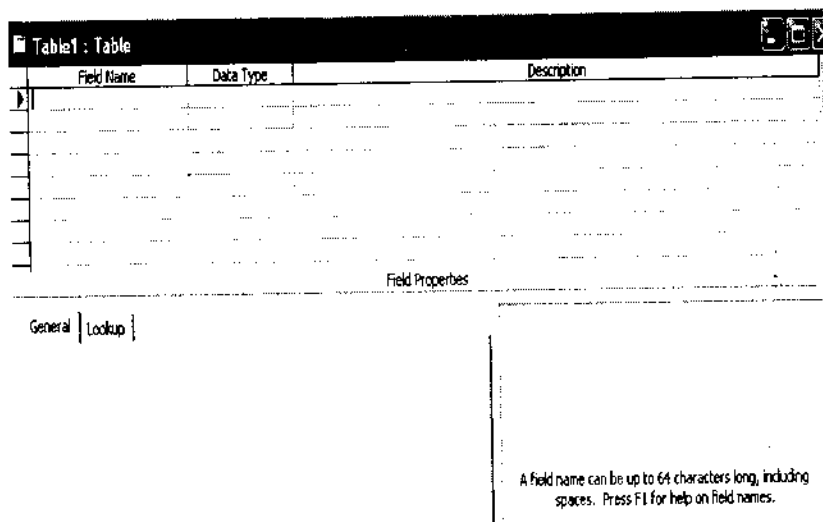
Nếu cửa sổ DataBase mà bạn cần tạo các table chưa có trên màn hình, thì trước tiên bạn phải mở cửa sổ DataBase bằng cách, vào Access, click chọn tên DataBase cần mở, click nút OK.

Từ cửa sổ DataBase này, bạn chọn ngăn **tables**, chọn **Design**.



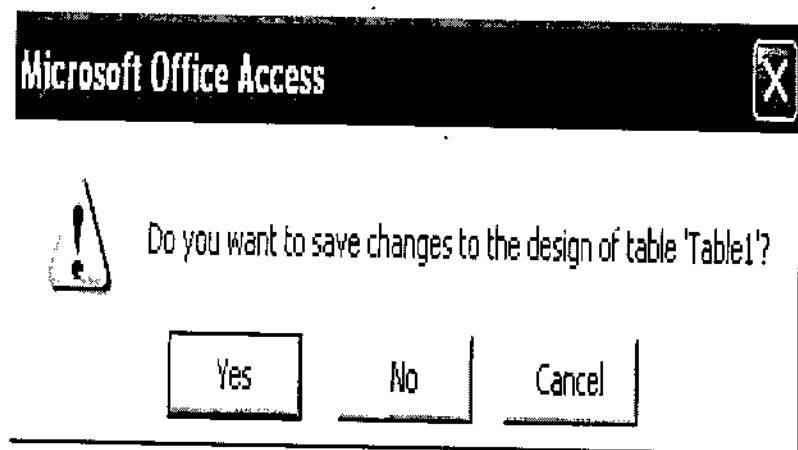
H103

Chọn xong, cửa sổ thiết kế bảng sẽ hiện ra như hình sau:

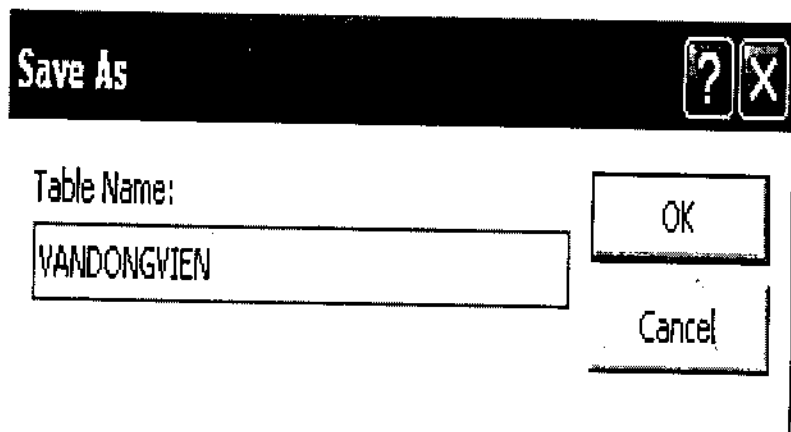


H104 Cửa sổ thiết kế bảng

Cột **Field Name** để đặt tên vùng, cột **Data Type** dùng để chọn kiểu của vùng, cột **Description** dùng để chú thích tên của vùng. Khi nhập xong các chi tiết và xác lập khóa cho một bảng, bạn click dấu X trên góc phải để đóng, hộp thoại hiện ra hỏi bạn có lưu hay không. Bạn click YES hộp thoại kế tiếp hiện ra cho các bạn đặt tên, bạn gõ tên vào và click OK. Ví dụ đối với bảng VANDONGVIEN.



H105



H106

Ghi chú:

Trong bảng VANDONGVIEN có mục TUOI kiểu Integer, bạn chọn Number sau đó chọn Integer.

Table1 : Table

	Field Name	Data Type	
	MSGT	Text	Ma so giai thuong
▶	DIEM	Number	

Field Properties

General | Lookup

Field Size

Format

Decimal Places

Input Mask

Caption

Default Value

Validation Rule

Validation Text

Required

Indexed

Smart Tags

Long Integer

Byte

Integer

Long Integer

Single

Double

Replication ID

Decimal

No

No

H107

☐ Các kiểu có trong cột Data Type

STT	Kiểu	Mô tả
1	Text	Chứa bất kỳ kí tự, dài tối đa 255 ký tự
2	Memo	Ghi chú, có thể chứa 65.535 ký tự
3	Number	Chứa giá trị số, trong này có nhiều loại
4	Date/Time	Ngày hoặc giờ
5	Currency	Giá trị tiền tệ, có dấu dollar đứng đầu
6	AutoNumber	Số tự động liên tục
7	Yes/No	Trị luận lý đúng hoặc sai
8	OLE Object	Trường liên kết và nhúng các đối tượng như hình ảnh, bảng tính, Word...
9	Hyperlink	Text hoặc số được chứa dưới dạng siêu

		liên kết. Dùng để chứa các địa chỉ Web site.
10	Lookup Wizard	Một liên kết đến một bảng khác, hoặc đến danh sách các giá trị để chèn dữ liệu vào bảng đang có. Chọn tùy chọn này sẽ khởi động một Wizard.
11	Unicode Compression	Bạn nên chọn Yes

▣ Các mục trong tab General bên dưới với kiểu Text

STT	Mục	Mô tả
1	Field Size	Chiều dài tối đa cho phép nhập liệu
2	Format	Định dạng số liệu hiển thị (không bắt buộc)
3	Input Mask	Quy định mặt nạ nhập liệu (không bắt buộc)
4	Caption	Tiêu đề cho field, dùng cho Form và Report
5	Default Value	Trị ngầm định nếu không nhập
6	Validation Rule	Biểu thức kiểm soát số liệu nhập
7	Validation Text	Thông báo khi không thoả Validation Rule
8	Required	Nếu chọn Yes, bắt buộc phải nhập vào field này
9	Allow Zero Length	Nếu chọn Yes, chấp nhận chuỗi rỗng
10	Indexed	Chọn No, Yes Duplicate, hoặc Yes No duplicate

▣ Các mục trong tab General bên dưới với kiểu Number

STT	Mục	Mô tả
1	Field Size	Chọn Byte, Integer, Long Integer, Single ...
2	Format	General, Currency, Standard, Fixed, Percent, Scientific.

3	Decimal Place	Số cột số lẻ hoặc chọn Auto
4	Input Mask	Quy định mặt nạ nhập liệu
5	Caption	Một diễn đạt khác cho Field, dùng cho Form và Report.
6	Default Value	Trị ngầm định.
7	Validation Rule	Biểu thức kiểm chính số liệu nhập
8	Validation Text	Thông báo khi không thoả Validation Rule
9	Required	Nếu chọn Yes, bắt buộc phải nhập vào field này
10	Indexed	Chọn No, Yes Duplicate, hoặc Yes No duplicate

Các mục trong tab General bên dưới với kiểu Date/Time

STT	Mục	Mô tả
1	Format	General Date, Long Date, Medium Date, Short Date, Long Time, Medium Time ...
2	Input Mask	Quy định mặt nạ nhập liệu
3	Caption	Tiêu đề cho field, dùng cho Form và Report
4	Default Value	Trị ngầm định nếu không nhập
5	Validation Rule	Biểu thức kiểm chính số liệu nhập
6	Validation Text	Thông báo khi không thoả Validation Rule
7	Required	Nếu chọn Yes, bắt buộc phải nhập vào field này
8	Indexed	Chọn No, Yes Duplicate, hoặc Yes No duplicate

Trên là ba kiểu thường dùng, các kiểu còn lại các bạn tự tìm hiểu.

Trong cột **Field Name**, bạn gõ vào **MSVDV**, nhấn **phím Tab**, con trỏ nhảy sang cột **Data Type**, trị gợi ý luôn là **Text**, ở đây là đúng rồi, bạn click chuột vào textbox có tên là **Field Size** bên dưới, sửa 50 thành 5. Nếu không phải là kiểu **Text**, bạn click chuột vào mũi tên bên phải của cột **Data Type** này để chọn kiểu thích hợp và chọn **độ lớn** của

kiểu này. Sau khi khai báo xong, bạn click con trỏ chuột ở cột **Description** để chú thích nếu như bạn cảm thấy cần thiết. Ở đây bạn nhập vào **Ma so van dong vien**.

Bạn nhấn **phím Tab** để tiếp tục khai báo các Field còn lại là **Hoten, Nam, Tuoi, Nuoc, Monthi**. Như bảng cấu trúc ở trên.

Ghi chú: Trong tuổi, bạn xác lập Validation rule là ≥ 10 (Vận động viên phải từ 10 tuổi trở lên). Đây là ràng buộc toàn vẹn trên một vùng. Trước khi giải thích tiếp, chúng tôi giới thiệu đến bạn các mối liên hệ giữa các bảng trong một cơ sở dữ liệu như sau:

◆ Các mối liên hệ trong cơ sở dữ liệu

Có 3 loại mối liên hệ như sau:

- Mối liên hệ một – một được kí hiệu "1-1" (One to one).
- Mối liên hệ một – nhiều được kí hiệu "1-n" (One to many).
- Mối liên hệ nhiều – nhiều được kí hiệu "n-n" (Many to many).

Liên hệ được thiết lập giữa một bảng dữ liệu cha mẹ (parent table) với bảng dữ liệu con cái (child table). Bảng cha mẹ là đối tượng chính, còn các bảng dữ liệu con cái là những đối tượng có liên hệ với bảng cha mẹ theo thể thức dữ liệu nào đó mà dữ liệu trong bảng con cái chỉ hiện hữu nếu nó có dữ liệu liên hệ trong bảng dữ liệu cha mẹ.

Khi nào dùng mối liên hệ một – một? Khi bạn muốn tạo một mối liên hệ trực tiếp giữa hai đối tượng, một bảng dữ liệu cha mẹ và một bảng dữ liệu con cái. Mối liên hệ một – một đòi hỏi các cột có liên hệ trên hai bảng dữ liệu phải là duy nhất. Thông thường thông tin hình thành mối liên hệ một – một có thể cho vào một bảng dữ liệu. Nhưng đôi khi cũng phải cho một vài thông tin trên một bảng dữ liệu riêng biệt, rồi thực thụ thiết lập một kết nối trực tiếp hoặc một mối liên hệ một – một giữa hai bảng dữ liệu mới.

Lý do tách thông tin ra hai bảng là do một trong những lý do sau:

1/ Cấu trúc bảng dữ liệu là quá lớn: tức là bảng có nhiều cột nhưng thường xuyên chỉ có vài cột là được truy cập. Chúng ta tách ra thành hai bảng, cho di chuyển những cột nào ít được truy cập lên bảng dữ liệu mới.

2/ Vì lý do an ninh: có một vài cột có nhiều rủi ro đối với vấn đề an ninh và như vậy sẽ được chuyển qua bảng dữ liệu mới, rồi tạo

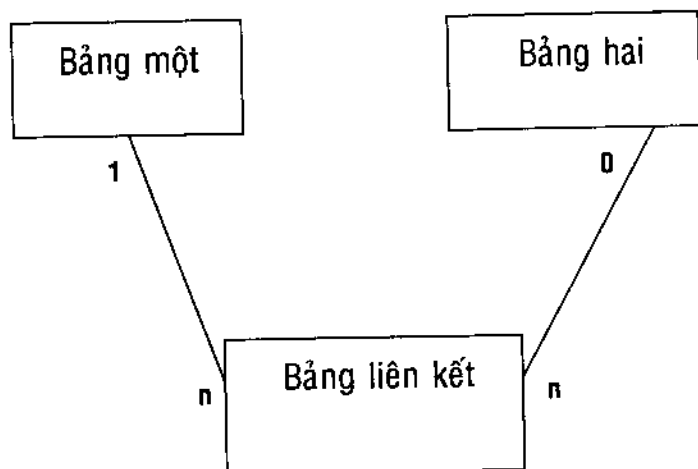
một mối liên hệ một – một. Chúng ta có thể khắc phục vấn đề này bằng cách tạo những view (khung nhìn)

Khi nào dùng mối liên hệ một – nhiều? Đây là mối liên hệ hay dùng nhất, bạn thấy trong các ví dụ mẫu, chúng tôi sẽ dùng mối liên hệ này. Loại liên hệ này hiện hữu giữa một bảng dữ liệu cha mẹ và một bảng dữ liệu con cái trong trường hợp một hàng dữ liệu cha mẹ có thể không có hoặc có nhiều hàng dữ liệu con cái.

Khi nào dùng mối liên hệ nhiều – nhiều? Mối liên hệ này hiện hữu trong những trường hợp theo đây một hàng dữ liệu trên bảng cha mẹ tương ứng với nhiều hàng dữ liệu liên hệ trên bảng dữ liệu con cái, và ngược lại là một hàng dữ liệu trên bảng con cái tương ứng với nhiều hàng dữ liệu liên hệ trên bảng dữ liệu cha mẹ.

Mối liên hệ nhiều – nhiều không thể được mô hình hóa trực tiếp giữa bảng dữ liệu cha mẹ và bảng dữ liệu con cái. Thay vào đó, bạn sẽ định nghĩa hai mối liên hệ một – nhiều sử dụng một bảng dữ liệu thứ ba mang tên là bảng kết nối. Bảng kết nối này chứa giữ một mã nhận diện ID duy nhất của hàng dữ liệu trên hàng dữ liệu cha mẹ cùng với một mã ID duy nhất của hàng dữ liệu trên bảng dữ liệu con cái. Do đó muốn tìm ra một hàng dữ liệu cụ thể nào đó trong bảng liên kết, bạn cần có mã ID duy nhất trên cả hàng dữ liệu cha mẹ lẫn hàng dữ liệu con cái. Cả hai mã ID là ứng viên cho một mục khóa chính hỗn hợp.

Mối liên hệ nhiều nhiều biến đổi như sau



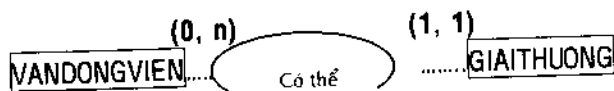
Tiếp theo phần trên là sau khi khai báo xong các Field, bạn di chuyển con trỏ đến tên Field mà bạn muốn chọn làm khóa click chọn và sau đó click **hình chiếc chìa khóa trên thanh công cụ**. Vậy là Field này sẽ là khóa chính, trong table này, chúng tôi chọn MSVDV. Click chuột vào **dấu X** trên góc phải để đóng cửa sổ, lúc này máy sẽ hỏi và bạn có lưu những gì mà bạn vừa tạo. Bạn nhấn **Yes**, hộp thoại cho bạn nhập tên của table vừa tạo hiện ra, bạn nhập vào VANDONGVIEN và click **OK**.

Tương tự như cách tạo bảng VANDONGVIEN, bạn tiếp tục tạo bảng **GIAITHUONG**, chọn vùng MSGT là **khóa chính**. Trong vùng Diem, bạn xác lập Validation rule là ≥ 0 (không được cho điểm âm). Trong table **GIAITHUONG** bạn thấy có field VDVMSVDV, field này là **khóa ngoại**, giá trị của field này sẽ giống với giá trị của field **MSVDV** trong table **VANDONGVIEN**, tương ứng với một vận động viên nào đó. Vậy nếu có một khóa ngoại ở một table nào đó thì chắc chắn sẽ tồn tại một khóa chính của một table khác có giá trị giống với giá trị của khóa ngoại, và table này sẽ có một mối liên hệ với table có chứa khóa ngoại.

Chính sự giống nhau của 2 field này sẽ giúp cho ta truy xuất thông tin chính xác của một vận động viên nào đó với kết quả ra sao.

- ☛ Một vận động viên có thể không được giải thưởng nào hoặc được nhiều giải thưởng.
- ☛ Một giải thưởng nào đó thì có một và chỉ một vận động viên được nhận mà thôi.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:



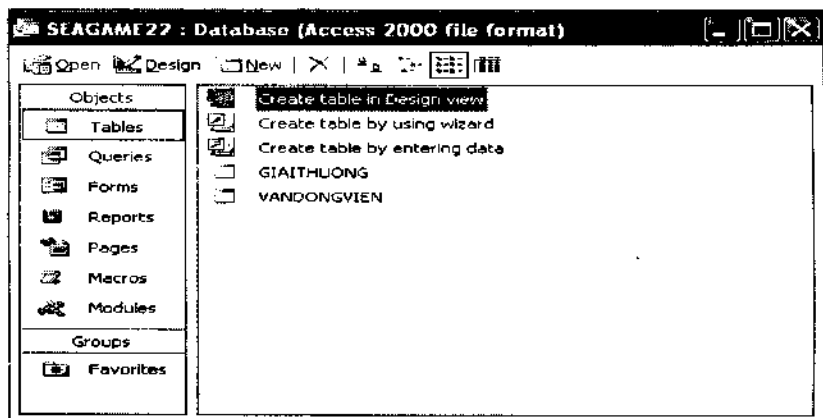
Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ GIAITHUONG qua VANDONGVIEN.

Khi có mối liên hệ một nhiều giữa hai table thì bạn nhớ là khóa ngoại sẽ nằm trong table có giá trị là (0, 1) hoặc (1, 1).

Trong hai Table trên, bạn thấy khoá ngoại sẽ nằm trong table GIAITHUONG, đó là field VDVMSVDV.

Bạn có thể tìm hiểu thêm trong các sách phân tích thiết kế hệ thống và cơ sở dữ liệu.

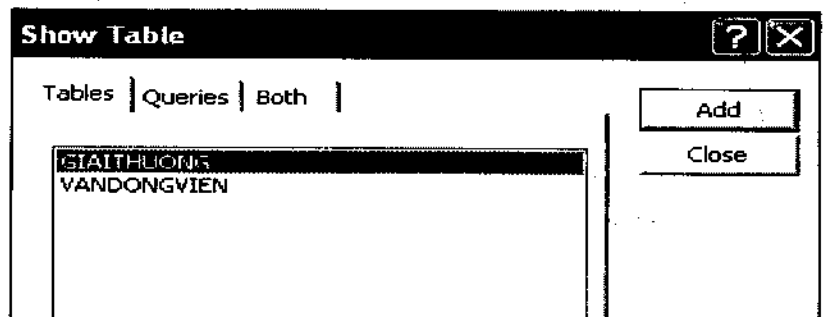
Khi tạo xong hai bảng VANDONGVIEN và GIAITHUONG, chúng ta sẽ thấy trong DataBase có hai bảng như hình sau:



H108

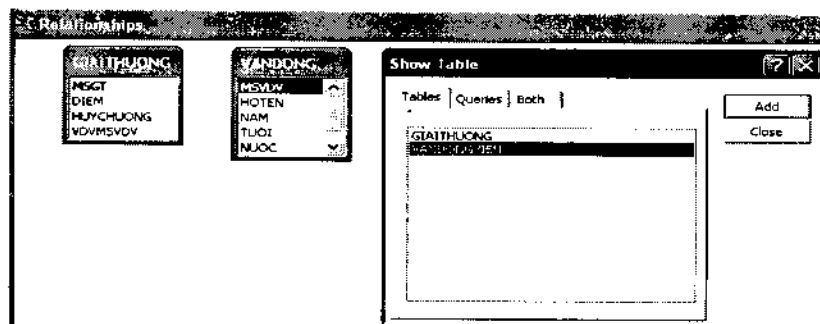
IV Cách tạo mối liên hệ giữa hai Table

Từ cửa sổ SEAGAMES23, chọn **Tools**, chọn **RelationShips**, hoặc click chuột phải ở giữa màn hình DataBase, một menu hiện ra, chọn **RelationShips**, hoặc click vào biểu tượng **RelationShips** trên thanh công cụ. Cả ba trường hợp, cửa sổ Show Table hiện ra, nếu chưa có thì bạn nhấn biểu tượng **Show Table** trong hộp công cụ:



H109 Cửa sổ Show Table

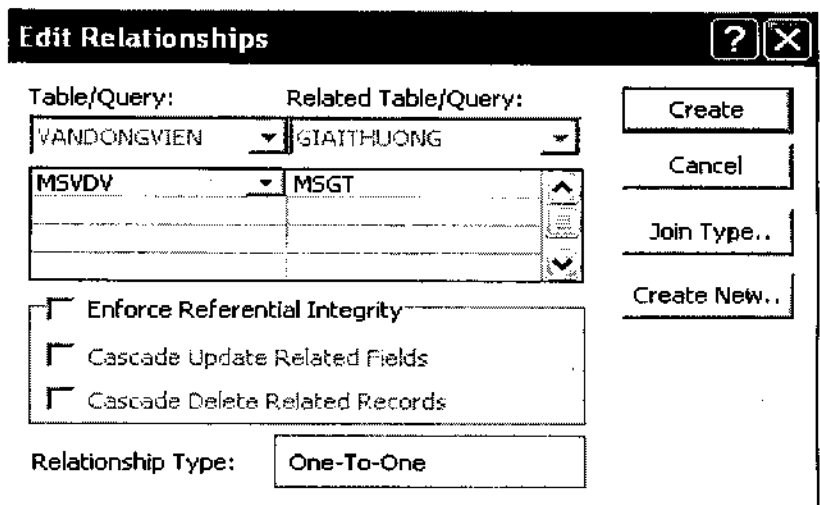
Click chọn lần lượt tên các **table** cần xuất hiện trong cửa sổ **Relationships**, sau mỗi lần chọn, bạn click vào nút **Add**, table vừa chọn sẽ có mặt trong cửa sổ **Relationships**.



H110

Khi cho hai table **GIAITHUONG** và **VANDONGVIEN** xuất hiện ở cửa sổ **Relationships**, bạn tiến hành tạo mối liên kết giữa hai table như sau như sau:

Click chuột vào **MSVDV** ở table **VANDONGVIEN** và drag chuột đến field có tên **VDVMSVDV** ở table **GIAITHUONG** cho đến khi xuất hiện hình chữ nhật nhỏ thì thả nút chuột ra, một cửa sổ sẽ hiện ra như sau:



H111 Cửa sổ cho bạn xác lập mối liên hệ giữa hai Table

Nhìn vào cửa sổ hộp TextBox bên trái có tên VANDONGVIEN và field MSVDV, hộp TextBox bên phải là GIAITHUONG và field VDVMSVDV, ý nghĩa là xác lập mối quan hệ giữa hai table qua hai field trên.

☑ Click chọn mục **Enforce Referential Integrity**, thông tin của 2 mục mờ bên dưới sẽ đậm ra, chọn mục **Cascade Update Related Fields**

Edit Relationships

Table/Query: VANDONGVIEN Related Table/Query: GIAITHUONG

MSVDV MSGT

☒ Enforce Referential Integrity

☒ Cascade Update Related Fields

☐ Cascade Delete Related Records

Relationship Type: One-To-One

Create

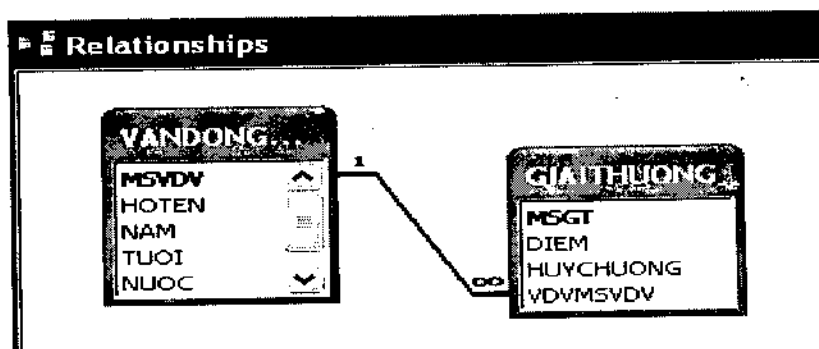
Cancel

Join Type..

Create New..

H112

☑ Click nút **Create**. Sẽ có một liên kết giữa hai field của hai table, một đầu là 1 và đầu kia là ∞ (vô cực). Ta gọi đây là quan hệ **One to Many (một đối nhiều)**, nghĩa là một vận động viên có thể có nhiều giải thưởng. Như hình sau:



H113 Cửa sổ Relationships chứa hai Table đã được kết nối

Bạn nhớ kéo bảng VANDONGVIEN và GIAITHUONG dài ra một chút để có thể thấy hết các field.

☐ Nếu không chọn mục **Enforce Referential Integrity** thì sẽ có quan hệ **One to One** (một một).

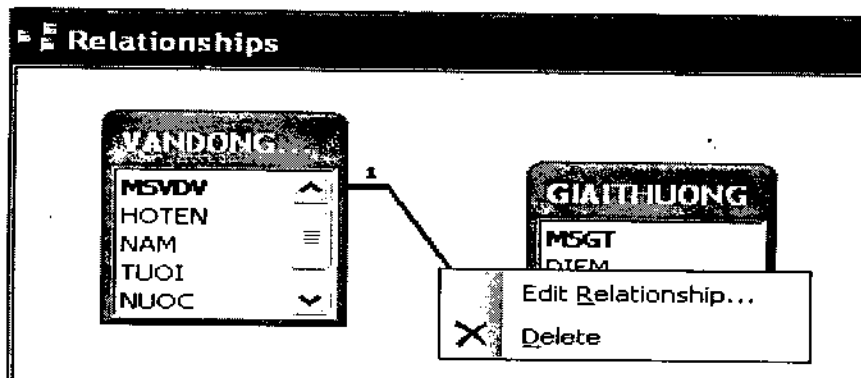
☐ Mục này sẽ buộc máy kiểm tra giá trị mục tin ở phần Many phải ứng với một nội dung nào đó ở phần One, nhờ đó người ta không thể nhập vào một mã số vận động viên không có ở bảng VANDONGVIEN vào table GIAITHUONG được

☐ Mục **Cascade Update Related Field** sẽ cho phép khi sửa mã số vận động viên ở table VANDONGVIEN (đầu one) thì Access sẽ tự động sửa lại mã đó trong tất cả các mẫu tin tương ứng của table GIAITHUONG (đầu many)

☐ Nếu bạn chọn thêm mục **Cascade Delete Related Records** thì khi bạn xóa một Record chứa mã số vận động viên nào ở table VANDONGVIEN thì lúc này Access sẽ tự động xóa những mẫu tin trong table GIAITHUONG có mã số vận động viên giống với mã số vận động viên mà bạn vừa xóa ở table VANDONGVIEN.

Tương tự như vậy, nếu bạn cần tạo thêm mối liên kết giữa hai table nào đó, bạn vào Menu Relationships chọn Show Table và đưa các table vào cửa sổ Relationships như chúng tôi đã hướng dẫn như trên.

Muốn hủy bỏ hay sửa đổi mối quan hệ mà bạn đã tạo, bạn đưa con trỏ chuột đến sợi dây nối hai table click chuột cho dây đậm ra, click phải chuột, bạn sẽ thấy một menu hiện ra gồm hai mục:



H114 Cửa sổ để bạn quy định lại hoặc hủy bỏ mối quan hệ

☐ Nếu click chọn **Edit Relationships** thì sẽ giúp bạn quy định lại cách quan hệ như trên.

☐ Nếu click chọn **Delete** thì sẽ cắt đứt mối quan hệ giữa hai table.

Ghi chú:

Muốn xoá bảng trong cửa sổ Relationships, bạn click nút phải chuột trên bảng đó, chọn Hide Table.

Để thuận tiện cho việc thực hiện thử các câu lệnh SQL, bạn nhập các thông tin sau vào hai table. Bạn nhập thông tin cho table VANDONGVIEN trước, nếu không nhập các mã số vận động viên thì khi nhập mã số vận động viên ở table GIAITHUONG thì Access sẽ báo lỗi và yêu cầu bạn phải bổ sung mã số vận động viên này ở table VANDONGVIEN (vì chưa có), đây là một ràng buộc mà chúng tôi đã đề cập ở trên.

Nhập thông tin cho Table

Trong cửa sổ SEAGAMES23, bạn chọn table **VANDONGVIEN**, click chuột vào nút **Open**. Cửa sổ nhập thông tin hiện ra, bạn nhập thông tin vào cho table VANDONGVIEN. Theo các tiêu đề bên trên của các Field. Giả sử bạn nhập các thông tin như sau:

VANDONGVIEN : Table						
	MSVDV	HOTEN	NAM	TUOI	NUOC	MONTHI
▶ *	BB001	Mai văn Hiếu	☒	24	Việt Nam	Bóng bàn
	* BB002	Kanjung	☒	21	Hàn Quốc	Bóng bàn
	* BB003	Nguyễn Anh Dũng	☒	20	Việt Nam	Bóng bàn
	* BR004	Nguyễn Hoàng Nam	☒	20	Việt Nam	Bóng rổ
	* BR005	Winza	☒	19	Mã Lai	Bóng rổ
	* BS009	Nguyễn Văn Hùng	☒	18	Việt Nam	Bắn súng
	* BS010	Kakazu	☒	18	Hàn Quốc	Bắn súng
	* DK006	Mai Quyết Thắng	☒	19	Việt Nam	Điện kinh
	* DK007	Nguyễn Thị Hoa	☐	18	Việt Nam	Điện kinh
	* DK008	Hanchunzu	☒	18	Hàn Quốc	Điện kinh
*			☒	0		

H115 Thông tin nhập vào bảng Vandongvien

Ghi chú: chúng tôi chỉ nhập vài vận động viên và vài môn thi, thực tế thì có rất nhiều,

khi hiểu rồi thì nhập bao nhiêu cũng được, chú ý ở cột Nam, khi nhập trong Access bạn nhấn phím Spacebar (phím dài nhất) để đánh dấu chọn nếu là Nam hoặc có thể click chuột vào ô chọn, không chọn gì cả sẽ là Nữ.

Khi nhập xong, bạn đóng table lại, hộp thoại hiện ra hỏi chúng ta có muốn lưu không? Lưu với tên gì?... các lần sau thông tin nhập sẽ tự động lưu vào tập tin VANDONGVIEN.

Khi muốn **sửa lại cấu trúc của table**, bạn chọn **table** trong cửa sổ **DataBase**, click chuột vào nút **Design**, sửa lại hoặc thêm vào sau đó lưu những gì mới sửa đổi.

Tương tự như vậy, bạn nhập thông tin như sau cho bảng Giaithuong.

GIAITHUONG : Table			
	MSGT	DIEM	HUYCHUONG
▶	001BB	20 Vàng	BB001
	001BR	20 Vàng	BR004
	001BS	20 Vàng	BS009
	001DK	20 Vàng	DK006
	002BB	19 Bạc	BB003
	002BR	18 Bạc	BR005
	002BS	19 Bạc	BS010
	002DK	19 Bạc	DK007
	003BB	18 Đồng	BB002
*		0	

H116 Nhập thông tin cho bảng Giaithuong trong Access

Phần trên các bạn đã học được một số kiến thức cơ bản về CSDL và cách tạo một cơ sở dữ liệu và các bảng trong Access, cách tạo mối quan hệ giữa các bảng, cách nhập thông tin cho các bảng. Chúng tôi chỉ giới thiệu hai bảng, nếu có nhiều bảng trong một CSDL thì cũng thực hiện như vậy.

Khi truy vấn thông tin từ các bảng của một CSDL, người ta thường dùng các câu lệnh SQL, chúng tôi sẽ giới thiệu các câu lệnh này trong các bài tập sau, bây giờ chúng ta sẽ qua bài tập nói về lập trình CSDL trên Visual Basic .NET sử dụng CSDL Access.

BÀI 1

THEO DÕI KẾT QUẢ SEAGAMES 23

Giới thiệu các bước khi thiết kế chương trình

Trước khi thiết kế chương trình trên Visual C# .NET dùng cơ sở dữ liệu Access. Bạn tiến hành các bước như sau:

◆ Thiết kế cơ sở dữ liệu trong Access, trong đó bao gồm tạo các bảng, tạo mối liên hệ giữa các bảng có liên quan, bạn xem hướng dẫn ở bài trên.

◆ Vào Visual Studio .NET, chọn **Visual C# Projects** ở khung bên trái, chọn **Windows Application** trong khung bên phải (các bài trong phần 2 chúng ta chọn Console Application)

Ghi chú:

Trong các bài tập về lập trình CSDL chúng ta sẽ dùng Windows Application.

- ◆ Tạo kết nối với cơ sở dữ liệu trong Access
- ◆ Đưa đối tượng `oleDbDataAdapter` vào khay công cụ
- ◆ Tạo đối tượng `dataSet` và đưa vào khay công cụ
- ◆ Đưa các điều khiển vào Dialog
- ◆ Viết code cho các điều khiển tác động

Trong bài này, chúng tôi dùng lại CSDL **SEAGAMES23.mdb** vừa tạo ở trên để dễ dàng và thuận tiện cho bạn. Chúng ta sẽ tạo một Dialog có hai điều khiển lưới, có hai `oleDbDataAdapter`, hai `dataset` và một đối tượng kết nối `oleDbConnection` (vì chúng ta chỉ kết nối đến một CSDL).

◆ Các đối tượng `oleDbDataAdapter1` và `datasetVANDONGVIEN1` dùng cho bảng VANDONGVIEN.

◆ Các đối tượng `oleDbDataAdapter2` và `datasetGIAITHUONG1` dùng cho bảng GIAITHUONG.

◆ Đối tượng `oleDbConnection1` dùng chung cho cả hai bảng.

* Yêu cầu

■ Tạo chương trình chỉ cần một Form, trên Form này có hai điều khiển lưới, một để thể hiện thông tin của bảng VANDONGVIEN, một để thể hiện thông tin của bảng GIAITHUONG. Hai bảng này độc lập với nhau. Có hai button, một để xem thông tin của hai bảng và một để cập nhật thông tin khi cần thiết.

* Form cần thiết kế

The screenshot shows a Windows application window with the title bar 'CHƯƠNG TRÌNH VIET BANG C# (QUAN LY SEAGAMES23)'. Inside the window, there are two data grids. The first grid has columns labeled MSVDV, HOTEN, NAM, TUOI, NUOC, and MONTHI. The second grid has columns labeled MSGT, DIEM, HUYCH, and VDVMS. Below the grids, there are two buttons: 'XEM THÔNG TIN' and 'CẬP NHẬT'.

oleDbDataAdapter1

oleDbConnection1

dataSetVANDONGVIEN1

oleDbDataAdapter2

dataSetGIAITHUONG1

H117

Bảng thuộc tính của những điều khiển trong Dialog

STT	Đối tượng	Thuộc tính	Giá trị thiết đặt
1	Form	Font Text	VNI-Times, size 12 Bold CHƯƠNG TRÌNH VIET BANG C# (QUAN LY SEAGAMES23)
2	DataGrid	Name BackColor BackgroundColor ForeColor	DataGrid1 Màu cam nhạt Màu hồng nhạt Màu xanh dương

		GridLineColor Font Anchor Datasource Datamember	Màu đỏ VNI-Times, size 12, bold Top, Left, Right, Bottom dataSetVANDONGVIEN1 VANDONGVIEN
3	DataGrid	Name BackColor BackgroundColor ForeColor GridLineColor Font Anchor Datasource Datamember	DataGrid1 Màu cam nhạt Màu hồng nhạt Màu xanh dương Màu đỏ VNI-Times, size 12, bold Top, Left, Right, Bottom dataSetGIAITHUONG1 GIAITHUONG
4	Button	Name BackColor Font Text	ButtonXem Màu xám VNI-Times, size 12, Bold XEM THÔNG TIN
5	Button	Name BackColor Font Text	ButtonCapnhathat Màu xám VNI-Times, size 12, Bold CẬP NHẬT

① Thiết kế Form của chương trình

Tạo Project mới với tên UNGDUNG1 nằm trong thư mục gốc của ổ đĩa E:\.

◆ Vào menu File, di chuyển vệt sáng đến New, chọn Project, hộp thoại New Project hiện ra

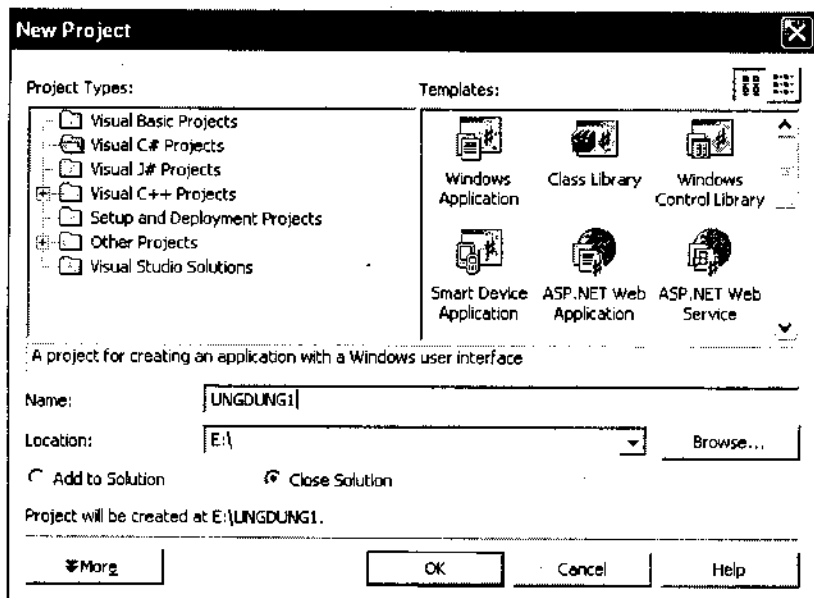
◆ Trong khung Project Types ở bên trái, chọn Visual C#. Projects

◆ Trong khung Templates bên phải, chọn Windows Application

◆ Trong khung Name, nhập tên UNGDUNG1

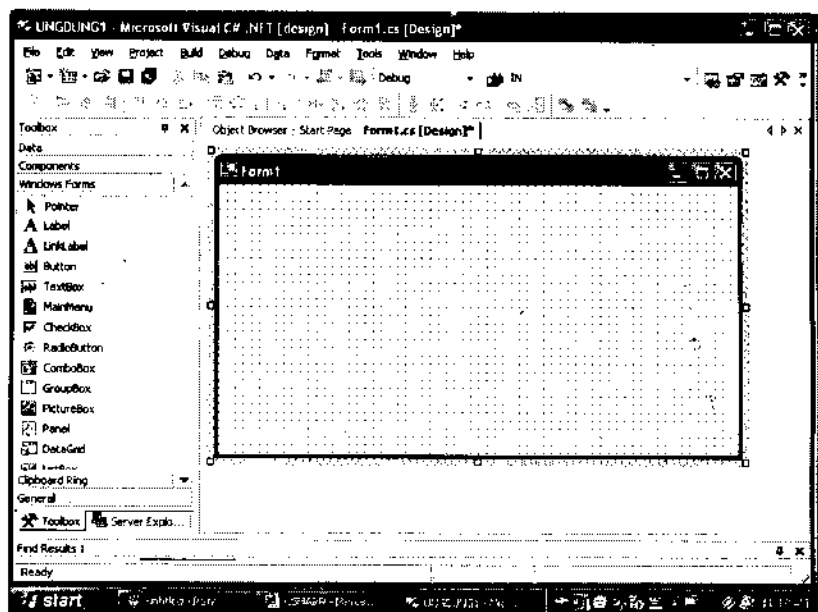
◆ Chọn E:\ trong khung Location

Bạn thấy như hình sau:



H118

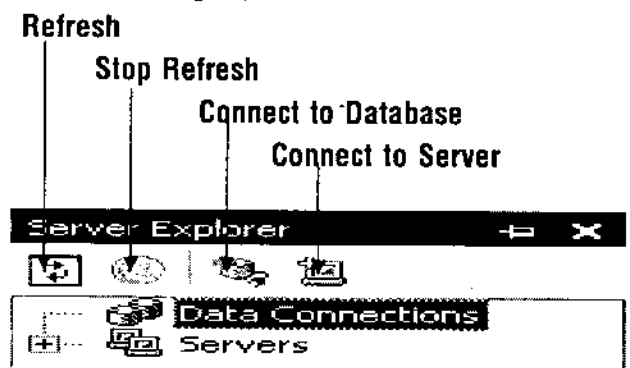
Click nút **OK**, cửa sổ thiết kế xuất hiện như hình sau:



H119

Đầu tiên chúng ta sẽ kết nối với cơ sở dữ liệu SEAGAMES23.mdb trong Access.

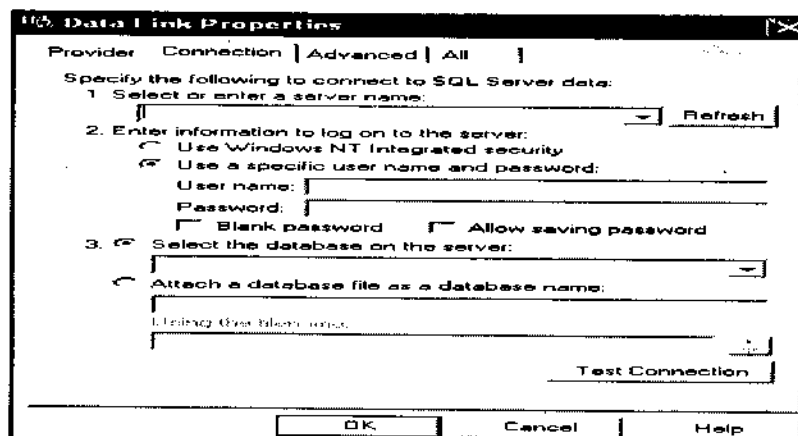
◆ Vào menu View, chọn **Server Explorer**. Cửa sổ này sẽ hiện tại vị trí bên trái của Diaolog. Bạn xem cửa sổ và các icon như hình sau:



H120

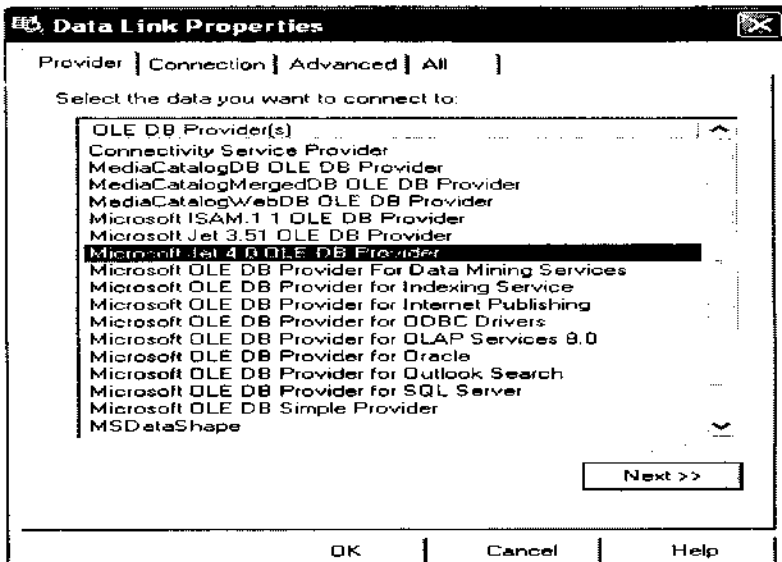
Đây là công cụ cho phép bạn thiết lập kết nối đến CSDL.

☐ Click nút **Connect to Database** trong cửa sổ **Server Explorer**, hộp thoại kết nối hiện ra (hộp thoại Data Link Properties).



H121

☐ Chọn tab **Provider**, cửa sổ Data Link Properties hiện ra. Bạn chọn Microsoft Jet 4.0 OLE DB Provider, bạn có thể thấy như hình sau:

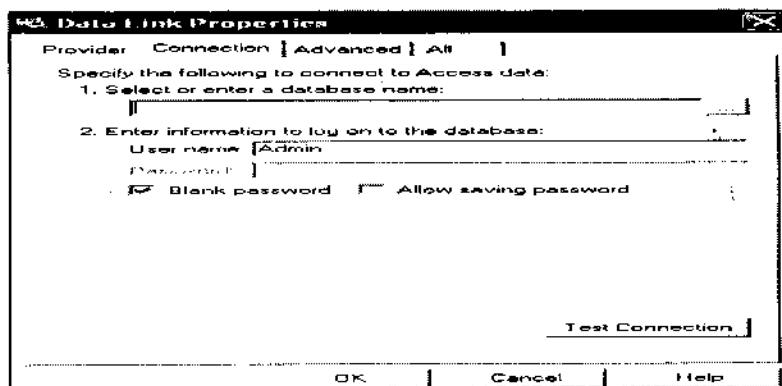


H122 Tab Provider, chọn Microsoft jet 4.0 OLE DB Provider

Ghi chú:

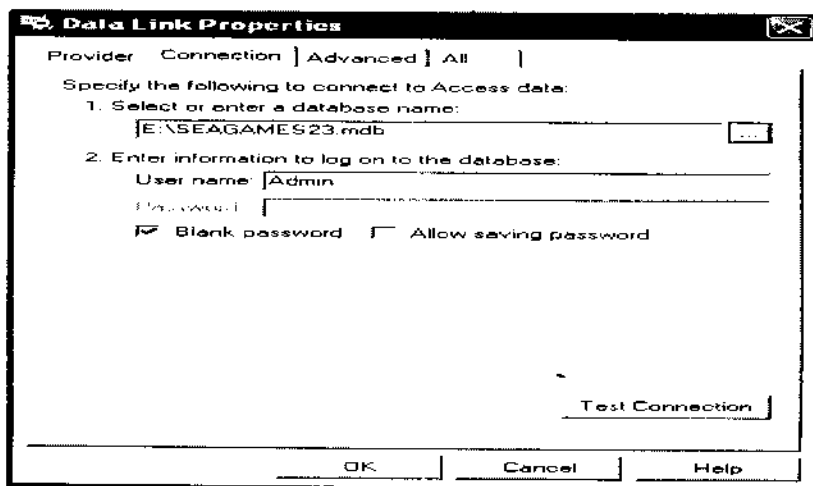
Microsoft Jet 4.0 OLE DB Provider là bộ máy cho phép truy xuất đến CSDL của Microsoft Access. Nếu bạn dùng CSDL **Oracle** thì bạn sẽ chọn **Microsoft OLE DB Provider for Oracle**. Nếu bạn dùng **SQL Server** thì bạn sẽ chọn **Microsoft OLE DB Provider for SQL Server**.

☐ Click nút **Next**, sẽ chuyển sang tab Connection và hộp thoại như hình sau hiện ra



H123 Hộp thoại khai báo Database cần kết nối

Click nút ba chấm (...) bên phải khung **Select or enter a database name** để tìm **Database** có tên **SEAGAMES23.mdb** (bạn có thể gõ trực tiếp địa chỉ trong khung nếu như biết chắc chắn, đó là **E:\SEAGAMES23.mdb**). Kết nối xong, bạn sẽ có hình như sau:

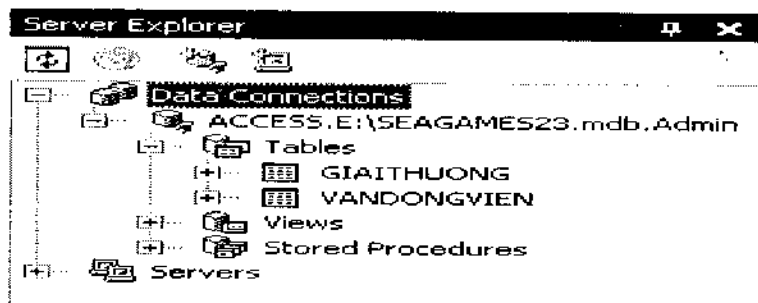


H124 Khi đã kết nối đến Database

Click nút **OK** trên hộp thoại, Visual Studio sẽ thêm kết nối vừa rồi vào **Server Explorer**.

Trong cửa sổ **Server Explorer** ở phía bên trái màn hình, Click dấu + trước dòng **ACCESS. E:\SEAGAME23.mdb.Admin**

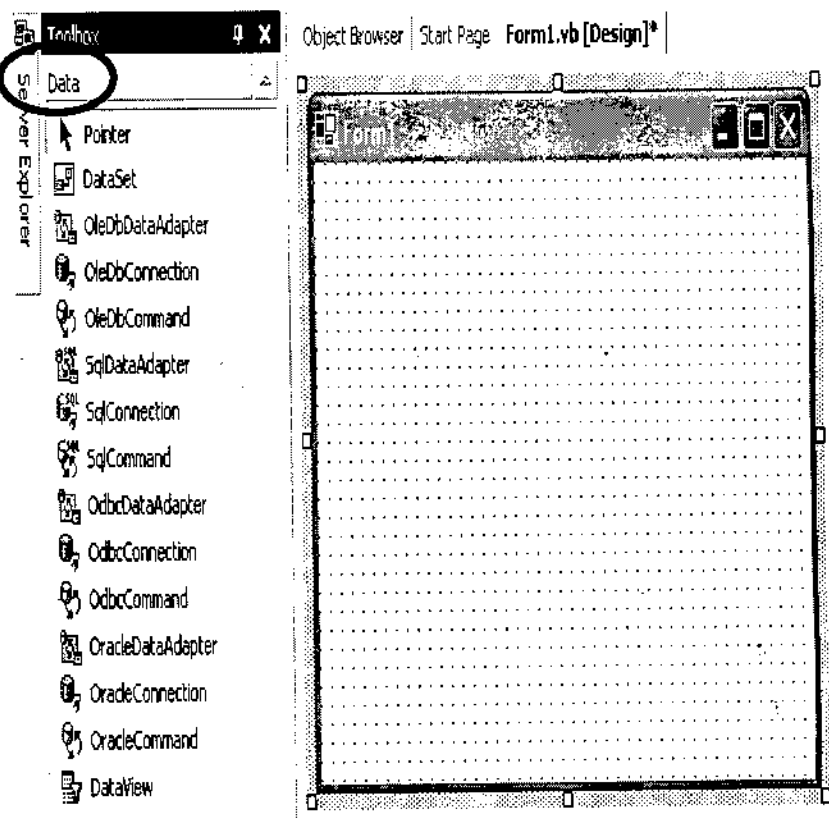
Click dấu + trước **Tables**, bạn sẽ thấy hình như sau



H125

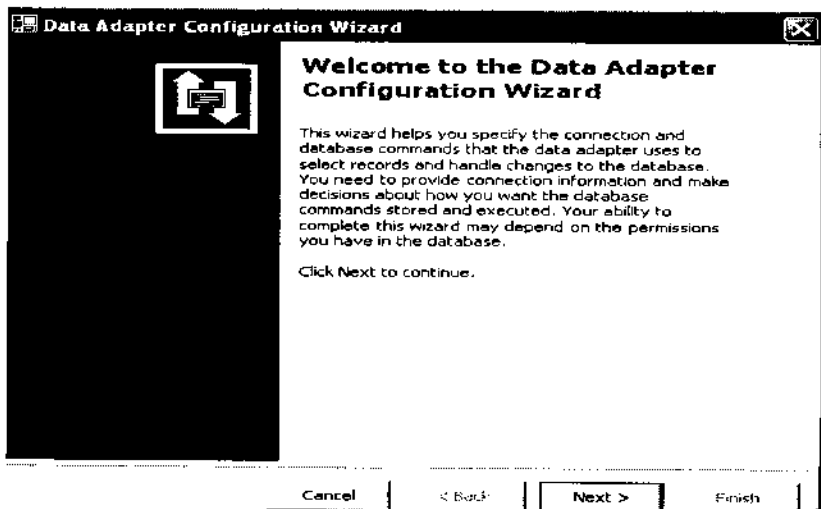
Các bước trên chúng ta đã kết nối đến CSDL có tên SEAGAME23.mdb. Để truy cập thông tin trong CSDL trên, chúng ta phải thêm một bộ điều phối dữ liệu (Data Adapter). Ở đây chúng ta sẽ sử dụng điều khiển OleDbDataAdapter có sẵn trong hộp công cụ (Toolbox). Nếu cửa sổ Toolbox chưa có, bạn click vào biểu tượng  trên thanh công cụ, hộp Toolbox sẽ hiện ra ngay vị trí của cửa sổ Server Explorer trước đó.

☐ Click chọn tab **Data** nằm phía trên trong hộp công cụ như hình sau:



H126 Server Explorer thu nhỏ lại bên trái nhường chỗ cho hộp công cụ

☐ Click chọn và kéo đối tượng OleDbDataAdapter từ hộp công cụ vào trong Form của chương trình, vẽ trong form tại vị trí bất kỳ. Visual Studio sẽ khởi tạo trình Data Adapter Configuration Wizard như hình sau:



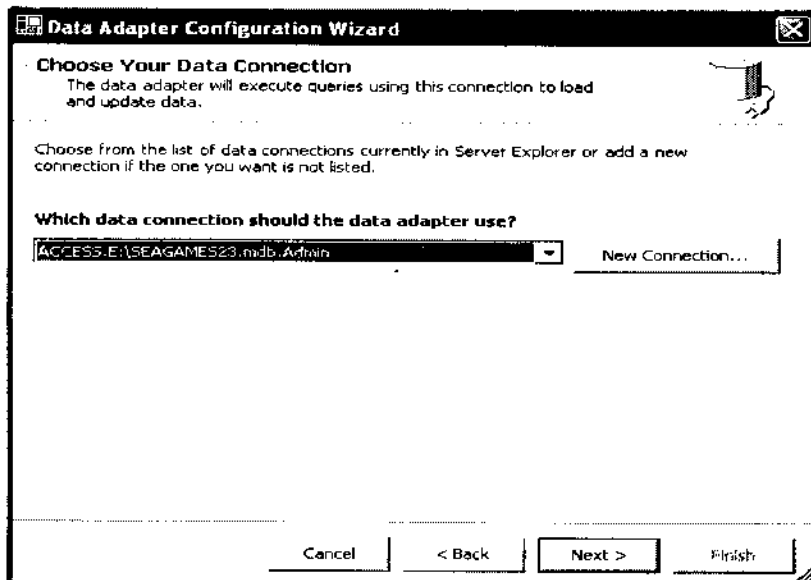
H126

☐ Click nút **Next**, cửa sổ Data Adapter Configuration Wizard hiện ra, nếu trước đó bạn đã viết các chương trình trên Visual Basic .NET hoặc trên Visual C++ .NET, hoặc trên Visual C# .NET ... đã kết nối đến các cơ sở dữ liệu khác, các bạn sẽ thấy chúng xuất hiện trong cửa sổ này khi bạn click vào đầu mũi tên quay xuống. Trong máy của chúng tôi lúc này chưa có cơ sở dữ liệu nào được kết nối trước đó. Chúng tôi đã xóa hết để đỡ rối cho các bạn khi nhìn vào.

Ghi chú:

Nếu đã kết nối với các CSDL trước đó, khi viết chương trình, vào cửa sổ Server Explorer, bạn sẽ thấy tên các CSDL đã kết nối này, thông thường trước tên các CSDL này có dấu gạch chéo màu đỏ, ý nghĩa là đang bị cắt, nếu đã có CSDL cần kết nối, bạn chỉ cần click vào dấu "+" phía trước, nó sẽ được kết nối trở lại.

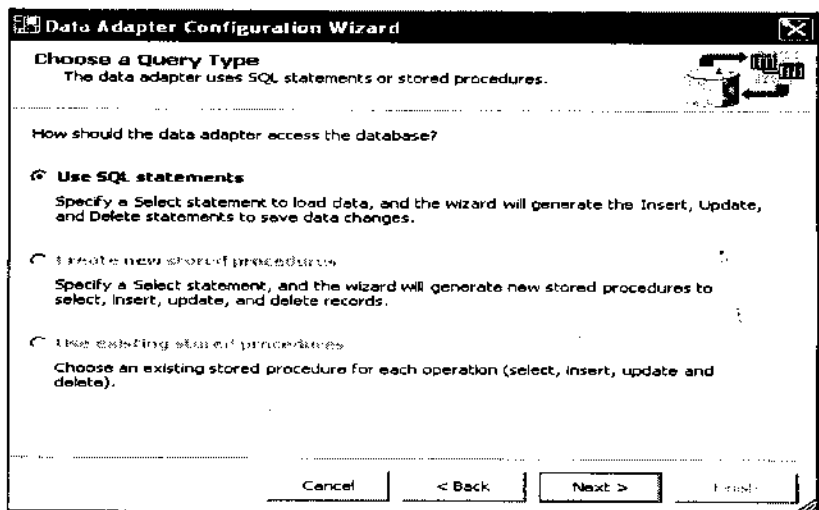
Bạn sẽ thấy cửa sổ Data Adapter Configuration Wizard hiện ra như hình sau:



H127

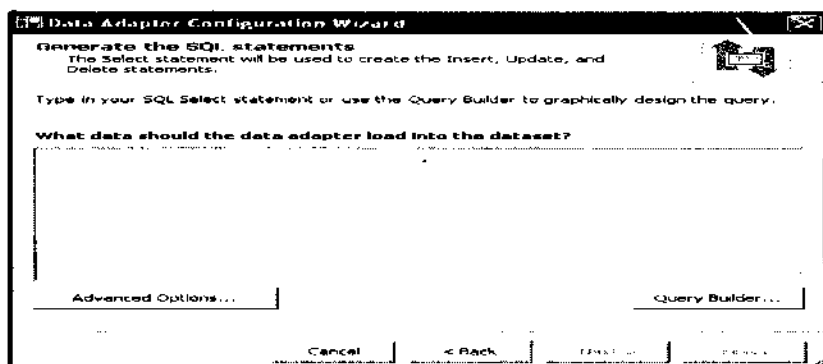
◆ Click đầu mũi tên quay xuống, trong danh sách các CSDL đã kết nối, bạn chọn CSDL có tên SEAGAMES23.mdb, chọn xong bạn sẽ được như hình trên.

◆ Click nút Next, cửa sổ sau hiện ra



H128

- ◆ Chọn Use SQL statements (nếu như chưa được chọn)
- ◆ Click nút Next, cửa sổ sau hiện ra



H129

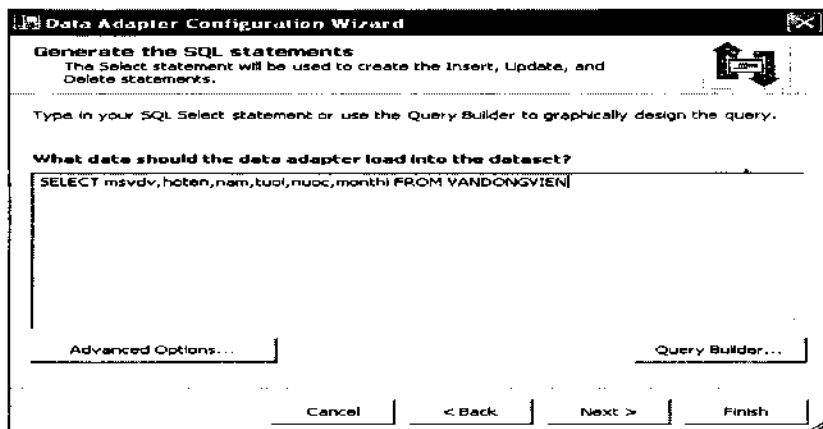
Bạn nhập câu lệnh SQL vào trong khung như sau:

SELECT msvdv, hoten, nam, tuoi, nuoc, monthi FROM VANDONGVIEN

Ghi chú:

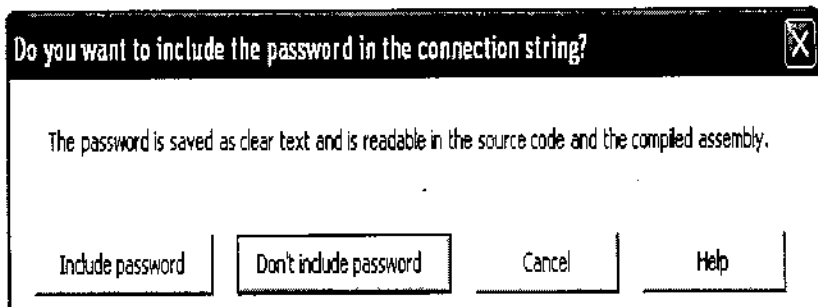
Đối với CSDL Access, nếu bạn nhập thứ tự các vùng như thế nào thì thứ tự này sẽ được thể hiện trên điều khiển lưới cũng như vậy. Nếu bạn nhập câu lệnh **SELECT * FROM VANDONGVIEN** thay cho câu lệnh SQL ở trên, thì các vùng trong điều khiển lưới sẽ hiện theo thứ tự Alphabet.

Khi nhập xong bạn sẽ thấy hình như sau:



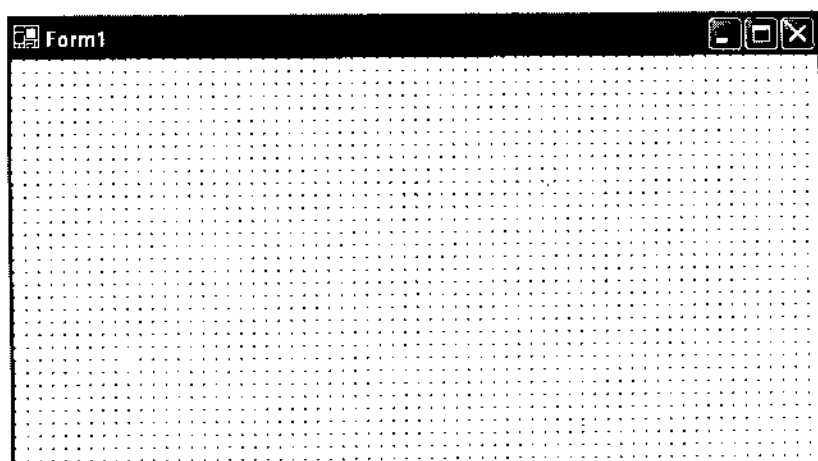
H130

- ◆ Click nút Finish bạn sẽ thấy hộp thoại sau hiện ra



H131

- ◆ Click nút Don't include password. Bạn sẽ thấy đối tượng OleDbDataAdapter1 và OleDbConnection1 xuất hiện trong khay công cụ bên dưới Dialog như hình sau:



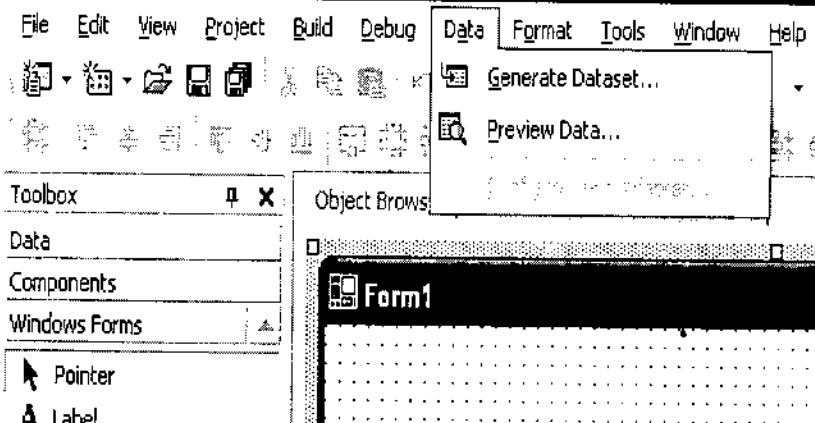
H132

Bước tiếp theo là đặt tên và đưa đối tượng dataSet vào khay công cụ.

Bạn thực hiện như sau:

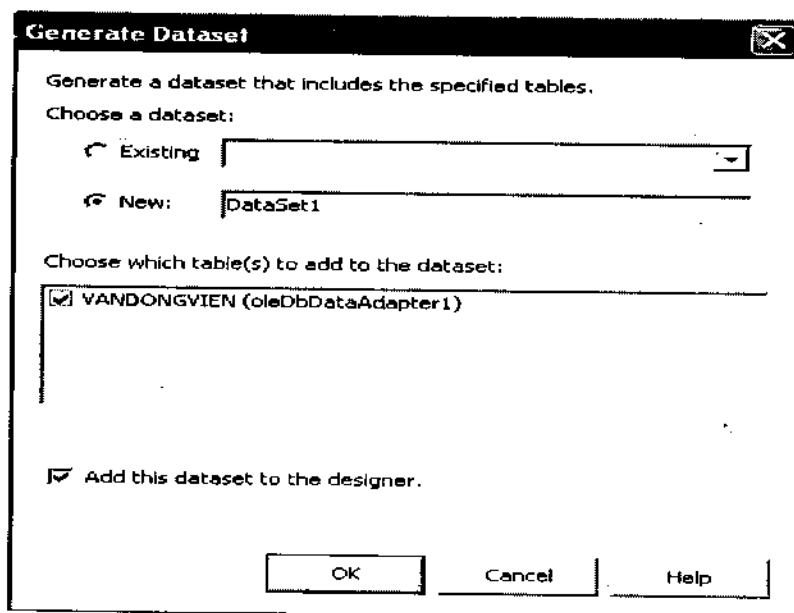
- ◆ Click chọn Dialog
- ◆ Vào menu Data, chọn Generate Dataset.

UNG DUNG 1 - Microsoft Visual C# .NET [design] - Form1.cs [Design]



H133

Click xong, bạn thấy hiện ra hộp thoại sau



H134

Giả sử đặt lại tên trong khung New là DataSetVANDONGVIEN như hình sau:

Generate Dataset

Generate a dataset that includes the specified tables.

Choose a dataset:

- ☐ Existing
- ☒ New:

Choose which table(s) to add to the dataset:

☒ VANDONGVIEN (oleDbDataAdapter1)

☒ Add this dataset to the designer.

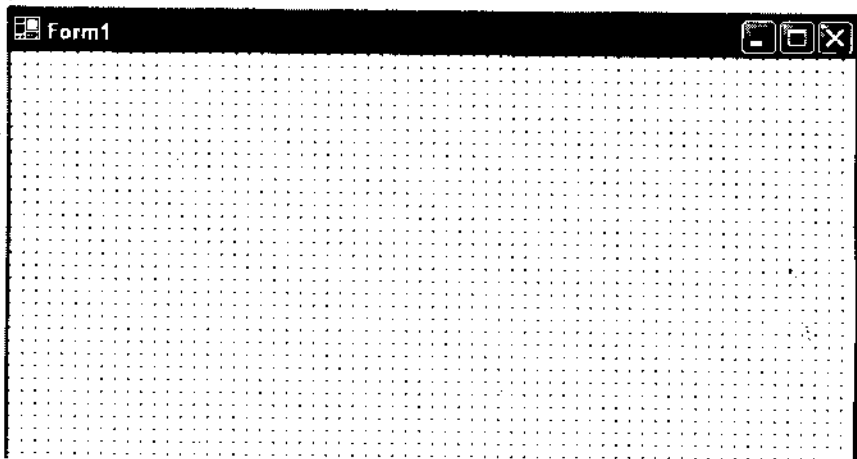
OK

Cancel

Help

H135

◆ Click nút OK để hoàn tất việc đặt tên cho DataSet và đưa DataSet này vào khay công cụ. Bạn sẽ thấy Dialog lúc này như sau



oleDbDataAdapter1

oleDbConnection1

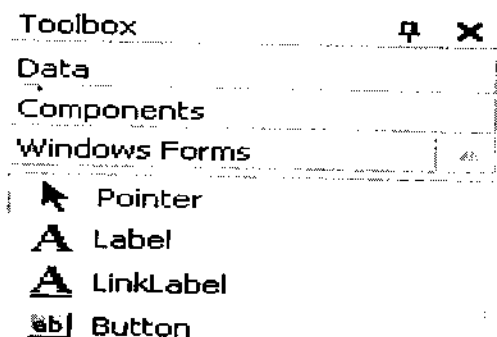
dataSetVANDONGVIEN1

H136

Vậy là bạn đã hoàn thành các bước cơ bản là kết nối với CSDL SEAGAMES23.mdb trong cơ sở dữ liệu Access, đưa được hai đối tượng OleDbDataAdapter 1 và dataSetVANDONGVIEN1 vào khay công cụ.

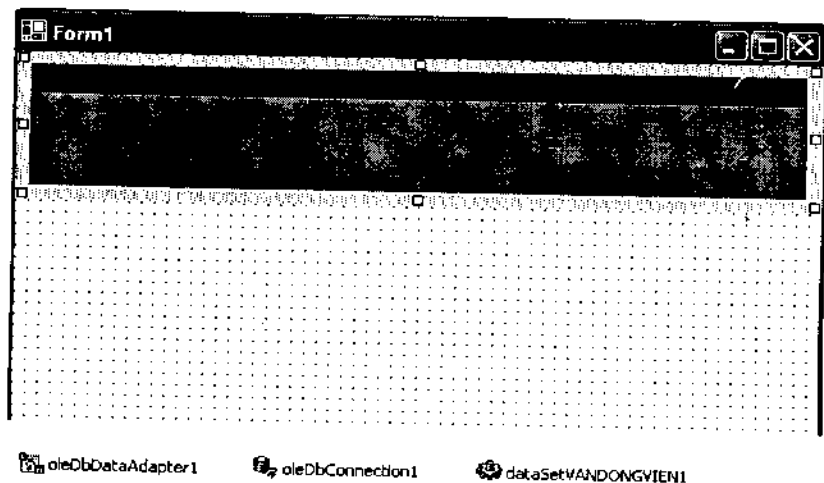
Bước tiếp theo là đưa các điều khiển từ hộp công cụ vào Form, đầu tiên là đưa điều khiển lưới DataGrid vào và vẽ để có dạng như hình kế tiếp

◆ Click chọn tab có tên **Windows Forms** trong hộp công cụ **Toolbox** như hình dưới.



H137

Click nút đầu mũi tên ở thanh trượt bên phải của hộp công cụ, click chọn điều khiển Datagrid, vẽ vào form để có như hình sau:

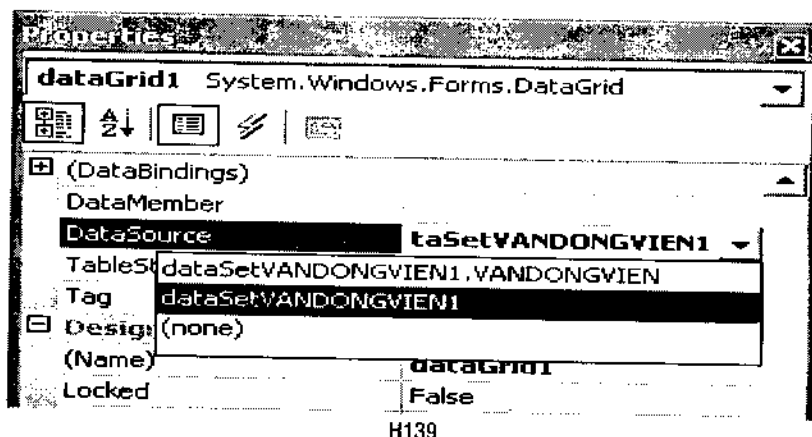


H138

◆ Click nút phải chuột trên điều khiển lưới, chọn **Properties** trong menu hiện ra. Ngoài việc xác lập các thuộc tính đã cho trong bảng trên, chúng tôi hướng dẫn việc xác lập hai thuộc tính quan trọng của điều khiển này, đó là thuộc tính **DataSource** và **DataMember**.

◆ Click chọn khung bên phải của thuộc tính **DataSource** để xuất hiện dấu mũi tên ở góc bên phải

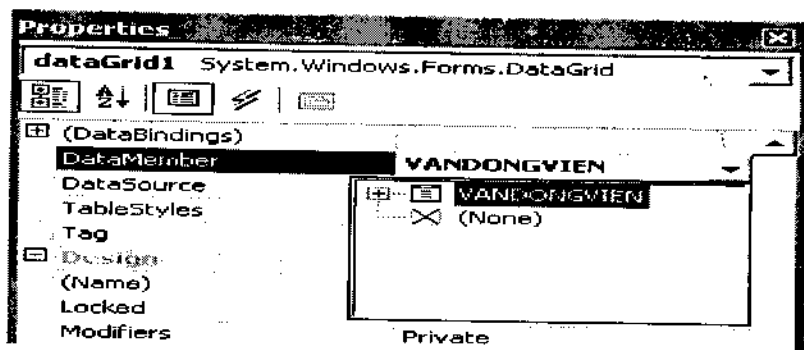
◆ Click vào dấu mũi tên, chọn **dataSetVANDONGVIEN1** trong danh sách buông xuống.



H139

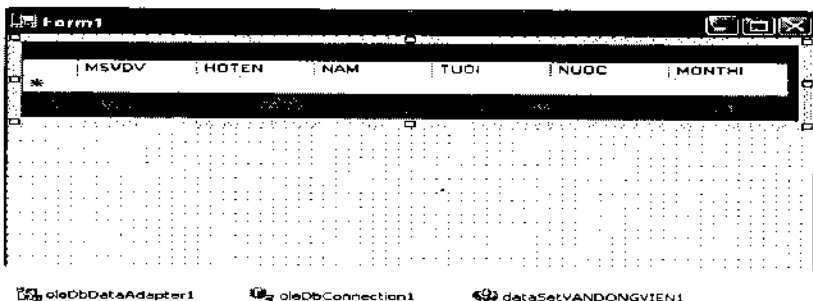
◆ Click chọn khung bên phải của thuộc tính **DataMember** để xuất hiện dấu mũi tên ở góc bên phải

◆ Click vào dấu mũi tên, chọn bảng **VANDONGVIEN** trong khung buông xuống



H140

Đóng hộp thoại Properties, bạn thấy form lúc này như hình sau:



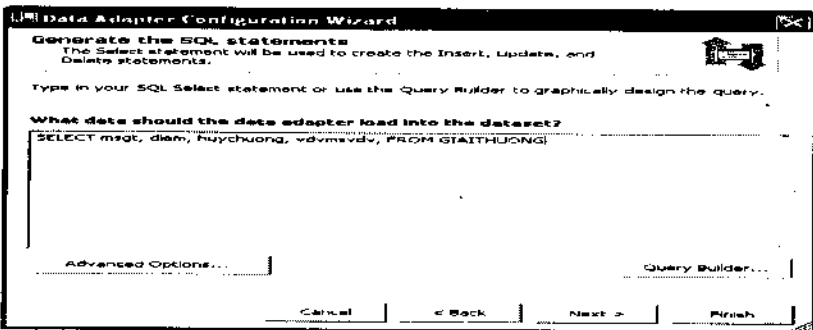
H141

Tiếp theo là tạo bộ điều phối dữ liệu cho bảng GIAITHUONG, bằng cách đưa đối tượng OleDbDataAdapter vào khay công cụ. Cách đưa vào cũng giống như các bước ở trên.

Vì đã kết nối với CSDL SEAGAMES23.mdb rồi nên bước này ta không cần kết nối nữa.

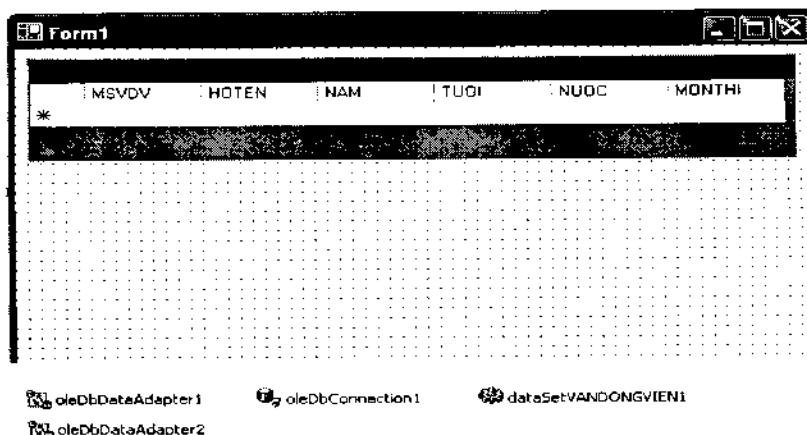
◆ Bạn click chọn tab có tên Data, click chọn đối tượng OleDbDataAdapter, di chuyển con trỏ chuột vào form, vẽ tại vị trí bất kỳ trong form, bạn thấy hộp thoại Data Adapter Configuration Wizard hiện ra, bạn thực hiện các thao tác như đối với bảng VANDONGVIEN cho tới khi hộp thoại để nhập câu lệnh SQL hiện ra. Bạn nhập vào câu lệnh như sau:

```
SELECT msgt, diem, huychuong, vdvmsvdv
FROM GIAITHUONG
```



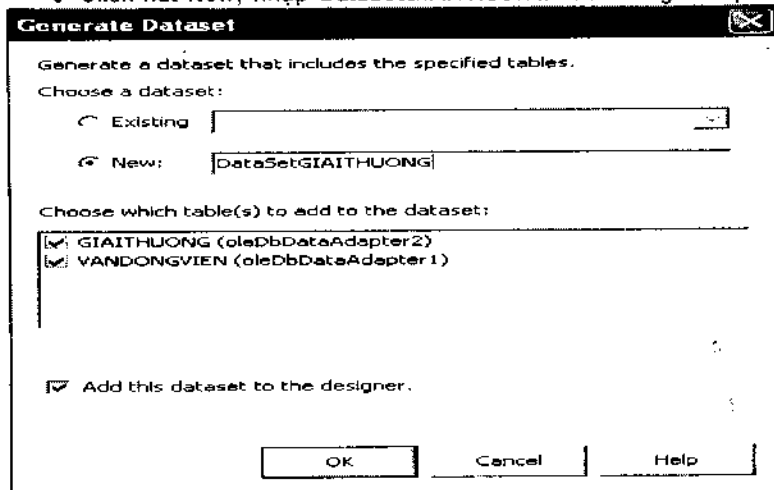
H142

◆ Click nút Finish, bạn thấy Dialog có thêm đối tượng OleDbDataAdapter2 xuất hiện trên khay công cụ như hình sau:



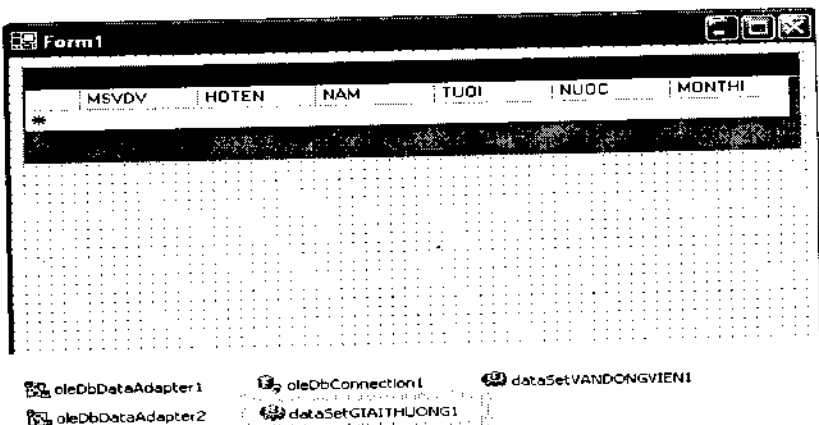
H143

- ◆ Click chọn form
- ◆ Vào menu Data, chọn Generate Dataset
- ◆ Click nút New, nhập DataSetGIAITHUONG vào khung bên phải



H144

◆ Click nút OK, bạn sẽ thấy đối tượng dataSetGIAITHUONG1 sẽ hiện ra trong khay công cụ



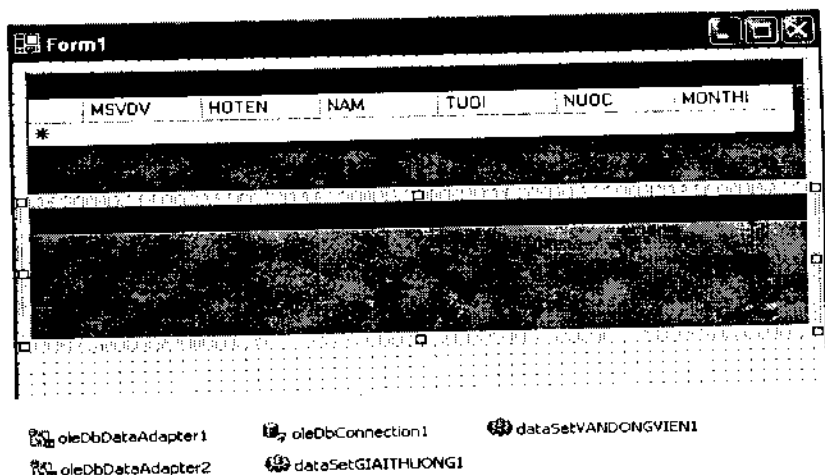
H145

Ghi chú:

Hai điều khiển lưới của hai bảng VANDONGVIEN và GIAITHUONG sẽ dùng chung đối tượng kết nối OleDbConnection1.

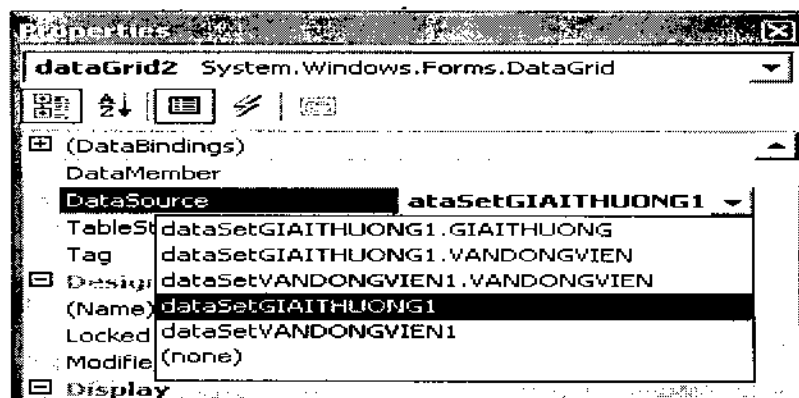
Tiếp theo là đưa điều khiển lưới DataGridView thứ hai vào Dialog để kết nối với bảng GIAITHUONG

- ◆ Click chọn tab có tên Windows Forms
- ◆ Click chọn điều khiển DataGridView
- ◆ Di chuyển con trỏ chuột vào form, vẽ để có dạng như hình sau:



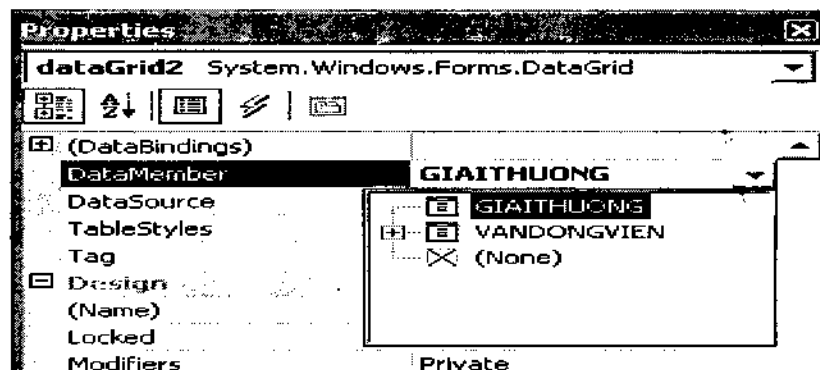
H146

- Click nút phải chuột trên điều khiển DataGrid, chọn Properties trong menu hiện ra, xác lập cho thuộc tính DataSource như sau:
- Click chọn khung bên phải của thuộc tính DataSource để xuất hiện đầu mũi tên ở góc bên phải
- Click vào đầu mũi tên, chọn dataSetGIAITHUONG1 trong danh sách bùng xuống.



H147

- Click chọn dataSetGIAITHUONG1
- Click chọn khung bên phải của thuộc tính DataMember để xuất hiện đầu mũi tên ở góc bên phải
- Click vào đầu mũi tên, chọn bảng GIAITHUONG trong khung bùng xuống



H148

Lúc này bạn sẽ có form như hình sau:

oleDbDataAdapter1 oleDbConnection1 dataSetVANDONGVIEN1
oleDbDataAdapter2 dataSetGIAITHUONG1

H149

Bạn đưa thêm hai điều khiển Button từ hộp công cụ vào trong form và xác lập các thuộc tính như trong bảng các thuộc tính đã cho ở trên.

Thực hiện xong, các bạn sẽ có hình như sau:

oleDbDataAdapter1 oleDbConnection1 dataSetVANDONGVIEN1
oleDbDataAdapter2 dataSetGIAITHUONG1

H150

Đến đây bạn xác lập các thuộc tính cho Form (bạn có thể xác lập đầu tiên, trước các điều khiển khác) bằng cách click nút phải chuột tại vị trí trống trên Form. Trong bảng hiện ra, chọn Properties, xác lập các thuộc tính của Form như đã cho trong bảng trên. Thực hiện xong, bạn sẽ thấy hình của Form như sau:

MSVDV	HOTEN	NAM	TUOI	NUOC	MONTHI
*					

MSGT	DIEM	HUYCH	VDVMS
*			

XEM THÔNG TIN
CẬP NHẬT

oleDbDataAdapter1

oleDbConnection1

dataSetVANDONGVIEN1

oleDbDataAdapter2

dataSetGIAITHUONG1

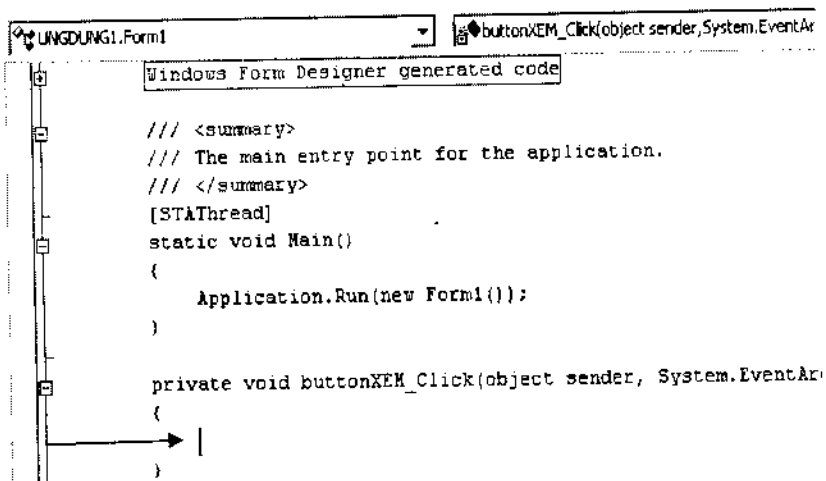
H151

Thực hiện tới đây, các bạn đã hoàn thành việc thiết kế Form cho chương trình.

② Viết và giải thích code cho chương trình

▣ Viết code cho ButtonXem

Nhấp đúp vào Button có tiêu đề "XEM THÔNG TIN", cửa sổ viết code hiện ra với tên thủ tục của ButtonXEM với sự kiện Click với cặp móc, bạn nhập các câu lệnh vào giữa hai cặp dấu móc này, bạn thấy cửa sổ viết code như hình sau:



H

Bạn viết code vào vị trí hình mũi tên trên để có được như sau:

private void buttonXEM_Click(object sender, System.EventArgs e)

i

```

dataSetVANDONGVIEN1.Clear();
dataSetGIAITHUONG1.Clear();
oleDbDataAdapter1.Fill(dataSetVANDONGVIEN1);
oleDbDataAdapter2.Fill(dataSetGIAITHUONG1);
}

```

Giải thích:

◆ Phương thức Clear() sẽ xóa nội dung dữ liệu trong Dataset mang tên dataSetVANDONGVIEN1 và dataSetGIAITHUONG1

◆ Phương thức Fill(dataSetVANDONGVIEN1) của oleDbDataAdapter1 sẽ đổ dữ liệu vào dataSetVANDONGVIEN1. Khi có dữ liệu, điều khiển lưới DataGrid1 sẽ thể hiện dữ liệu trên nó.

◆ Phương thức Fill(dataSetGIAITHUONG1) của oleDbDataAdapter2 sẽ đổ dữ liệu vào dataSetGIAITHUONG1. Khi có dữ liệu, điều khiển lưới DataGrid2 sẽ thể hiện dữ liệu trên nó.

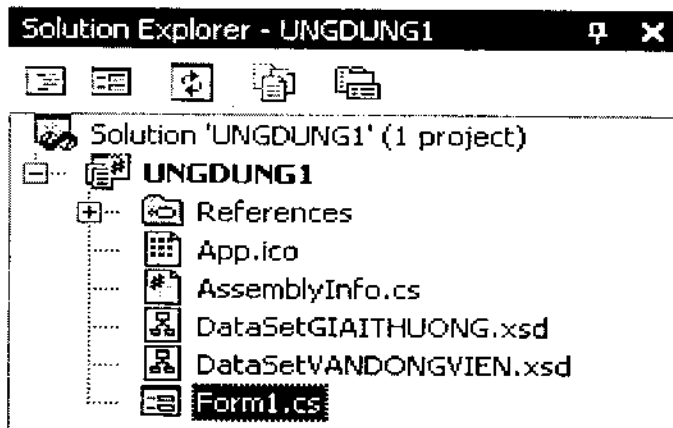
■ Viết code cho ButtonCapnhap

Nhấp đúp vào Button có tiêu đề "CẬP NHẬT", bạn nhập các câu lệnh vào giữa hai cặp dấu móc

Ghi chú:

Từ cửa sổ viết code, muốn thấy Form thiết kế, bạn có thể thực hiện như sau:

Click chọn Icon có tên Solution Explorer  trên thanh công cụ. Bạn sẽ thấy cửa sổ Solution Explorer hiện ra bên phải Form như hình sau:



H152

Click chọn mục Form1.cs, bạn sẽ thấy xuất hiện thêm hai Icon phía trên hộp thoại Solution Explorer như hình sau:

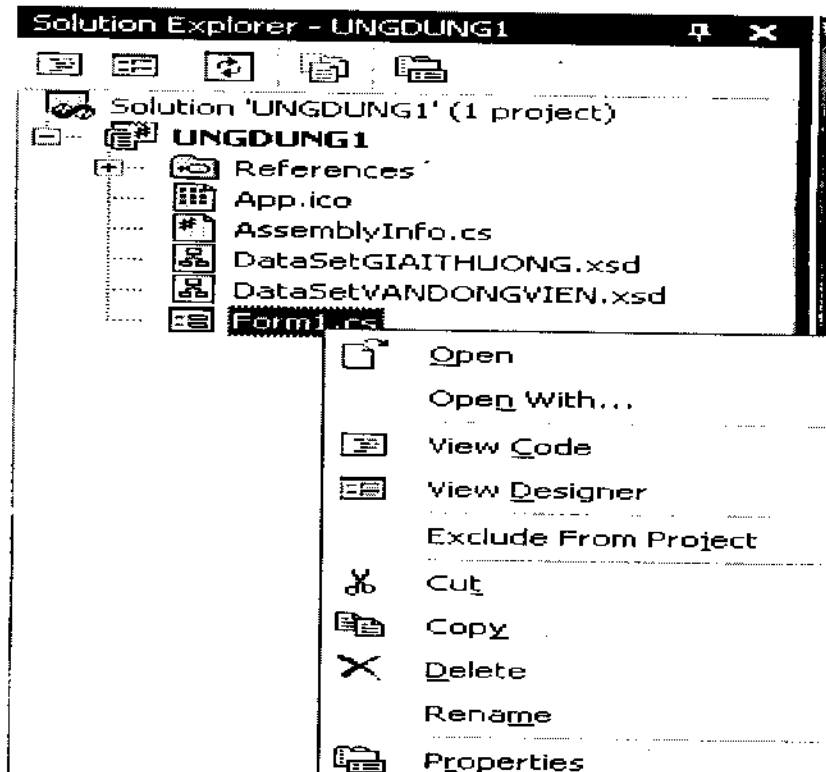
View Code

View Designer



H153

Bạn có thể click nút phải chuột trên tên Form1.cs, chọn View Code hay View Designer trong menu hiện ra



H154

```
private void buttonCAPNHAT_Click(object sender, System.EventArgs e)
{
    OleDbDataAdapter1.Update(dataSetVANDONGVIEN1);
    OleDbDataAdapter2.Update(dataSetGIAITHUONG1);
}
```

Giải thích:

◆ Hai câu lệnh trên nhờ phương thức Update sẽ cập nhật vào bảng những gì mà bạn sửa đổi trên điều khiển lưới.

Như vậy nếu sửa đổi dữ liệu trên điều khiển lưới mà không click vào nút có tiêu đề "CẬP NHẬT" thì thông tin vừa sửa sẽ không được cập

nhập vào bảng tương ứng. Đây là cơ chế an toàn mà Visual Studio .NET dùng cho cơ sở dữ liệu.

③ Chạy thử chương trình

Nhấn phím F5 để chạy thử chương trình, Nếu không có gì sai sót, bạn sẽ có hình như sau:

MSVDV	HOTEN	NAM	TUOI	NUOC	MONTHI
(null)	(null)	☐	(null)	(null)	(null)

MSGT	DIEM	HUYCH	VDVMS
*			

XEM THÔNG TIN **CẬP NHẬT**

H155

◆ Click vào Button có tiêu đề "XEM THÔNG TIN", bạn sẽ thấy kết quả như hình sau:

MSVDV	HOTEN	NAM	TUOI	NUOC	MONTHI
BB002	Kanjung	☒	21	Hàn Quốc	Bóng bà

MSGT	DIEM	HUYCH	VDVMS
002BB	19	Bạc	BB003

XEM THÔNG TIN **CẬP NHẬT**

H156

Bạn có thể di chuyển con trỏ dọc bên phải hai điều khiển lưới, bạn sẽ thấy các mẫu tin trong hai bảng VANDONGVIEN và GIAITHUONG.

Giả sử bạn di chuyển con trỏ dọc của điều khiển lưới bên trên, nhập thêm một vận động viên mới như sau:

DK013 Hoàng v 17 Việt Nam Điền kinh

Như hình sau:

CHƯƠNG TRÌNH VIỆT BANG C# (QUAN LY SEAGAMES23)

MSVDV	HOTEN	NAM	TUOI	NUOC	MONTH
DK013	Hoàng	☒	17	Việt Na	Điền kin

MSGT	DIEM	HUYCH	VDVMS
002BS	19	Bạc	BS010

XEM THÔNG TIN
CẬP NHẬT

H157

Nếu không click nút có tiêu đề "CẬP NHẬT" sau khi nhập thông tin vào điều khiển lưới, thì thông tin vừa nhập sẽ không được cập nhật vào bảng. Bạn có thể mở CSDL SEAGAME23.mdb trong Access để kiểm tra điều trên. Bạn cũng có thể kiểm tra bằng cách chạy lại chương trình, kéo con trỏ và nhìn thông tin trong điều khiển lưới, bạn sẽ không thấy mẫu tin vừa nhập.

Bây giờ bạn thử nhập vào một vận động viên khác, sau đó click nút "CẬP NHẬT" bạn sẽ thấy thông tin vừa nhập được lưu vào bảng.

BÀI 2

THEO DÕI KẾT QUẢ SEAGAMES 23 THIẾT KẾ BẢNG DATA FORM WIZARD

Trong bài trên, bạn thiết kế bằng Windows Application, công việc nhiều và kết quả chưa hay. Bài này, chúng tôi hướng dẫn các bạn dùng **Data Form Wizard** để triển khai ứng dụng cơ sở dữ liệu bằng cách sử dụng những khả năng cho nhìn xem, nhật tu và gỡ bỏ của công cụ này. Đây có thể là cách nhanh nhất để có thể triển khai những ứng dụng cơ sở dữ liệu trên Visual Studio .NET.

Khi dùng Data Form Wizard để viết một ứng dụng, nó sẽ cho bạn một chương trình thực hiện với đầy đủ các chức năng như chèn, nhật tu và gỡ bỏ dữ liệu mà không cần phải viết một dòng lệnh nào.

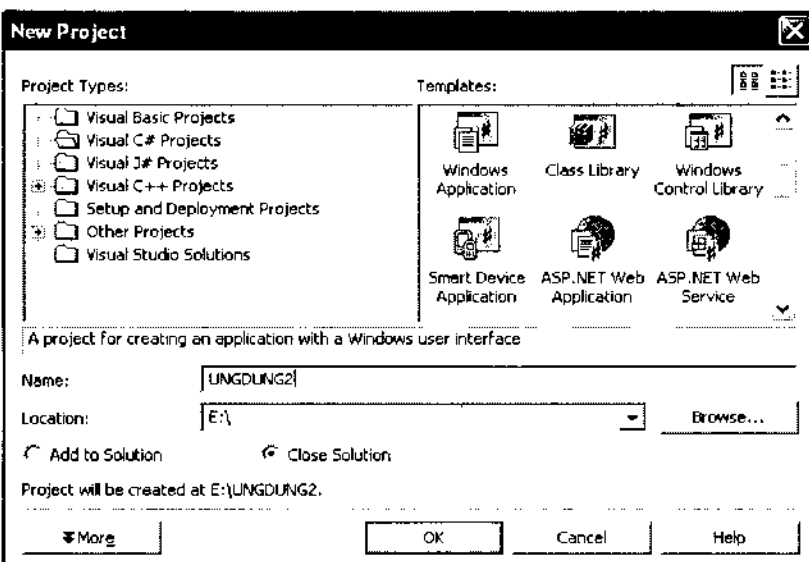
Chúng tôi vẫn dùng lại CSDL SEAGAMES23.mdb để giới thiệu cho cách thiết kế này.

Bạn tạo một ứng dụng Windows Forms mới bằng cách:

◆ Vào menu **File**, di chuyển vệt sáng tới **New**, chọn **Project**.

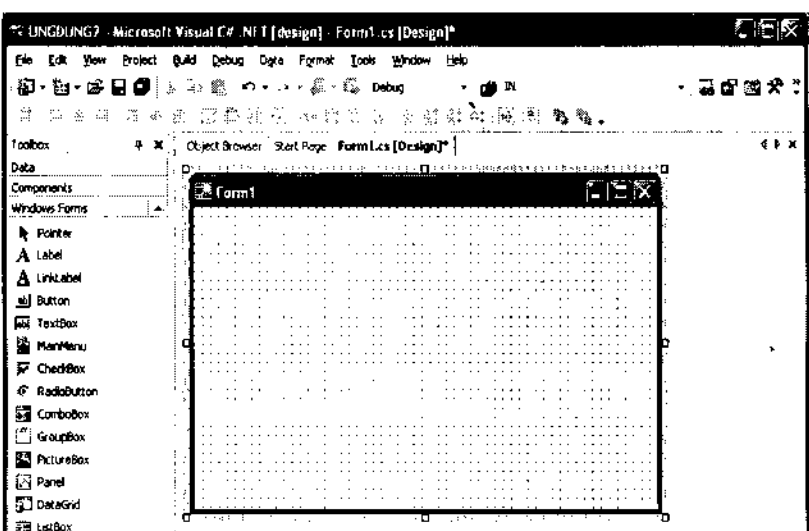
◆ Trong hộp thoại hiện ra, chọn **Visual C# Projects** trong khung **Project Types**. Chọn **Windows Application** trên **Templates**.

◆ Đặt tên cho ứng dụng chẳng hạn UNGDUNG2 như hình sau:



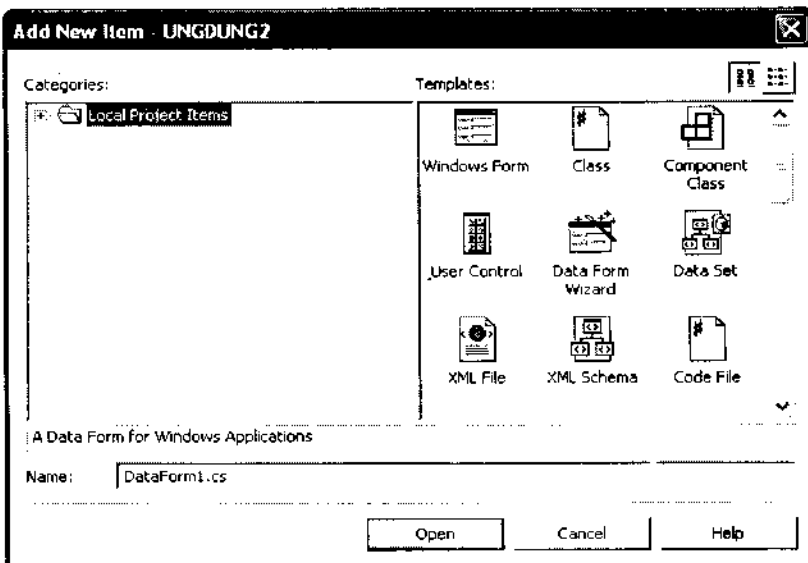
H158

- ◆ Click nút **OK**
- ◆ Cửa sổ sau hiện ra:



H159

Từ cửa sổ này bạn thêm một Data Form Wizard bằng cách:
Vào menu Project, chọn Add New Item, hộp thoại sau hiện ra



H160

◆ Bạn chọn **Data Form Wizard** từ **template** có sẵn như hình trên. Bạn có thể đặt lại tên của lớp Data Form vào khung Name nếu như không thích tên này. Ở đây chúng tôi vẫn giữ nguyên.

◆ Click nút **Open**, bạn thấy hình sau:



H161

◆ Click nút **Next**, chúng ta được

Data Form Wizard

Choose the dataset you want to use

The dataset contains the tables and columns to work with in your form.



The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

☐ Use this following dataset:

Cancel

< Back

Next >

Finish

H162

Bạn đặt tên cho dataset để truy xuất dữ liệu, giả sử nhập tên SEAGAMES23 vào khung trắng.

Data Form Wizard

Choose the dataset you want to use

The dataset contains the tables and columns to work with in your form.



The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

SEAGAMES23

☐ Use this following dataset:

Cancel

< Back

Next >

Finish

H163

◆ Click nút **Next**, ta sẽ thấy hộp thoại kết nối hiện ra, chúng ta chọn cơ sở dữ liệu có tên **SEAGAMES23.mdb** trong ổ đĩa E để có hình như sau:

Data Form Wizard

Choose a data connection

The data connection specifies how to find and log into a specific server and database.



Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.

Which connection should the wizard use?

ACCESS E:\SEAGAMES23.mdb Admin

New Connection...

Cancel

< Back

Next >

Finish

H164

Ghi chú:

Do bài trước chúng ta đã kết nối với cơ sở dữ liệu SEAGAMES23.mdb nên bây giờ đã có sẵn. Nếu chưa có, bạn thấy hộp thoại kết nối hiện ra, bạn thực hiện các bước kết nối như ở bài 1. Sau khi kết nối xong bạn thực hiện tiếp các mục sau:

◆ Click nút **Next**, bạn thấy hộp thoại hiện ra cho chúng ta chọn các **Tables** hoặc **Views** mà bạn muốn kết nối về **dataset**.

Data Form Wizard

Choose tables or views

The tables or views you choose will determine which columns will be available to display on your form.

The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

- Tables
 - GIAITHUONG
 - VANDONGVIEN
- Views

Selected item(s):

- Tables
- Views

Cancel < Back Next > Finish

H165

◆ Trong hộp thoại này, chúng tôi chọn bảng **GIAITHUONG**, trên khung **Available item(s)** sau đó click nút có dấu mũi tên hướng qua khung bên phải ">" để chuyển qua khung **Selected item(s)**. Sau đó chọn bảng **VANDONGVIEN** và đưa qua khung **Selected item(s)** như trên. Như vậy trên khung **Selected items** chúng tôi đã chọn hai bảng dữ liệu: **GIAITHUONG** và **VANDONGVIEN**.

Data Form Wizard

Choose tables or views
The tables or views you choose will determine which columns will be available to display on your form.

The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

- Tables
- Views

Selected item(s):

- Tables
 - GIAITHUONG
 - VANDONGVIEN
- Views

Buttons: Cancel, < Back, Next >, Finish

H166

Bây giờ, bạn đã sẵn sàng tạo ra một mối liên hệ giữa hai bảng dữ liệu này. Bạn nhấn nút **Next** để bước qua màn hình kế tiếp.

Data Form Wizard

Create a relationship between tables
The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.

Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name: _____

Parent table: _____

Child table: _____

Keys:

Buttons: Cancel, < Back, Next >, Finish

Màn hình này cho phép bạn tạo một mối liên hệ giữa VANDONGVIEN và GIAITHUONG.

Giả sử bạn nhập TAOMOILIEHENVANDONGVIENVAGIAITHUONG trong khung Name

◆ Click đầu mũi tên quay xuống của khung **Parent table**, chọn bảng **VANDONGVIEN** (bảng cha mẹ)

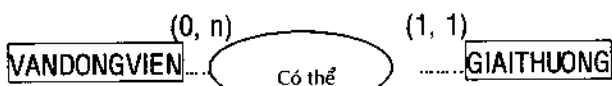
Lúc này bạn thấy bảng **GIAITHUONG** tự động xuất hiện bên khung **Child table** (bảng con cái)

◆ Chọn khóa của bảng VANDONGVIEN và GIAITHUONG ở hai khung tương ứng bên dưới.

Bạn phải tạo một mối liên hệ một nhiều trên cơ sở dữ liệu. Nói cách khác:

- ☛ Một vận động viên có thể không được giải thưởng nào hoặc được nhiều giải thưởng.
- ☛ Một giải thưởng nào đó thì có một và chỉ một vận động viên được nhận mà thôi.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:



Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ GIAITHUONG qua VANDONGVIEN.

Khi có mối liên hệ một nhiều giữa hai table thì bạn nhớ là khoá ngoại sẽ nằm trong table có giá trị là (0, 1) hoặc (1, 1).

◆ Bạn click nút ">" báo cho Wizard biết là bạn chuyển qua ngụy tại ô **Relation** phía tay phải, các ô khác phía tay trái trở thành rỗng.

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.

Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

Parent table:

Child table:

Keys:

Relations:

TAOMOILIEHVEVANDON

Cancel

< Back

Next >

Finish

H168

◆ Click nút Next để qua màn hình kế tiếp như sau:

Data Form Wizard

Choose tables and columns to display on the form

Your form can display any of the tables and columns available in the dataset.

If you display more than one table on the form, the tables have a master-detail relationship

What tables and columns do you want to display on the form?

Master or single table:

GIATHUONG

Detail table:

Columns:

- ☒ DIEM
- ☒ HUYCHUONG
- ☒ MSGT
- ☒ YDVM5VDV

Columns:

Cancel

< Back

Next >

Finish

H169

◆ Click chọn đầu mũi tên quay xuống, chọn bảng VANDONGVIEN.
Bạn sẽ thấy màn hình sau:

Data Form Wizard

Choose tables and columns to display on the form

Your form can display any of the tables and columns available in the dataset.



If you display more than one table on the form, the tables have a master-detail relationship

What tables and columns do you want to display on the form?

Master or single table:

VANDONGVIEN

Detail table:

GIAITHUONG

Columns:

- ☒ HOTEN
- ☒ MONTHI
- ☒ MSVDV
- ☒ NAM
- ☒ NUOC
- ☒ TUOI

Columns:

- ☒ DIEM
- ☒ HUYNHCHUONG
- ☒ MSGT
- ☒ VDVMSVDV

Cancel

< Back

Next >

Finish

H170

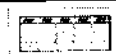
Màn hình này cho phép bạn chọn bảng dữ liệu nào, cột nào bạn muốn cho hiển thị lên biểu mẫu. Trong ví dụ này, chúng ta chọn tất cả các cột của cả hai bảng dữ liệu.

◆ Click nút Next để qua màn hình kế tiếp

Data Form Wizard

Choose the display style

You can display a single record at a time or display all records at once.



How do you want to display your data?

- ☒ All records in a grid
- ☐ Single record in individual controls

What additional controls do you want on the form?

☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

- ☐ Add - (Adds a new record)
- ☐ Delete - (Deletes the current record)
- ☐ Cancel - Cancels changes to the current record
- ☐ Navigation controls - Moves to first, previous, next, or last record

The wizard now has the information it needs. Click Finish to exit and generate your new form.

Cancel

< Back

Next >

Finish

H171

Màn hình này rất quan trọng, đây là trang tạo biểu mẫu của bạn. Hiện thời Data Form Wizard thêm một biểu mẫu Windows với vài ô control trên biểu mẫu và viết đoạn mã để điền, nhập tu, xóa sổ và lèo lái giữa dòng dữ liệu.

Có hai cách nhìn xem dữ liệu. Chúng ta có thể chọn một trong hai cách này, đó là:

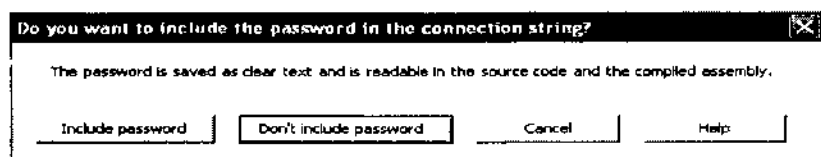
- ◆ All records in a grid.
- ◆ Single record in individual control.

Nếu chọn All records in a grid. Thì tất cả các mẫu tin sẽ được ghi lên một khung lưới.

Nếu chọn Single record in individual control. Thì mỗi mẫu tin duy nhất trên những ô control riêng rẽ.

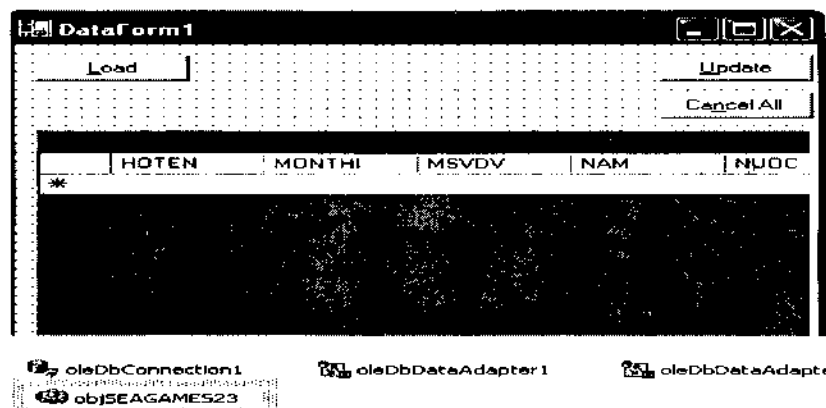
Tới đây giả sử bạn chọn mục All records in a grid.

- ◆ Click nút Finish. Bạn thấy hộp thoại sau hiện ra:



H172

◆ Click nút **Don't include password**. Bạn thấy biểu mẫu do Data Form Wizard thiết kế khi chọn mục trên. Bạn thấy Form có dạng như hình sau:



H173

Tới đây bạn đã hoàn thành việc thiết kế biểu mẫu cho ứng dụng. Tiếp theo bạn phải triệu gọi Data Form Wizard từ ứng dụng.

- ◆ Bạn cần triệu gọi DataForm1 khi bạn bắt đầu ứng dụng. Theo mặc nhiên, ứng dụng của bạn triệu gọi biểu mẫu Form1 khi khởi động.

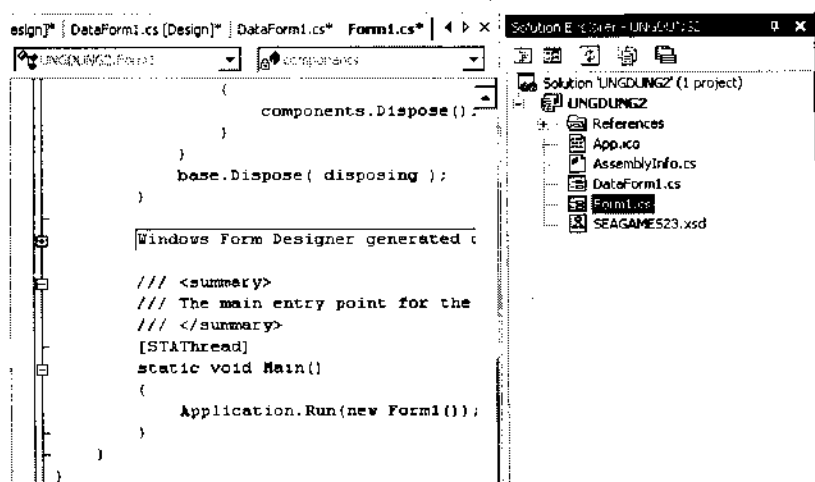
- ◆ Bạn thay đổi theo các bước như sau:

- ◆ Vào cửa sổ Solution Explorer.

- ◆ Click chọn Form1

- ◆ Click nút View code nằm phía trên trong cửa sổ Solution Explorer

- ◆ Di chuyển đến thủ tục static void Main() như hình sau:



H174

Sửa lại như sau:

Thay vì:

```
static void Main()
```

```
{
    Application.Run(new Form1());
}
```

Các bạn thay lại như sau:

```
static void Main()
{
    Application.Run(new DataForm1());
}
```

Nhấn phím F5 để chạy chương trình, bạn thấy kết quả hiện ra như hình sau:

The screenshot shows a window titled "DataForm1" with a "Load" button on the left and "Update" and "Cancel All" buttons on the right. The form contains five input fields: "HỌTÊN", "MONTHI", "MSVDV", "NAM", and "NƯỚC". Below these fields is a large, dark, noisy area representing a list of data.

H175

Click nút Load trên biểu mẫu, bạn thấy kết quả như hình sau:

The screenshot shows a window titled "DataForm1" with a "Load" button on the left and "Update" and "Cancel All" buttons on the right. The form contains a table with the following columns: "HỌTÊN", "MONTHI", "MSVDV", "NAM", and "NƯỚC". The table contains 10 rows of data. Below the table is a section with fields: "DIEM", "HUYCHUON", "MSGT", and "VDVMSVDV".

HỌTÊN	MONTHI	MSVDV	NAM	NƯỚC
Nguyễn Anh	Buồng bẻn	BB002		Hai
Nguyễn Hoa	Buồng bẻn	BB003		Viea
Winza	Buồng roã	BR004		Viea
Mai Quyát T	Niên kinh	DK005		Mã
Nguyễn Thò	Niên kinh	DK006		Viea
Hanchunzu	Niên kinh	DK007		Viea
Nguyễn Vae	Baen suing	BS008		Hai
Kakazu	Baen suing	BS010		Viea
KKK	Niên kinh	DK009		Hai
BBB	Niên kinh	DK010		Trun

Below the table, the fields are: "DIEM" (18), "HUYCHUON" (Noang), "MSGT" (00388), and "VDVMSVDV" (BB002).

H176

Bạn thấy trên trên điều khiển lưới của bảng VANDONGVIEN hiện ra tất cả mẫu tin, mẫu tin hiện hành có MSVDV là BB002, do tạo mới

liên hệ giữa hai bảng VANDONGVIEN và GIAITHUONG qua khóa chính là MSVDV của bảng VANDONGVIEN và khóa ngoại VDVMSVDV của bảng GIAITHUONG, nên bạn thấy trong bảng GIAITHUONG mẫu tin có VDVMSVDV là BB002 sẽ hiện ra.

Bạn thử click chuột chọn vận động viên có MSVDV là DK009 trong điều khiển lưới của bảng VANDONGVIEN, bạn sẽ thấy kết quả như hình sau:

HỌTÊN	MONTHI	MSVDV	NAM	NƯỚC
Kanlung	Boung baen	BB002		Hai
Nguyen Anh	Boung baen	BB003		Viet
Nguyen Hoa	Boung ro	BR004		Viet
Winza	Boung ro	BR005		Ma
Mai Quyet T	Nien kinh	DK006		Viet
Nguyen Tho	Nien kinh	DK007		Viet
Hanchunzu	Nien kinh	DK008		Hai
Nguyen Va	Beon suong	BS009		Viet
Kakazu	Beon suong	BS010		Hai
KKK	Nien kinh	DK009		Trun
BBB	Nien kinh	DK010		Viet

DIEM	HUYCHUON	MSGT	VDVMSVDV
17	Noang	0090K	DK009

H177

Kết quả trên điều khiển lưới của bảng GIAITHUONG bên dưới sẽ xuất hiện mẫu tin có VDVMSVDV là DK009.

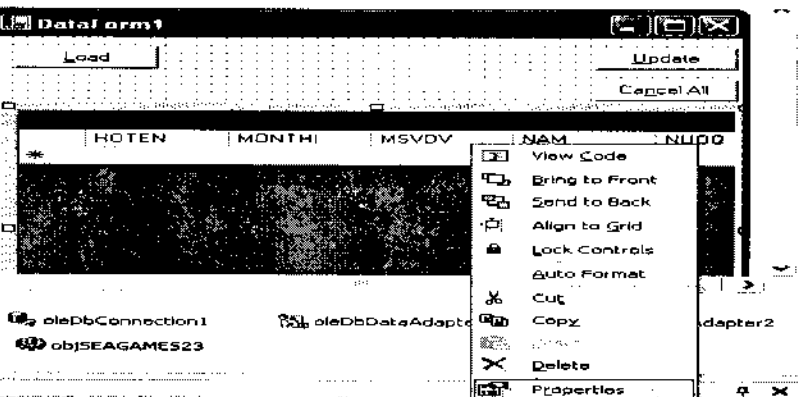
Nhìn vào hai điều khiển lưới ở hình trên, các bạn thấy font chữ không phải là font VNI. Để thấy được tiếng Việt, bạn thực hiện như sau:

◆ Cho biểu mẫu của chương trình hiện ra (cho cửa sổ solution Explorer hiện ra, click chọn tập tin có tên DataForm1, click biểu tượng View Design phía trên)

◆ Click chọn điều khiển lưới thứ nhất ở phía trên.

◆ Click nút phải chuột trên điều khiển lưới này

◆ Trong menu hiện ra, click chọn Properties như hình sau:



H178

◆ Chọn thuộc tính Font trong bảng Properties, chọn Font VNI-Times, với size 12.

◆ Di chuyển con trỏ dọc để thấy điều khiển lưới thứ hai bên dưới.

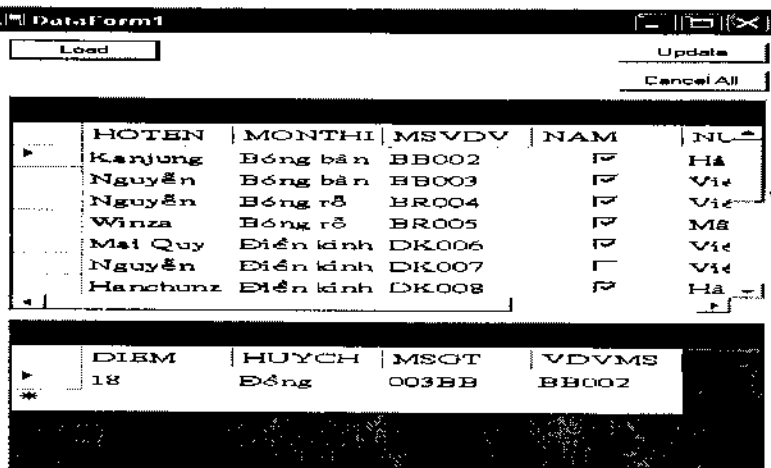
◆ Click chọn điều khiển này

◆ Click nút phải chuột trên điều khiển

◆ Chọn Properties trong menu hiện ra

◆ Chọn Font VNI-Times với size là 12

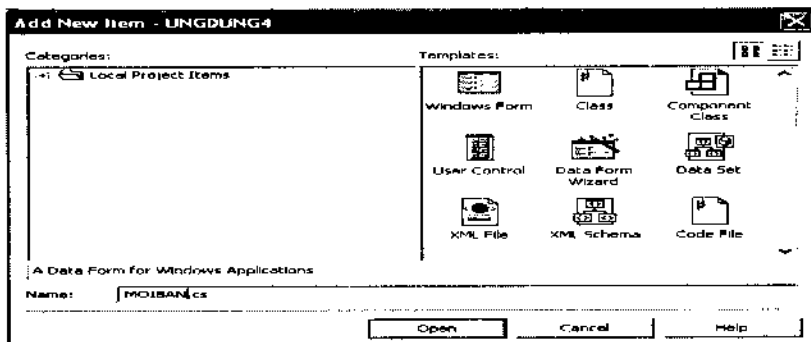
Nhấn phím F5 chạy thử chương trình, click nút Load, bạn thấy kết quả bây giờ như hình sau:



H179

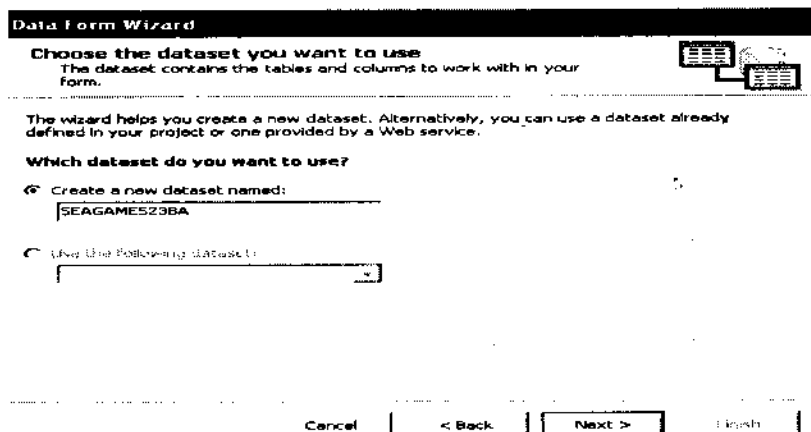
Bây giờ bạn thử thực tập lại bằng cách tạo một chương trình quản lý seagames23 theo cách chọn thứ hai trong trang cuối cùng của Data Form Wizard đó là Single record in individual controls.

- ◆ Vào Visual Studio .NET
- ◆ Tạo một Project có tên UNGDUNG4 bằng Visual C# Project và Windows Application.
- ◆ Vào menu Project, chọn Add New Item
- ◆ Chọn Data Form Wizard ở khung bên phải của cửa sổ hiện ra



H180

- ◆ Đặt tên trong khung Name là MOIBAN.cs
- ◆ Click nút Open
- ◆ Hộp thoại hiện ra, click nút Next



H181

◆ Đặt tên cho Dataset là SEAGAMES23BA

◆ Click nút Next

Thực hiện các bước tiếp theo giống như bài trên cho đến khi hiện ra trang cuối cùng.

Ở trang này thay vì chọn mục All records in a grid, bạn chọn mục **Single record in individual control** như hình sau:

Data Form Wizard

Choose the display style
You can display a single record at a time or display all records at once.

How do you want to display your data?

☐ All records in a grid

☒ Single record in individual controls

What additional controls do you want on the form?

☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

☒ Add - Creates a new record.

☒ Delete - Deletes the current record.

☒ Cancel - Cancels changes to the current record.

☒ Navigation controls - Moves to first, previous, next, or last record.

The wizard now has the information it needs. Click Finish to exit and generate your new form.

Cancel < Back Next > Finish

H182

Khi chọn mục hai này, các bạn thấy các dấu chọn bên dưới đậm lên, nếu muốn bỏ mục nào, các bạn có thể click để bỏ. Ở đây chúng ta vẫn chọn đầy đủ.

◆ Click nút Finish, bạn sẽ có kết quả như hình sau:

H183

Bạn thấy có nhiều nút hiện ra, nút Load để tải thông tin trong các bảng hiện ra trong các TextBox và điều khiển lưới, nút Update để cập nhật những điều vừa chỉnh sửa hoặc mới thêm, nút Add để thêm mẫu tin mới, nút Delete để xóa mẫu tin, nút xem mẫu tin kế tiếp, mẫu tin trước đó, mẫu tin đầu, mẫu tin cuối ... thông tin của các vùng trong bảng VANDONGVIEN được thể hiện trong các TextBox và tựa của các vùng sẽ hiện trong các Label. Thông tin của bảng GIAITHUONG sẽ hiện ra trong điều khiển lưới (khác với cách 1, biểu mẫu sẽ thể hiện hai điều khiển lưới, số nút ít hơn).

Tiếp theo là sửa lại tên biểu mẫu, trong bài này thay vì gọi Form1, bạn sửa lại là MOIBAN.

◆ Bạn vào cửa sổ Solution Explorer, chọn Form1, click nút View Code để sửa lại.

```
static void Main()
{
    Application.Run(new Form1());
}
```

Các bạn thay lại như sau:

```
static void Main()
{
    Application.Run(new MOIBAN());
}
```


Nhấn phím F5 để chạy chương trình, bạn thấy kết quả hiện ra như hình sau:

The screenshot shows the MOIBAN application window. It has a title bar with standard Windows controls. Inside, there's a 'Load' button on the left and 'Update' and 'Cancel All' buttons on the right. Below these are three input fields: 'HOTEN' (Name), 'MONTHI' (Month), and 'MSVDV' (ID). To the right of these are labels 'NAM' (Year), 'NUOC' (Country), and 'TUOI' (Age). Below the input fields are navigation buttons: '<<', '<', '>', '>>', 'Add', 'Delete', and 'Cancel'. At the bottom, there's a table with columns: DIEM, HUYCHUON, MSGT, and VDVMSVDV. The table is currently empty, showing only a header row and a row with an asterisk in the first column.

H184

◆ Click nút Load, bạn thấy màn hình như sau:

The screenshot shows the MOIBAN application window after clicking the 'Load' button. The input fields are now populated: 'HOTEN' is 'Kanjung', 'MONTHI' is 'Boong ban', and 'MSVDV' is 'BB002'. The labels 'NAM', 'NUOC', and 'TUOI' are now '21', 'Haen Quoc', and '21' respectively. The navigation buttons now show '1 of 13' in the center. The table at the bottom now contains one record: DIEM: 18, HUYCHUON: Noang, MSGT: 003BB, VDVMSVDV: BB002. The table has a header row and a row with an asterisk in the first column.

H185

Bạn thấy mẫu tin thứ nhất của vận động viên có mã số BB002 thuộc bảng VANDONGVIEN hiện ra trong các TextBox, đây là mẫu tin thứ 1 trên tổng số 13 mẫu tin có trong bảng.

Trong điều khiển lưới sẽ hiện ra mẫu tin tương ứng của bảng GIAITHUONG có mã số vận động viên là BB002 (do chúng ta tạo mối quan hệ).

◆ Thử click nút >> đến đến mẫu tin cuối cùng của bảng VANDONGVIEN, bạn sẽ có kết quả như hình sau:

MOIBAN

Load Update Cancel All

HOTEN: Hoàng NAM: ☒

MONTHI: Niền kinh NUOC: Việt Nam

MSVDV: DK013 TUOI: 17

<< < 13 of 13 > >>

Add Delete Cancel

DIEM	HUYCHUON	MSGT	VDVMSVDV
*			

H186

Bạn thấy vận động viên có mã số DK013 không đạt huy chương. Thử click nút < để xem mẫu tin liền trước, tức là mẫu tin thứ 12/13, bạn thấy kết quả như hình sau:

MOIBAN

Load Update Cancel All

HOTEN: Phan Nhật Nam NAM: ☒

MONTHI: Niền kinh NUOC: Việt Nam

MSVDV: DK011 TUOI: 17

<< < 12 of 13 > >>

Add Delete Cancel

DIEM	HUYCHUON	MSGT	VDVMSVDV
17	Vàng	004DK	DK011

H187

Bạn thấy vận động viên có mã số DK011 đạt huy chương đồng.

Để có thông tin bằng tiếng Việt trong các TextBox cũng như trong các điều khiển lưới, thay vì phải chọn từng TextBox và điều khiển lưới để xác lập thuộc tính Font theo kiểu tiếng Việt. Bạn có thể thực hiện như sau:

- ◆ Cho biểu mẫu của chương trình hiện ra
- ◆ Click nút phải chuột tại vị trí trống trên Form
- ◆ Chọn Properties trong menu hiện ra
- ◆ Giả sử chọn Font VNI-Times có size 12

Nhấn phím F5 để chạy thử chương trình, click nút Load, bạn sẽ có kết quả như hình sau:

MOIBAN

Load Update Cancel

HOTEN Kanjung NAM ☒

MONTHI Bống bàn NUOC Hàn Quốc

MSVDV BB002 TUOI 21

<< < 1 of 13 > >>

Add Delete Cancel

	DIEM	HUYCH	MSGT	VDVMS
▶	18	Đồng	003BB	BB002
*				

BÀI 3

QUẢN LÝ SINH VIÊN THIẾT KẾ BẢNG DATA FORM WIZARD

Để thực hiện được chương trình này, đầu tiên chúng ta phải tạo một cơ sở dữ liệu gồm các bảng. Các bạn có thể tạo trong SQL Server, hoặc trong Oracle hoặc trong Access, nếu tạo ở CSDL nào thì khi kết nối, bạn phải kết hợp với bộ máy thích hợp.

Bạn vào Access, tạo một Database có tên QLSINHVIEN.mdb gồm 4 bảng đó là: SINHVIEN, MONHOC, KHOA và KETQUA. Nội dung của 4 bảng có các vùng như sau:

Các vùng có gạch dưới là vùng khóa, trong bảng SINHVIEN khóa chính là MSSV, bảng KHOA vùng khóa là Makhoa, bảng KETQUA cả ba vùng Mamh, Lanthi và Masv là khóa, bảng MONHOC vùng khóa là Mamh.

SINHVIEN

Field Name	Data Type	Description
<u>MSSV</u>	Text (5)	Mã số sinh viên
Holot	Text (30)	Họ lót
Ten	Text (7)	Tên
Nam	Yes/No	Nam
Ngaysinh	Date/Time	Ngày sinh
Noisinh	Text (30)	Nơi sinh
Diachi	Text (30)	Địa chỉ
TinhTP	Text (20)	Tỉnh Thành phố
Hocbong	Number (Long Integer)	Học bổng
Lylich	Memo	Lý lịch
Khoamakh	Text (5)	Mã khoa

KHOA

Field Name	Data Type	Description
<u>Makhoe</u>	Text (5)	Mã khoa
Tenkhoe	Text (20)	Tên khoa

KETQUA

Field Name	Data Type	Description
<u>Mamh</u>	Text (5)	Mã môn học
Lanthi	Number (Integer)	Lần thi
Masv	Text (5)	Mã số sinh viên
Diem	Number (Integer)	Điểm

MONHOC

Field Name	Data Type	Description
<u>Mamh</u>	Text (5)	Mã môn học
Tenmh	Text (20)	Tên môn học

Bạn tạo cấu trúc cho các bảng như trên, cách tạo xem bài 1.

Microsoft Access

File Edit View Insert Tools Window Help

OLSVINHVIEN Database (Access 2000) File Edit View Database Window Help

Table1 : Table

Field Name	Data Type	
MSSV	Text	Mã số sinh viên
Holot	Text	Họ lót
Ten	Text	Tên
Nam	Yes/No	Nam
Ngaysinh	Date/Time	Ngày sinh
Noisinh	Text	Nơi sinh
Diachi	Text	Địa chỉ
TinhTP	Text	Tỉnh Thành phố

Field Properties

Tạo xong, bạn click chọn mẫu tin là vùng khóa, click hình chiếc chìa khóa trên thanh công cụ. Khi click xong, bạn sẽ thấy hình chiếc *chìa khóa* nằm trước **MSSV** về phía bên trái. Nếu click hình chiếc chìa khóa trên thanh công cụ lần nữa, sẽ thôi chọn vùng khóa (hình chiếc chìa khóa nằm phía trước MSSV sẽ biến mất).

☐ Click dấu **X** để đóng bảng, hộp thoại hiện ra hỏi bạn có lưu bảng vừa nhập không?

Save As

Table Name:

H190

☐ Bạn nhập **SINHVIEN** vào khung **Table Name** và click nút **OK**, tập tin **SINHVIEN** sẽ lưu vào **CSDL QLSINHVIEN**.

☐ Click nút **Design**, để tạo cấu trúc cho bảng kế tiếp

Microsoft Access

File Edit View Insert Tools Window Help

QLSINHVIEN - Database - Access 2000 file format

Table: Table

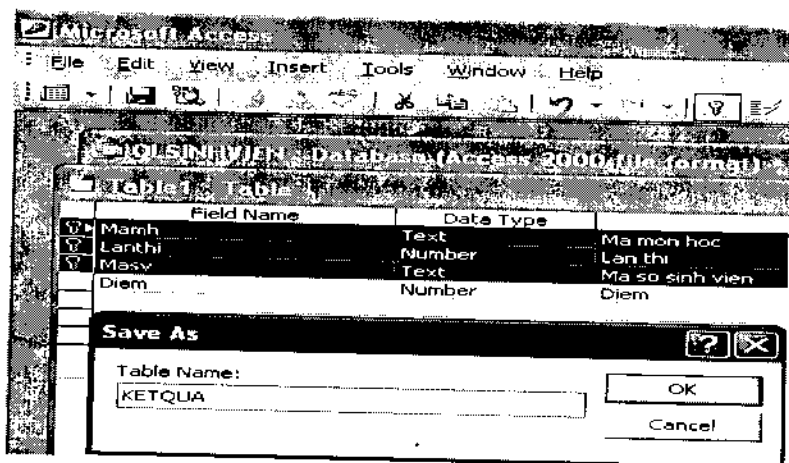
Field Name	Data Type	
Makhoa	Text	Ma khoa
Tenkhoe	Text	Ten khoa

Save As

Table Name:

H191 Tạo và lưu bảng KHOA

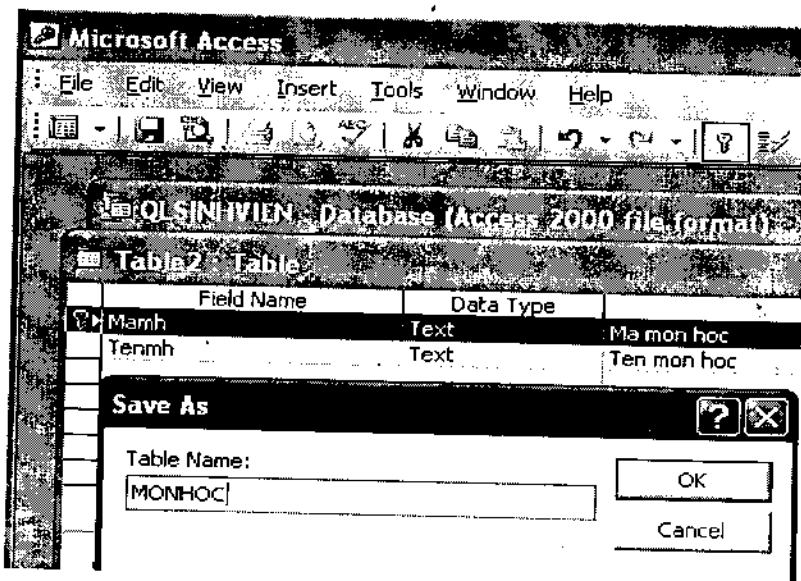
☐ Tiếp tục click nút Design để tạo bảng kế tiếp



H192

☐ Trong bảng **KETQUA**, bạn nhấn phím **Shift** và chọn 3 vùng như trên, click hình chìa khóa trên thanh công cụ, bạn sẽ được như hình trên. Lưu tập tin với tên **KETQUA**

☐ Tiếp tục chọn **Design** trong **CSDL** để tạo bảng cuối cùng



H193

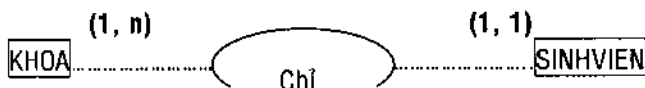
▣ Tạo và lưu bảng **MONHOC** như trên.

Suy luận:

☞ Một sinh viên thì chỉ học ở một khoa mà thôi.

☞ Một khoa thì sẽ có 1 hoặc nhiều sinh viên theo học.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:

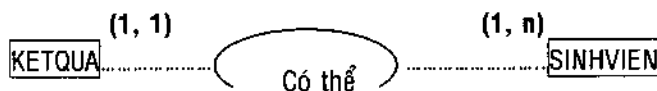


Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ **SINHVIEN** qua **KHOA**.

☞ Một sinh viên thì có 1 hoặc nhiều kết quả (nhiều môn).

☞ Một kết quả thì chỉ của một sinh viên mà thôi.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:

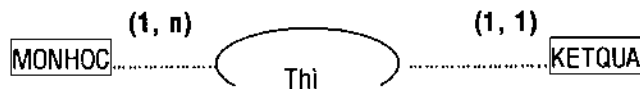


Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ **KẾT QUẢ** qua **SINHVIEN**.

☞ Một môn học thì có một hoặc nhiều kết quả.

☞ Một điểm kết quả thì chỉ thuộc một môn học.

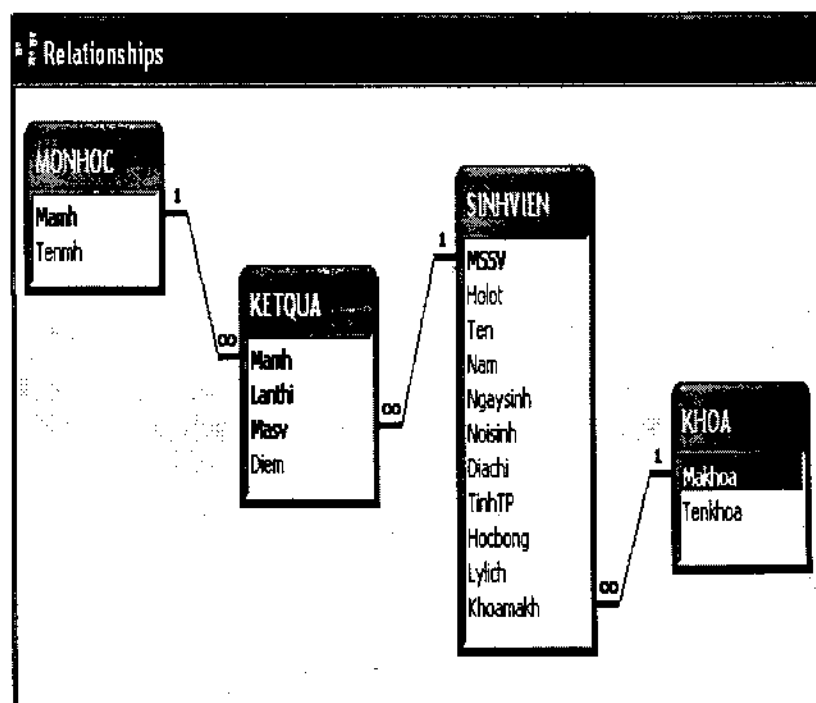
Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:



Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ KETQUA qua MONHOC.

Với suy luận như trên, ta sẽ tạo mối liên kết dễ dàng, drag chuột từ khoá chính (bên bảng có (1, n) hoặc (0, n)) sang tên của khoá ngoại có cùng giá trị với khoá chính (tên có thể khác) ở bảng có (1, 1).

Click biểu tượng Relationships trên thanh công cụ để bắt đầu tạo mối liên hệ, đưa các bảng vào cửa sổ, tạo liên kết giữa các bảng, nếu quên xem lại bài 1, bạn sẽ được như hình sau:



H194

Sau khi đã tạo các mối liên hệ như trên, bạn nhập thông tin vào cho 4 bảng. Đầu tiên bạn nhập cho bảng KHOA, kế đến nhập bảng SINHVIEN, kế đến nhập bảng MONHOC và sau cùng nhập bảng KETQUA. Bạn có thể theo thứ tự khác một chút, nhưng lưu ý phải có khoa trước thì mới có sinh viên đăng ký theo học, phải có môn học và sinh viên thì mới có kết quả.

Giải sử nội dung của các bảng như sau:

☉ Bảng **KHOA**

KHOA : Table			
		Makhoa	Tenkhoa
▶	+	TH	Tin học
	+	TT	Toán tin
	+	VL	Vật lý
*			

H195

Trong trường có rất nhiều khoa, ở đây chúng tôi chỉ nhập các khoa có mã khoa là TH (Tin học), TT (Toán tin), và VL (Vật lý). Bạn lưu ý trong Bảng SINHVIEN có vùng là mã khoa, bạn phải nhập mã có trong bảng KHOA, nếu nhập sai Access sẽ báo lỗi (vì bạn đã tạo mối liên hệ giữa hai bảng KHOA và SINHVIEN)

© Bảng SINHVIEN

SINHVIEN : Table							
	MSSV	Holot	Ten	Nam	Ngaysinh	Noisinh	Diachi
▶	TH001	Hoàng	Anh	☒	12/3/1987	TP.HCM	P24 Q10
	TH002	Minh	Bình	☒	3/24/1987	Hà Nội	P26 QBT
	TH003	Khánh	Hiền	☐	5/23/1986	Bến Tre	Ký túc xá ĐH
	TT001	Minh	Nam	☒	7/12/1987	TP.HCM	P1 Q10
	TT002	Hoàng	Phát	☒	8/12/1987	Bình Dương	234 P1
	VL001	Minh	Anh	☐	4/12/1987	Quảng Ngãi	P25 QBT
	VL002	Phan Nhật	Nam	☒	5/13/1987	TP.HCM	P12 Q1
*				☐			

H196

Trong bảng này, mã số sinh viên nếu là khoa Tin học chúng tôi dùng hai kí tự đầu để chỉ khoa, ba kí tự sau để chỉ số thứ tự. Ví dụ: TH001, TH002 ... Nếu là khoa Toán tin, chúng tôi dùng 2 kí tự đầu là TT, 3 kí tự sau là số thứ tự, tương tự như vậy với khoa Vật lý.

Bạn cũng cần nhớ MSSV (mã số sinh viên) để khi nhập Mssv trong bảng KETQUA, bạn cũng phải nhập đúng mã của sinh viên đó, nếu nhập sai, máy sẽ báo lỗi vì bạn đã tạo mối quan hệ giữa hai bảng SINHVIEN và KETQUA.

Kéo con trượt ngang để thấy các vùng bị che bên phải, trong bảng SINHVIEN có vùng Khoamakh có mối liên hệ với Makhoa trong bảng

KHOA, bạn xem thông tin như thế nào, có thuộc một trong các mã khoa hay không?

Nhìn hình bên dưới, bạn thấy thông tin của vùng Khoamakh phù hợp phải không các bạn.

SINHVIEN : Table				
	TinhTP	Hocbong	Lylich	Khoamakh
▶	* TP.HCM	0		TH
	* TP.HCM	0		TH
	* TP.HCM	0		TH
	* TP.HCM	0		TT
	* Binh Duong	0		TT
	* TP.HCM	0		VL
	* TP.HCM	0		VL
*		0		

H197

◎ Bảng MONHOC

MONHOC : Table		
	Mamh	Tenmh
▶	* DTMT	Cấu trúc máy tính
	* DSCC	Đại số cao cấp
	* DTS	Điện tử số
	* LT	Lập trình
	* MMT	Mạng máy tính
	* TBD	Trình biên dịch
	* THCC	Toán học cao cấp
	* TMT	Toán máy tính
	* TP	Tổ pò
	* TRR	Toán rời rạc
	* VLLT	Vật lý lượng tử
	* VLPT	Vật lý phân tử
*		
Record: 1 of 12		

H198

Trong bảng MONHOC, các bạn cần nhớ mã môn học (Mamh), vì bảng này có mối liên hệ với bảng KETQUA, nên khi nhập kết quả của sinh viên, bạn phải nhập đúng mã môn học của sinh viên này.

◎ Bảng **KETQUA**

KETQUA : Table				
	Mamh	Lanthi	Masv	Diem
▶	CTMT		1 TH001	10
	CTMT		1 TH002	10
	CTMT		1 TH003	4
	CTMT		2 TH003	6
	THCC		1 TT001	9
	THCC		1 TT002	3
	THCC		2 TT002	5
	VLLT		1 VL001	8
	VLLT		1 VL002	2
	VLLT		2 VL002	4
	VLLT		3 VL002	6
*			0	0

H199

Trong bảng này Mamh phải giống với một trong các mã số môn học ở bảng MONHOC và Masv phải giống với một trong các mã sinh viên ở bảng SINHVIEN.

Nhờ sự giống nhau này mà chúng ta truy xuất đúng thông tin của từng sinh viên.

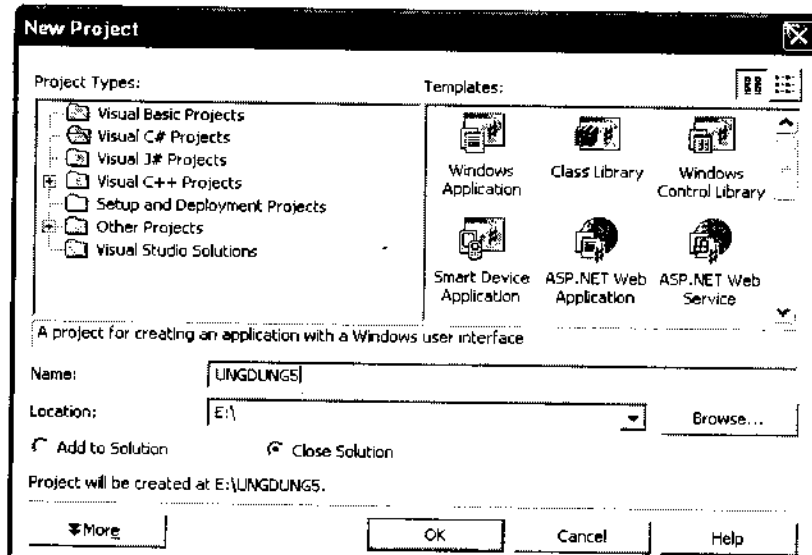
Bạn bắt tay vào thiết kế chương trình trong Visual C# như sau:

Bạn tạo một ứng dụng Windows Forms mới bằng cách:

◆ Vào menu **File**, di chuyển vệt sáng tới **New**, chọn **Project**.

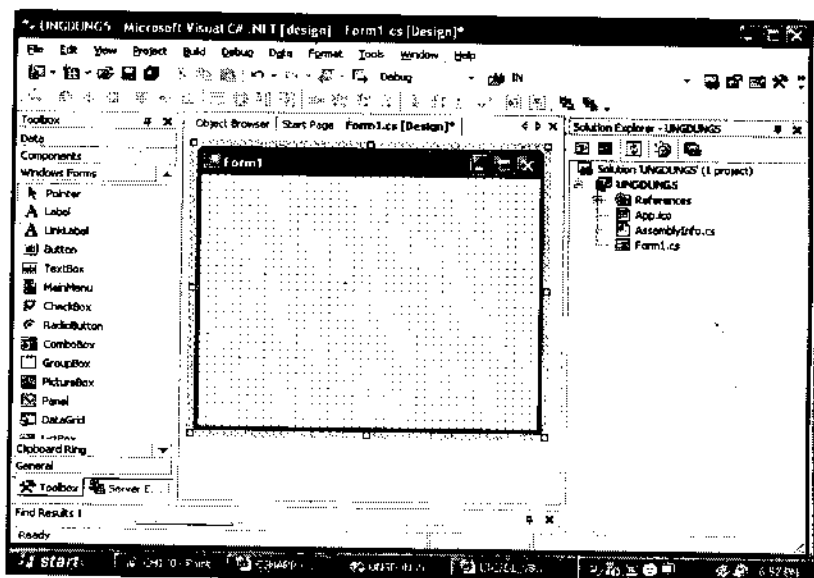
◆ Trong hộp thoại hiện ra, chọn **Visual C# Projects** trong khung **Project Types**. Chọn **Windows Application** trên **Templates**.

◆ Đặt tên cho ứng dụng chẳng hạn UNGDUNG5 như hình sau:



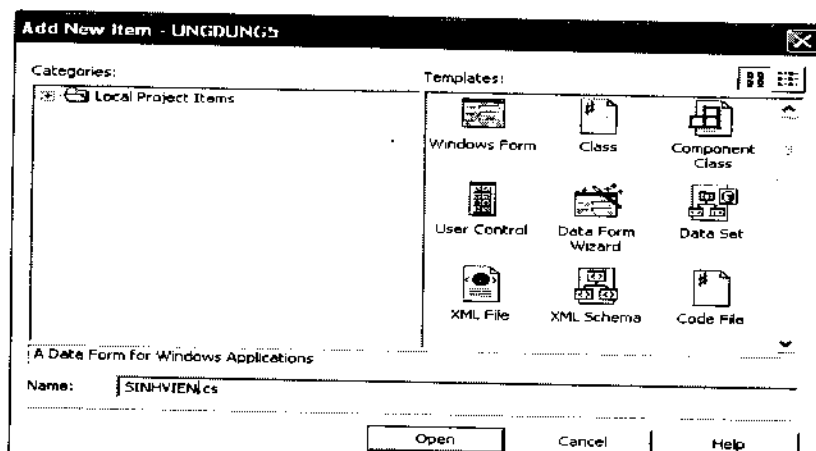
H200

- ◆ Click nút **OK**
- ◆ Cửa sổ sau hiện ra:



H201

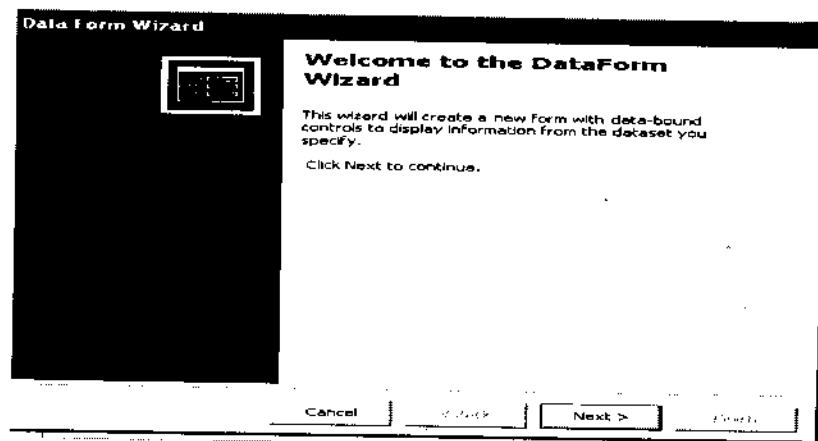
Từ cửa sổ này bạn thêm một Data Form Wizard bằng cách:
Vào menu Project, chọn Add New Item, hộp thoại sau hiện ra



H202

◆ Bạn chọn **Data Form Wizard** từ **template** có sẵn như hình trên.
Bạn có thể đặt lại tên của lớp Data Form vào khung Name nếu như không thích tên này. Giả sử chúng ta đặt tên là SINHVIEN.cs.

◆ Click nút **Open**, bạn thấy hình sau:



H203

◆ Click nút **Next**, chúng ta được

Data Form Wizard

Choose the dataset you want to use
The dataset contains the tables and columns to work with in your form.

The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

☐ Use the following dataset:

Cancel < Back Next > Finish

H204

Bạn đặt tên cho dataset để truy xuất dữ liệu, giả sử nhập tên BANGSINHVIEN vào khung trống.

Data Form Wizard

Choose the dataset you want to use
The dataset contains the tables and columns to work with in your form.

The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

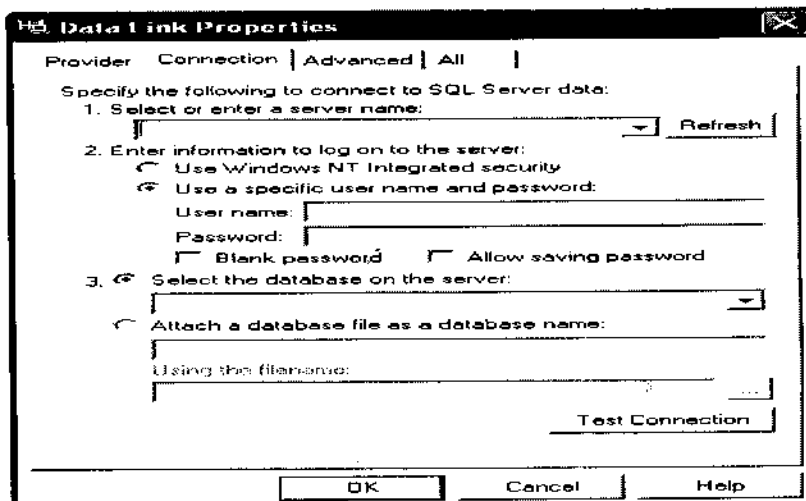
☒ Create a new dataset named:

☐ Use the following dataset:

Cancel < Back Next > Finish

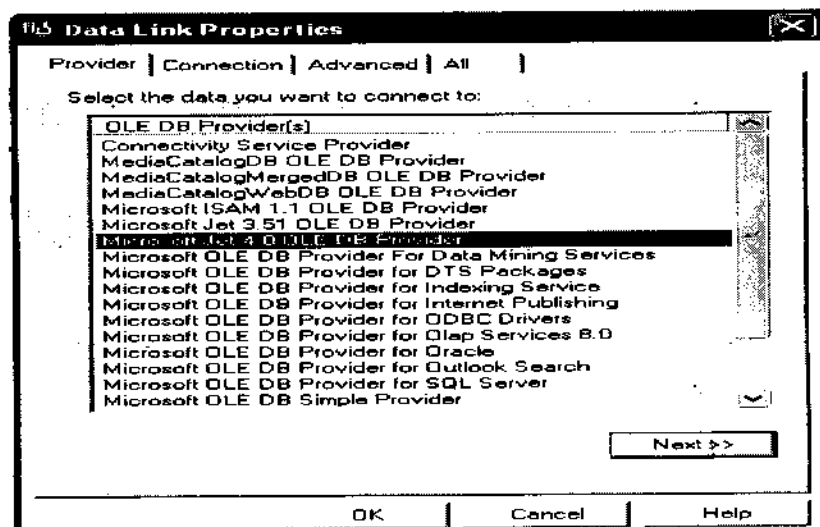
H205

◆ Click nút **Next**, ta sẽ thấy hộp thoại kết nối hiện ra. Bạn click nút New Connection để tìm đến cơ sở dữ liệu QLSINHVIEN.mdb. Hình sau sẽ hiện ra:



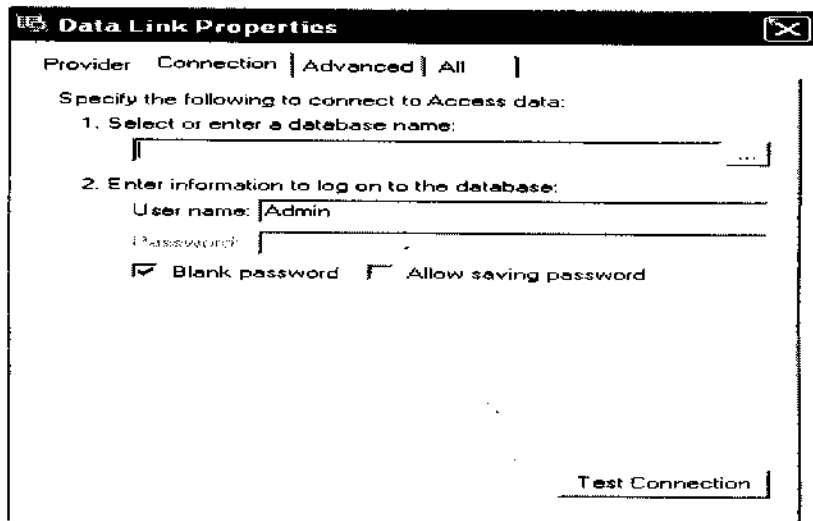
H206

- ◆ Click chọn tab có tên Provider
- ◆ Chọn Microsoft jet 4.0 OLE DB Provider như hình sau:



H207

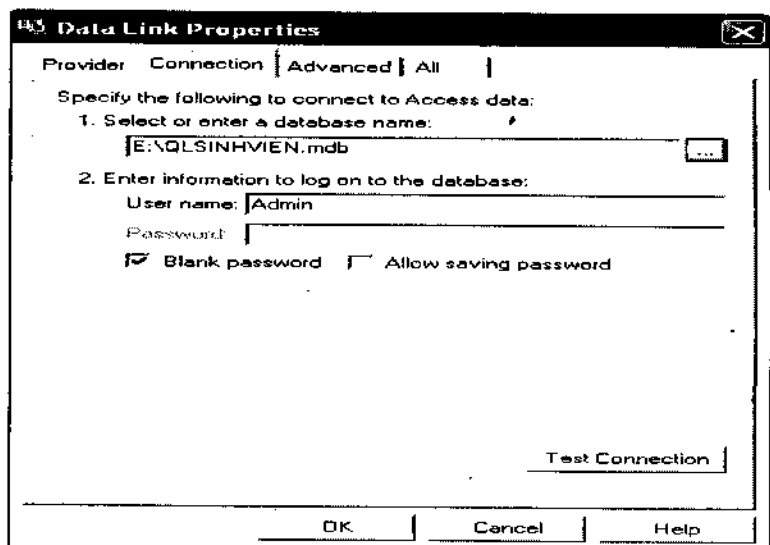
- ◆ Click nút Next, hộp thoại sau hiện ra



H208

◆ Click nút có dấu ba chấm "..." bên phải để tìm đến cơ sở dữ liệu có tên QLSINHVIEN.mdb.

Sau khi tìm chọn xong, bạn thấy hộp thoại như hình sau:



H209

◆ Click nút OK, bạn thấy kết quả như hình sau:

Data Form Wizard

Choose a data connection

The data connection specifies how to find and log into a specific server and database.



Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.

Which connection should the wizard use?

ACCESSIBLE\SQLSVN\SVN.mdb, Admin...

New Connection...

Cancel

< Back

Next >

Finish

H210

◆ Click nút **Next**, bạn thấy hộp thoại hiện ra cho chúng ta chọn các **Tables** hoặc **Views** mà bạn muốn kết nối về **dataset**.

Data Form Wizard

Choose tables or views

The tables or views you choose will determine which columns will be available to display on your form.



The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

Tables
- KETQUA
- KHOA
- MONHOC
- SINHVIEN
Views

Selected item(s):

Tables
Views

Cancel

< Back

Next >

Finish

H211

◆ Bạn chọn bảng **SINHVIEN**, sau đó click nút ">" để đưa bảng này qua khung bên phải. Thực hiện xong bạn thấy hình như sau:

Data Form Wizard

Choose tables or views
The tables or views you choose will determine which columns will be available to display on your form.

The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

- Tables
 - MONHOC
 - KETQUA
 - KHOA
- Views

Selected item(s):

- Tables
 - SINHVIEN**
- Views

Buttons: Cancel, < Back, Next >, Finish

H212

◆ Click nút Next , bạn thấy bảng sau:

Data Form Wizard

Choose tables and columns to display on the form
Your form can display any of the tables and columns available in the dataset.

If you display more than one table on the form, the tables have a master-detail relationship

What tables and columns do you want to display on the form?

Master or single table:

SINHVIEN

Detail table:

Columns:

- ☒ Diachi
- ☒ Hocbong
- ☒ Holot
- ☒ Khoamakh
- ☒ Lylich
- ☒ MSSV
- ☒ Nam

Columns:

Buttons: Cancel, < Back, Next >, Finish

H213

Vì biểu mẫu thứ nhất này, ý đồ của chúng ta là chỉ cho một bảng SINHVIEN hiện ra trong biểu mẫu, do đó chúng ta chỉ chọn một bảng mà thôi.

Chúng ta chọn tất cả các vùng của bảng SINHVIEN đều hiện ra trong biểu mẫu.

◆ Click nút Next, chúng ta có bảng lựa chọn cách thể hiện của dữ liệu trên biểu mẫu. Giả sử chúng ta lựa chọn Single record in individual controls.

Chọn xong bạn có hình như sau:

Data Form Wizard

Choose the display style
You can display a single record at a time or display all records at once.

How do you want to display your data?

☐ All records in a grid

☒ Single record in individual controls

What additional controls do you want on the form?

☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

☒ Add - Creates a new record.

☒ Delete - Deletes the current record.

☒ Cancel - Cancels changes to the current record.

☒ Navigation controls - Moves to first, previous, next, or last record.

The wizard now has the information it needs. Click Finish to exit and generate your new form.

Cancel < Back Next > Finish

H214

◆ Click nút Finish để kết thúc, chúng ta có hộp thoại hiện ra hỏi như hình sau:

Do you want to include the password in the connection string?

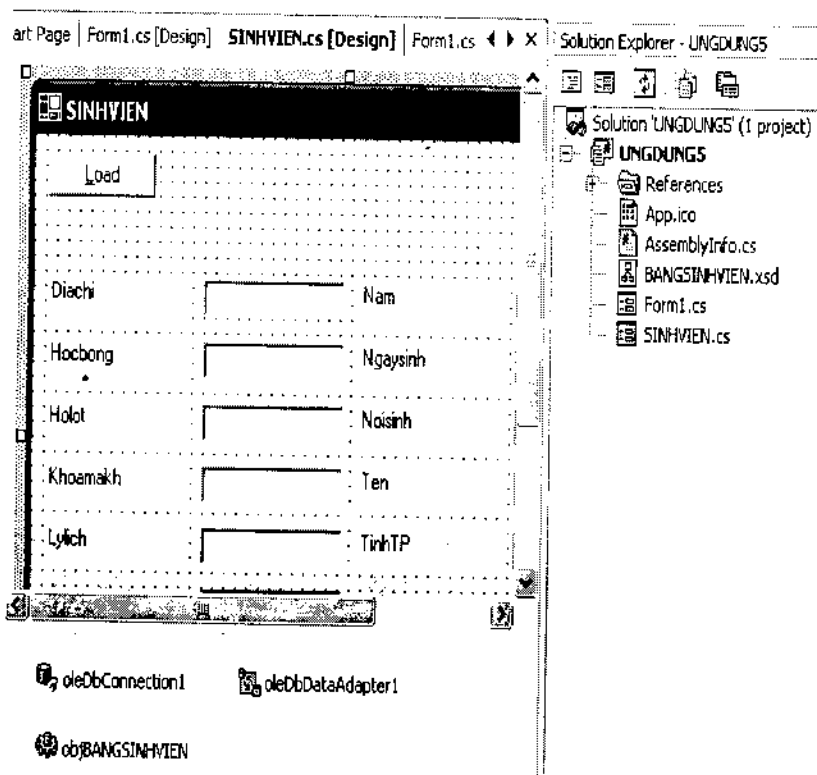
The password is saved as clear text and is readable in the source code and the compiled assembly.

Include password Don't include password Cancel Help

H215

◆ Bạn click nút "Don't include password"

Biểu mẫu của chương trình như sau:



H216

Click chọn Form1, click nút View Code, sửa lại câu lệnh thay vì static void Main()

```
{
    Application.Run(new Form1());
}
```

Thành

static void Main()

```
{
    Application.Run(new SINHVIEN());
}
```

Nhấn phím F5 để chạy thử chương trình, bạn sẽ hình như sau:

SINHVIEN

Load Update
Cancel All

Diachi		Nam	
Hocbong		Ngaysinh	
Holot		Noisinh	
Khoamakh		Ten	
Lylich		TinhTP	
MSSV			

<< < No Records > >>
Add Delete Cancel

H217

◆ Click nút Load trên biểu mẫu, bạn có kết quả như sau:

SINHVIEN

Load Update
Cancel All

Diachi	P24 Q10	Nam	☒
Hocbong	0	Ngaysinh	12/3/1987 12:00:00
Holot	Hoang	Noisinh	TP.HCM
Khoamakh	TH	Ten	Anh
Lylich		TinhTP	TP.HCM
MSSV	TH001		

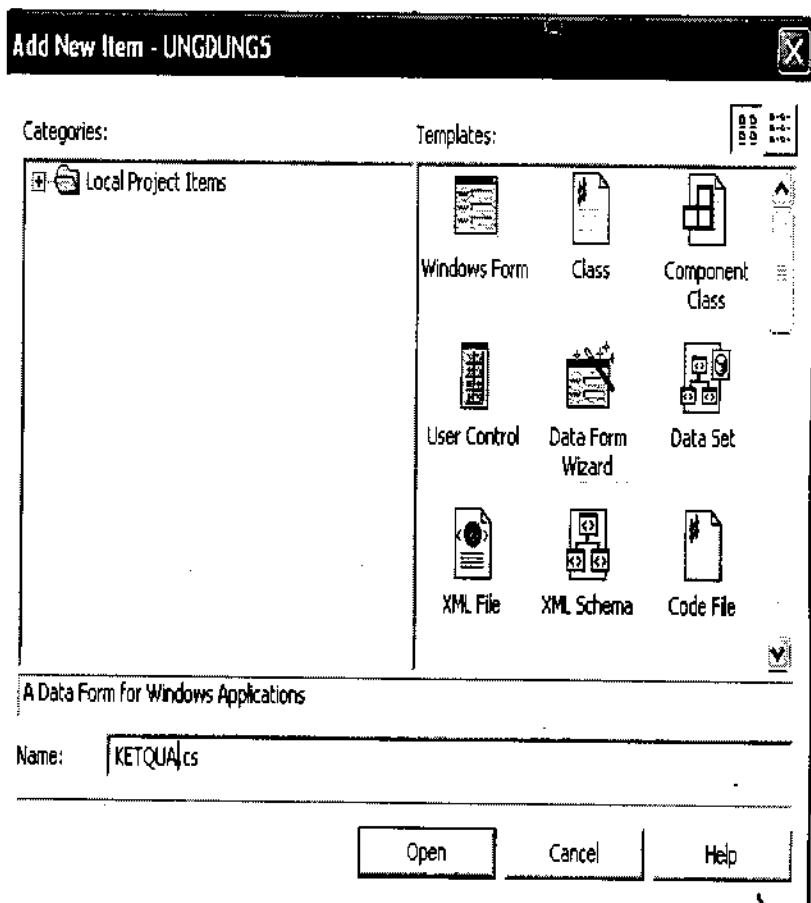
<< < 1 of 7 > >>
Add Delete Cancel

H218

Bạn thấy tập tin này có 7 mẫu tin, hiện hành là mẫu tin thứ nhất, bạn thử nhập thêm một số sinh viên nữa vào trong bảng SINHVIEN này.

Tiếp theo bạn tạo một biểu mẫu chỉ chứa một bảng KETQUA

Trong chương trình, vào menu Project, chọn Add New Item, trong hộp thoại hiện ra đặt tên là KETQUA.cs



H219

◆ Click nút Open

Sau đó thực hiện các bước giống như trên, đặt tên Dataset là BANGKETQUA

Data Form Wizard

Choose the dataset you want to use

The dataset contains the tables and columns to work with in your form.



The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

BANGKETQUA

☐ Use the following dataset:

LINGDUNG5.BANGSINHVIEN

Cancel

< Back

Next >

Finish

H220

◆ Click nút Next và thực hiện các bước tiếp theo, vì đã kết nối với CSDL QLSINHVIEN.mdb nên bạn chỉ cần click nút Next để tiếp tục, lưu ý chỉ chọn bảng KETQUA đưa qua khung bên phải. Ở trang cuối vẫn chọn cách 2 đó là Single record in individual controls. Click nút Finish để kết thúc.

Click chọn Form1, click nút View Code, sửa lại câu lệnh thay vì static void Main()

```
{  
    Application.Run(new SINHVIEN());  
}
```

Thành

static void Main()

```
{  
    Application.Run(new KETQUA());  
}
```


Nhấn phím F5 để chạy thử chương trình, bạn sẽ hình như sau:

The screenshot shows the KETQUA application window. It has a title bar with the text 'KETQUA' and standard window controls. Inside the window, there is a 'Load' button in the top left. On the right side, there are 'Update' and 'Cancel All' buttons. Below these, there are two rows of input fields: 'Diem' and 'Lanthi' on the left, and 'Mamh' and 'Masv' on the right. All fields are empty. At the bottom, there is a status bar showing '<<' and '<' buttons on the left, 'No Records' in the center, and '>' and '>>' buttons on the right. Below the status bar, there are three buttons: 'Add', 'Delete', and 'Cancel'.

H221

◆ Click nút Load, bạn có kết quả như hình sau:

The screenshot shows the KETQUA application window after clicking the 'Load' button. The 'Diem' field now contains the value '10' and the 'Mamh' field contains 'CTMT'. The 'Lanthi' field contains '1' and the 'Masv' field contains 'TH001'. The status bar now shows '<<' and '<' buttons on the left, '1 of 11' in the center, and '>' and '>>' buttons on the right. The 'Add', 'Delete', and 'Cancel' buttons are still present at the bottom.

H222

Tiếp theo bạn tạo một biểu mẫu chỉ chứa một bảng KHOA

Trong chương trình, vào menu Project, chọn Add New Item, trong hộp thoại hiện ra đặt tên là KHOA.cs

chọn cách 2 đó là Single record in individual controls. Click nút Finish để kết thúc.

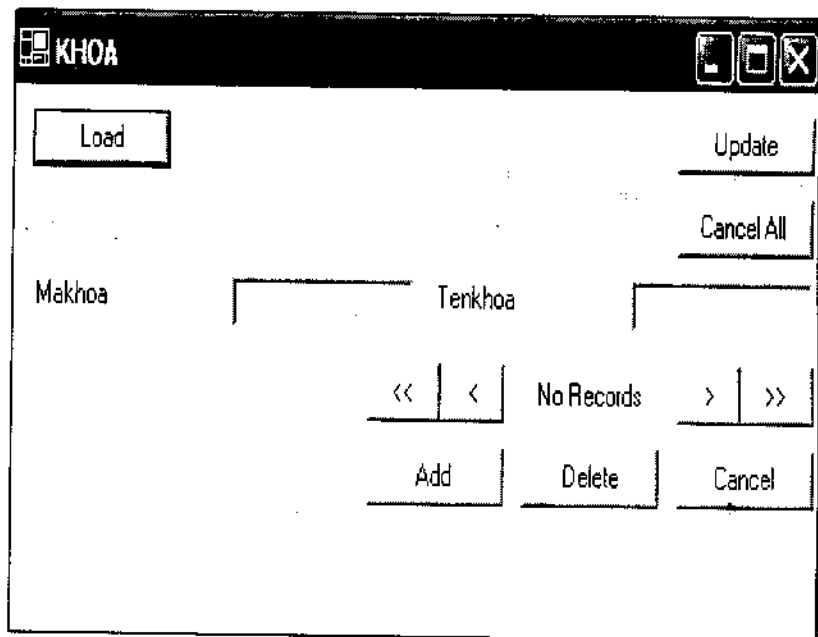
◆ Click chọn Form1, click nút View Code, sửa lại câu lệnh thay vì static void Main()

```
{  
    Application.Run(new KETQUA());  
}
```

Thành
static void Main()

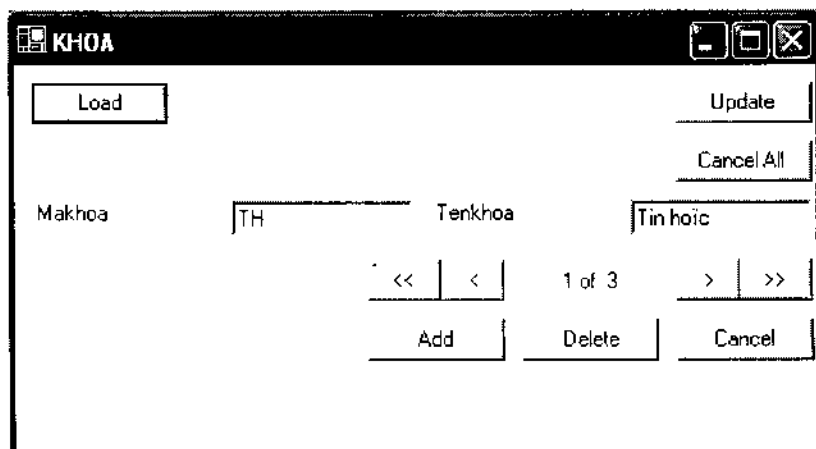
```
{  
    Application.Run(new KHOA());  
}
```

Nhấn phím F5 để chạy thử chương trình, bạn sẽ hình như sau:



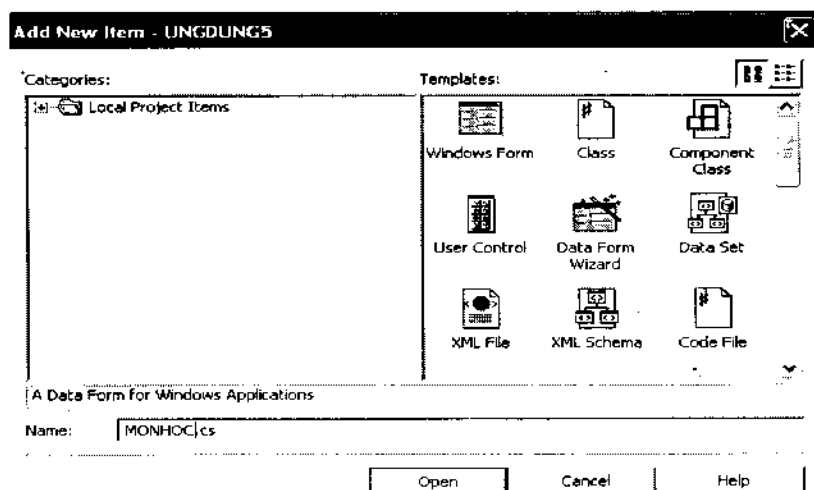
H225

◆ Click nút Load, bạn có kết quả như hình sau:



H226

Tiếp theo bạn tạo một biểu mẫu chỉ chứa một bảng MONHOC
 Trong chương trình, vào menu Project, chọn Add New Item, trong
 hộp thoại hiện ra đặt tên là MONHOC.cs



H227

◆ Click nút Open

Sau đó thực hiện các bước giống như trên, đặt tên Dataset là
 BANGMONHOC

Data Form Wizard

Choose the dataset you want to use

The dataset contains the tables and columns to work with in your form.



The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

BANGMONHOC

☐ Use the following dataset:

LINKDUNGS.BANGKETQUA

Cancel

< Back

Next >

Finish

H228

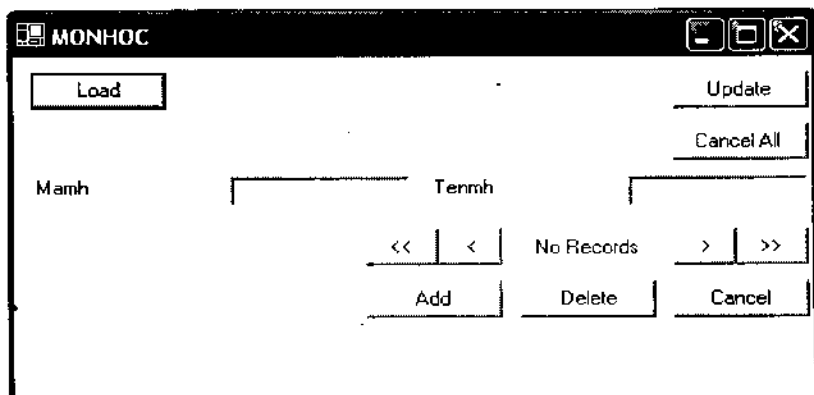
◆ Click nút Next và thực hiện các bước tiếp theo, vì đã kết nối với CSDL QLSINHVIEN.mdb nên bạn chỉ cần click nút Next để tiếp tục, lưu ý chỉ chọn bảng MONHOC đưa qua khung bên phải. Ở trang cuối vẫn chọn cách 2 đó là Single record in individual controls. Click nút Finish để kết thúc.

◆ Click chọn Form1, click nút View Code, sửa lại câu lệnh thay vì static void Main()

```
{  
    Application.Run(new KHOA());  
}
```

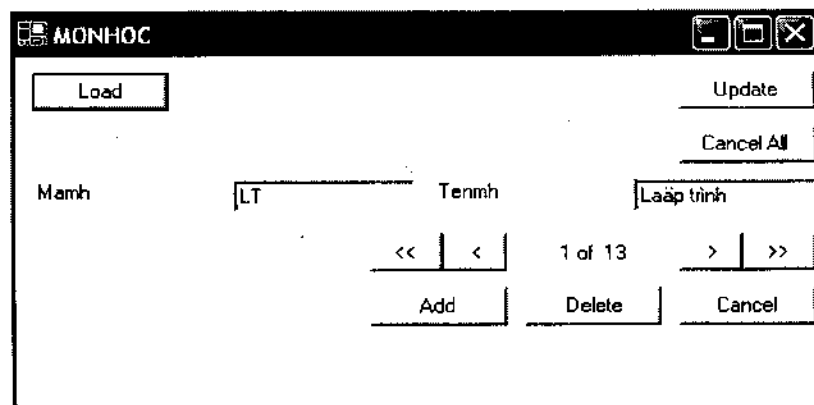
```
Thành  
static void Main()  
{  
    Application.Run(new MONHOC());  
}
```

Nhấn phím F5 để chạy thử chương trình, bạn sẽ hình như sau:



H229

◆ Click nút Load, bạn có kết quả như hình sau:



H230

Bạn hãy xác lập thuộc tính Font cho 4 biểu mẫu trên, để thông tin hiện ra trong các điều khiển là tiếng Việt.

Bạn thử thực tập thêm lần lượt ba biểu mẫu mới. Đưa hai bảng SINHVIEN và KETQUA vào trong biểu mẫu thứ nhất. Đưa hai bảng MONHOC và KETQUA vào trong biểu mẫu thứ hai và hai bảng SINHVIEN và KHOA vào trong biểu mẫu thứ ba.

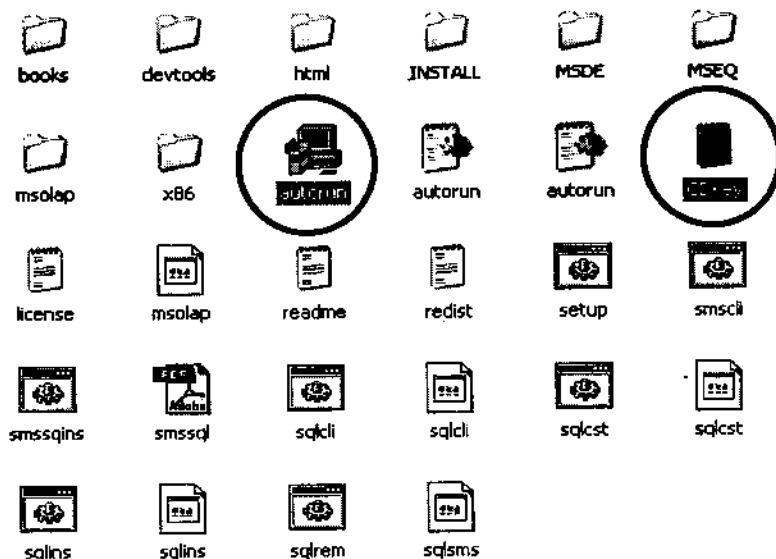
Sau này khi muốn biểu mẫu nào xuất hiện khi chạy chương trình, bạn sửa lại code, bằng cách gõ tên biểu mẫu cần xuất hiện như các bài tập ở trên.

LẬP TRÌNH VISUAL C# DÙNG CƠ SỞ DỮ LIỆU SQL SERVER

Cách cài đặt SQL SERVER 2000

- ☐ Cho đĩa CD **SQL SERVER 2000** vào ổ đĩa
- ☐ Click nút phải chuột tại nút **Start**, chọn **Explorer**
- ☐ Trong cửa sổ click chọn tên ổ đĩa, bạn sẽ thấy các tập tin có trong đĩa CD như hình sau:

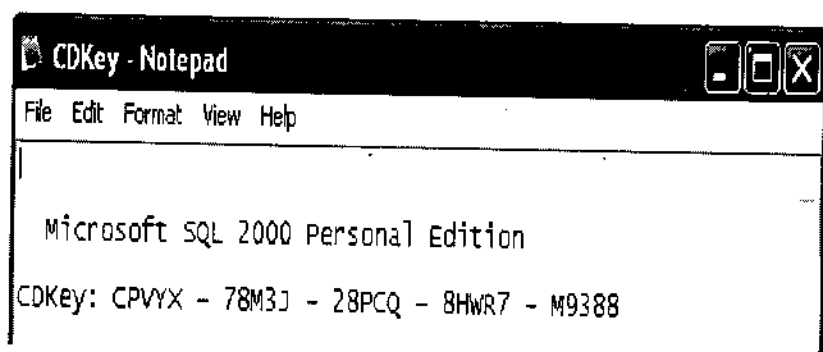
Files Currently on the CD



H231

Nhìn trong danh sách các mục, bạn thấy có tập tin **CDKey** và tập tin **Autorun.exe**

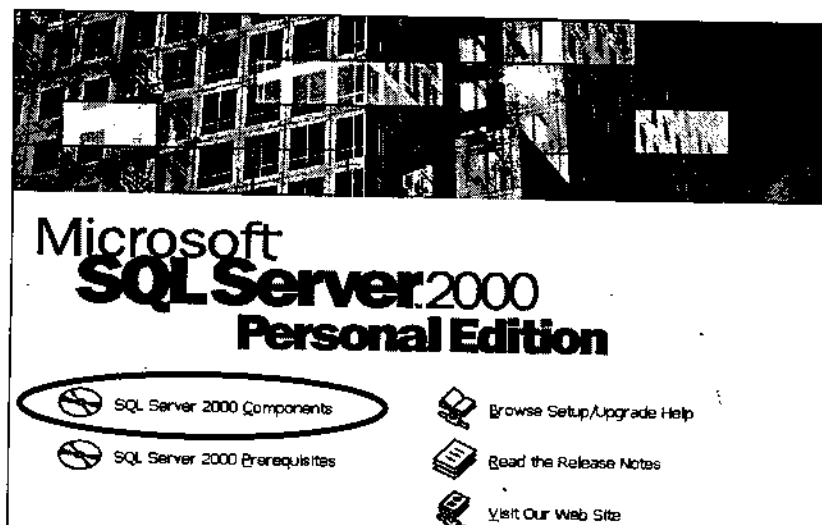
Click nút *phải chuột* trên tập tin **CDKey**, chọn **Open**, bạn sẽ thấy hình sau:



H232

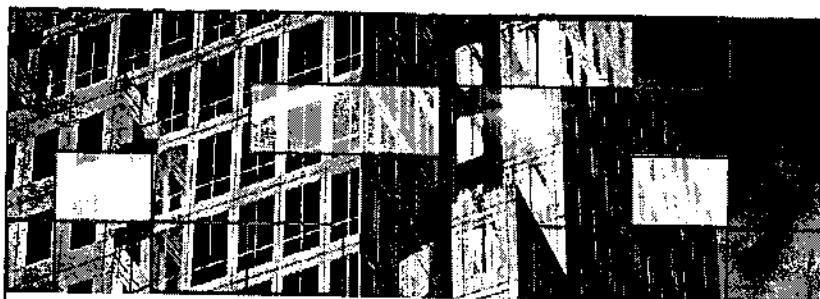
Bạn ghi CDKey ra giấy để khi cài đặt sẽ điền vào hộp thoại. Đó là
CPVYX – 78M3J – 28PCQ – 8HWR7 – M9388

Click nút *phải chuột* trên tập tin **autorun.exe**, click **Open** trong menu hiện ra. Bạn thấy hình như sau:



H233

Trong trường hợp này bạn chọn **SQL Server 2000 Components**. Bạn thấy hình sau hiện ra



Microsoft **SQL Server 2000** **Personal Edition**

Install Components



Install Database Server



Install Analysis Services



Install English Query

Back

Exit

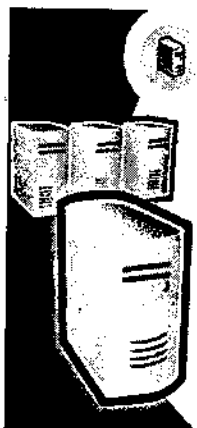
H234

Trong hình có ba mục:

- Cài đặt cơ sở dữ liệu SQL Server 2000
- Cài đặt dịch vụ phân tích
- Cài đặt dịch vụ truy vấn anh ngữ

☐ Bạn chọn mục **Install Database Server** (cài đặt CSDL SQL Server 2000). Bạn thấy hình sau hiện ra

Welcome



Welcome to the Microsoft SQL Server Installation Wizard.

The Installation Wizard allows you to install a new instance of SQL Server or modify an existing instance.

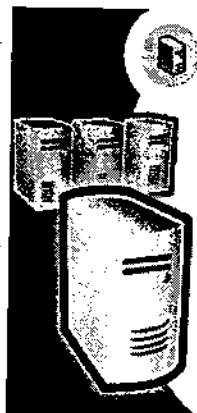
Next >

Cancel

H235

☐ Click nút **Next**, bạn có hình như sau:

Computer Name



Enter the name of the computer on which you want to create a new instance of SQL Server or modify an existing instance of SQL Server.

Or, enter the name of a new or existing Virtual SQL Server to manage.

COMPUTER

- ☒ Local Computer
- ☐ Remote Computer
- ☐ Virtual Server

Browse...

Help

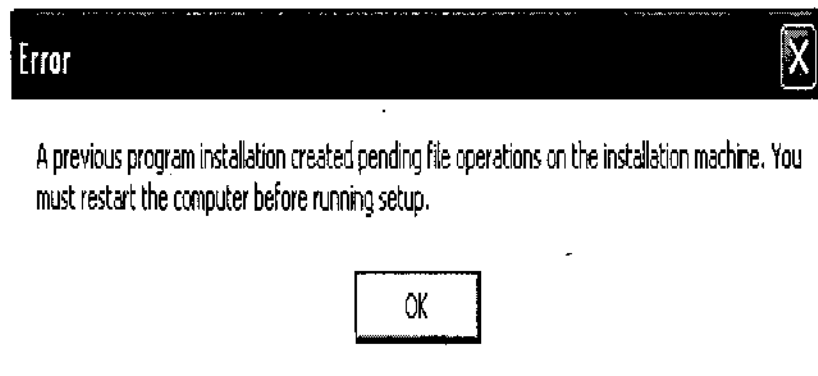
< Back

Next >

Cancel

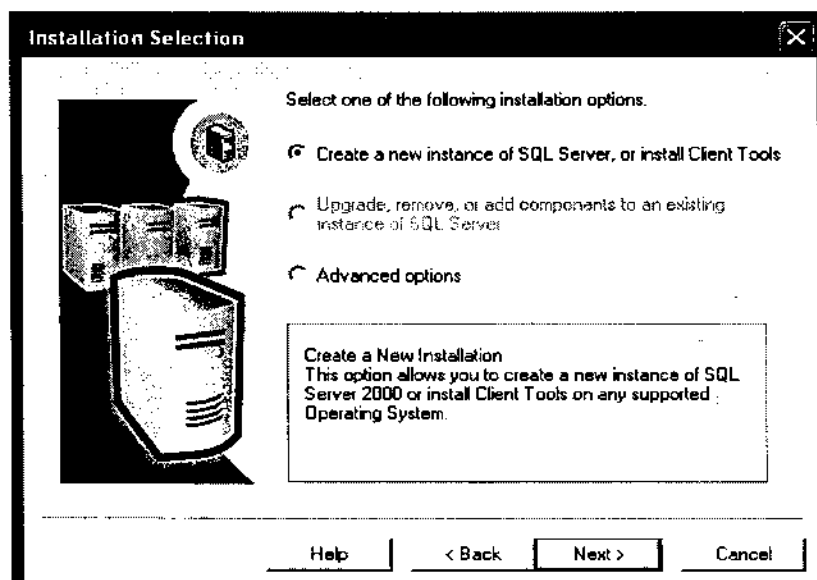
H236

☐ Click nút **Next**, hộp thoại kế tiếp hiện ra (nếu có hộp thoại báo lỗi hiện ra vì một lý do nào đó, yêu cầu bạn khởi động lại rồi mới chạy chương trình cài đặt, click OK và khởi động lại máy)



H237

☐ Khởi động lại máy tính và thực hiện cài đặt lại, khi cài đặt tới hình H5, Click nút **Next**, bạn sẽ thấy hình như sau:

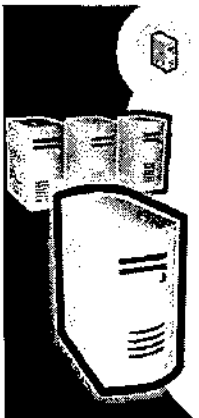


H238

☐ Click nút **Next**, bạn thấy hình sau hiện ra

User Information [X]

Enter your name below. It is not necessary to enter a company name.



Name:

Company:

< Back Next > Cancel

H239

Do máy chúng tôi mua ở Phong Vũ khi cài đặt Windows nên có tên như vậy, Click nút **Next**, bạn thấy hộp thoại sau:

Software License Agreement [X]

 Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement.

ADDENDUM TO THE MICROSOFT END USER LICENSE AGREEMENT FOR MICROSOFT SQL SERVER 2000

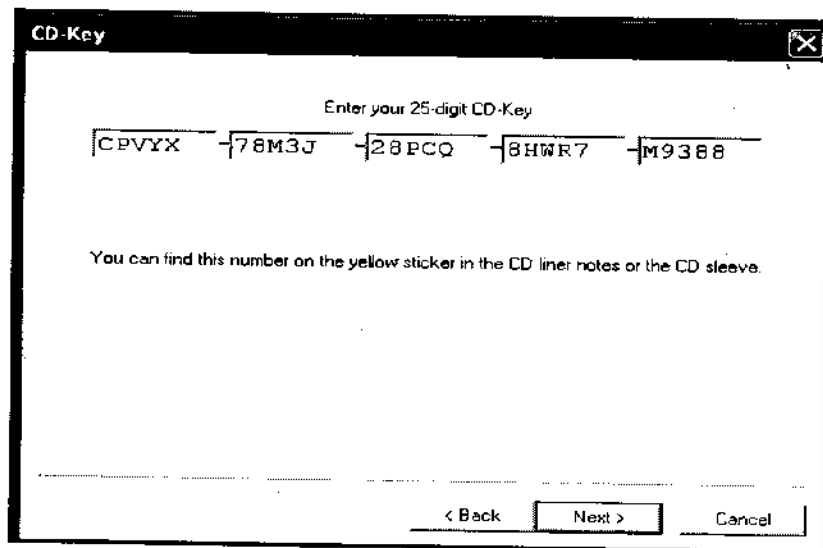
The software accompanying this Addendum, Microsoft SQL Server Personal Edition (the "Client Software") is provided to you for use under the terms and conditions of the end user license agreement you acquired with Microsoft SQL Server (Standard or Enterprise Edition) (the "EULA"). Please refer to the EULA for license rights and requirements associated with Client Software. The Client Software is deemed part of the Product (as defined in the EULA), and as such, if you do not have a validly licensed copy of the Product, you are not authorized to use the Client Software. Any capitalized terms used in this Addendum shall have the same meaning as set forth in the EULA, unless otherwise set forth in this Addendum. All terms and conditions of the EULA remain in full force and effect.

Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install Microsoft SQL Server 2000, you must accept this agreement.

< Back Yes No

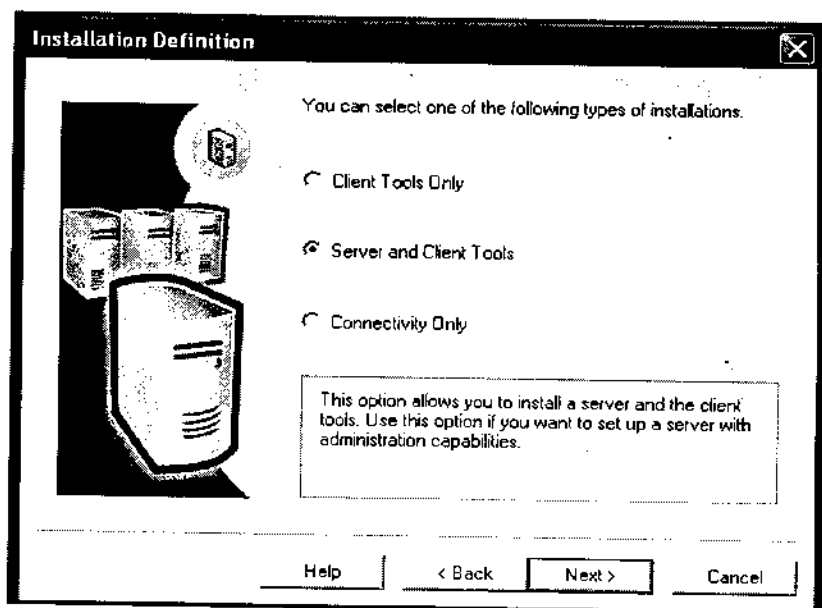
H240

☐ Click nút **Yes**, bạn sẽ có hộp thoại như hình sau:



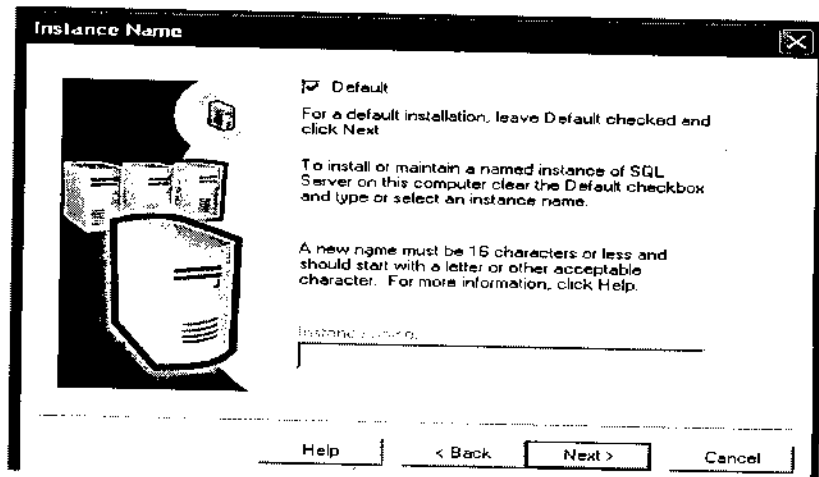
H241

☐ Bạn nhập **CDKey** mà bạn ghi ngoài giấy vào trong các ô, click nút **Next** bạn có hình như sau:



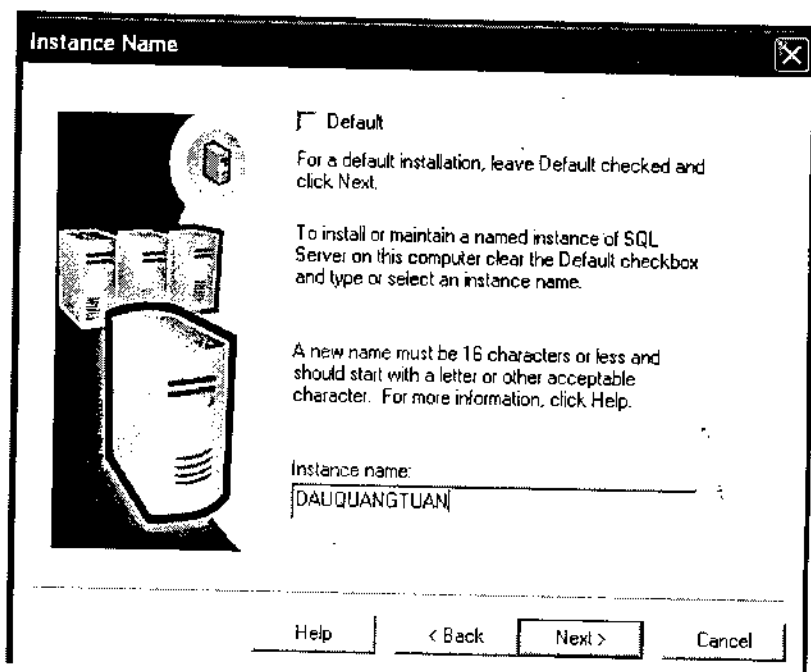
H242

■ Click nút **Next**, bạn có hình như sau:



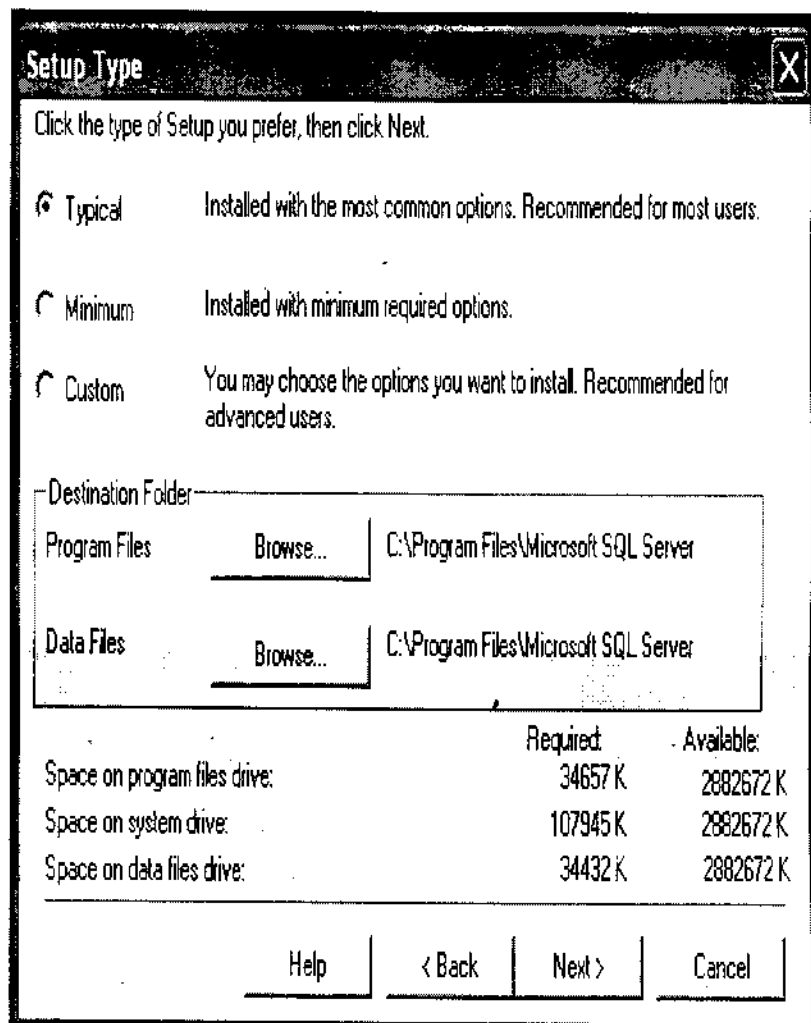
H243

☐ Click bỏ dấu chọn mục **Default**. Trong khung **Instance name** giả sử chúng tôi nhập **DAUQUANGTUAN**



H244

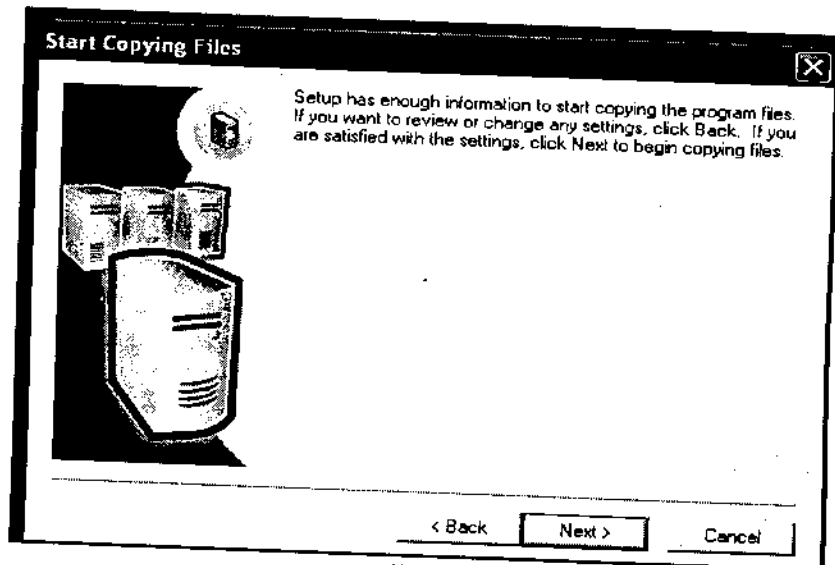
☐ Click nút **Next**, bạn sẽ thấy hình như sau:



H245

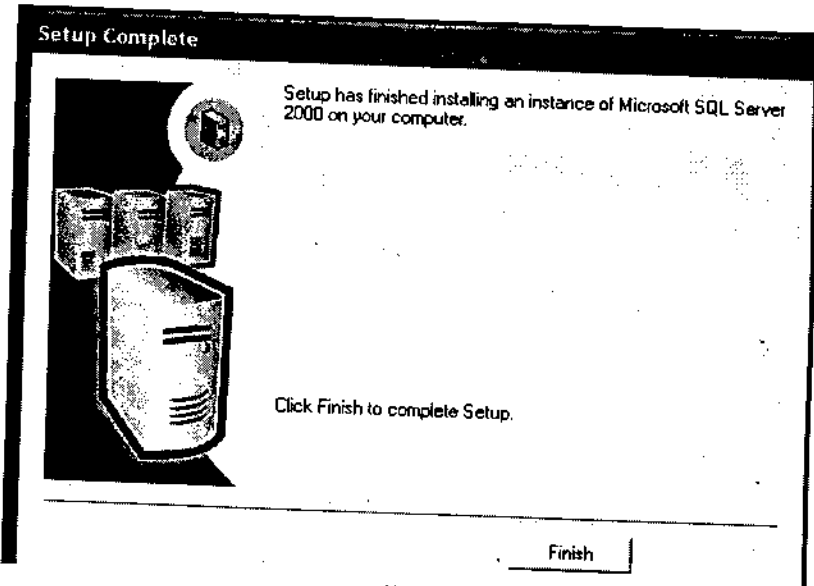
Nếu ổ đĩa C:\ còn dung lượng ít, bạn có thể chọn cài đặt tập tin SQL Server 2000 vào đĩa C:\ và lưu trữ cơ sở dữ liệu của SQL Server 2000 và cơ sở dữ liệu của người dùng vào một ổ đĩa khác. Nếu cần thay đổi vị trí cài đặt hay lưu trữ dữ liệu, chúng ta chọn nút **Browse**. Nếu dung lượng ổ đĩa C:\ còn trống, bạn có thể cài đặt chung cả hai phần này trên cùng một ổ đĩa.

☐ Click nút **Next**, click chọn mục **Use the Local System account** như hình sau:



H248

☐ Click nút **Next** quá trình chép các files bắt đầu, bạn chờ một thời gian là hoàn tất việc cài đặt SQL Server 2000. Khi cài đặt xong, bạn thấy hộp thoại sau hiện ra



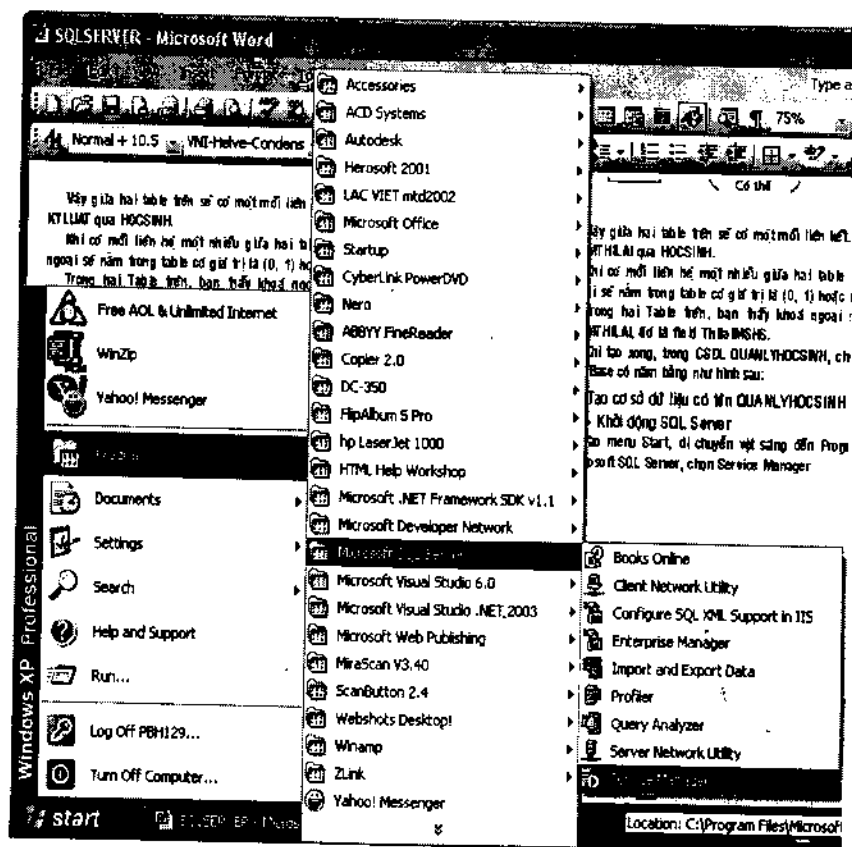
H249

☐ Click nút **Finish** để hoàn tất việc cài đặt

Tạo cơ sở dữ liệu, bảng Và mối liên hệ giữa các bảng trong SQL Server 2000

➤ Khởi động SQL Server

◆ Vào menu Start, di chuyển vệt sáng đến Programs, di chuyển đến Microsoft SQL Server, chọn Service Manager



H250

Bạn sẽ thấy cửa sổ SQL Server Service Manager hiện ra như hình sau:

SQL Server Service Manager

Server: COMPUTER\DAUQUANGTUAN

Services: SQL Server

Refresh services...



Start/Continue



Pause



Stop

☒ Auto-start service when OS starts

Stopped - \\COMPUTER\DAUQUANGTUAN - MSSQLSer

H251

Click nút đầu mũi tên có tên **Start/Continue**

SQL Server Service Manager

Server: COMPUTER\DAUQUANGTUAN

Services: SQL Server

Refresh services...



Start/Continue



Pause



Stop

☒ Auto-start service when OS starts

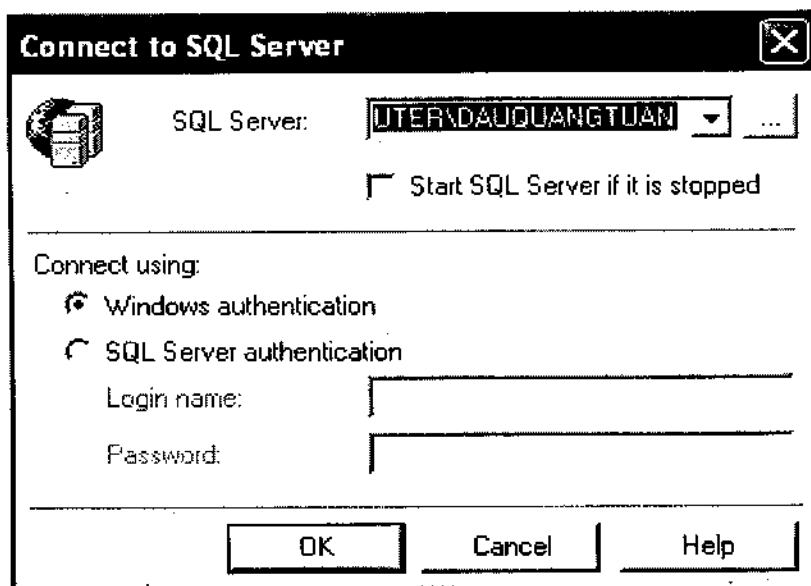
Running - \\COMPUTER\DAUQUANGTUAN - MSSQLSer

H252

Do khi cài đặt chúng tôi đặt tên **DAUQUANGTUAN** trong hộp thoại hiện ra, nên bây giờ các bạn thấy tên hiện ra trong khung Server.

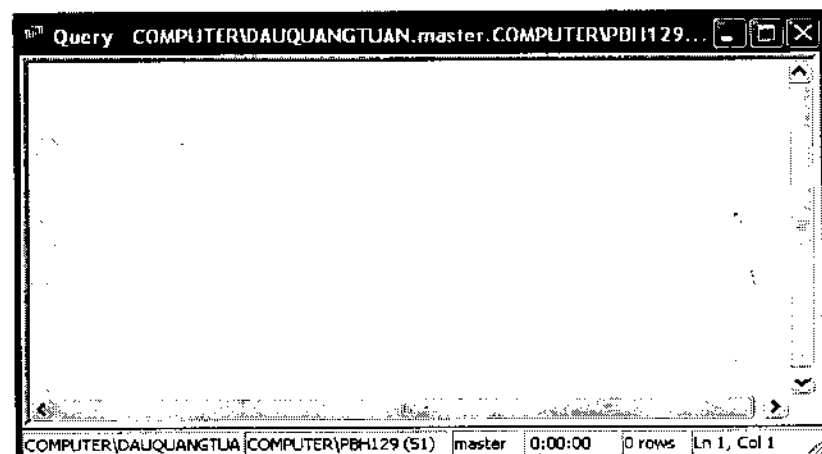
Khởi động Query Analyzer

◆ Vào menu **Start**, di chuyển vệt sáng đến **Programs**, di chuyển đến **Microsoft SQL Server**, chọn Query Analyzer



H253

◆ Click nút OK, bạn thấy hình sau:

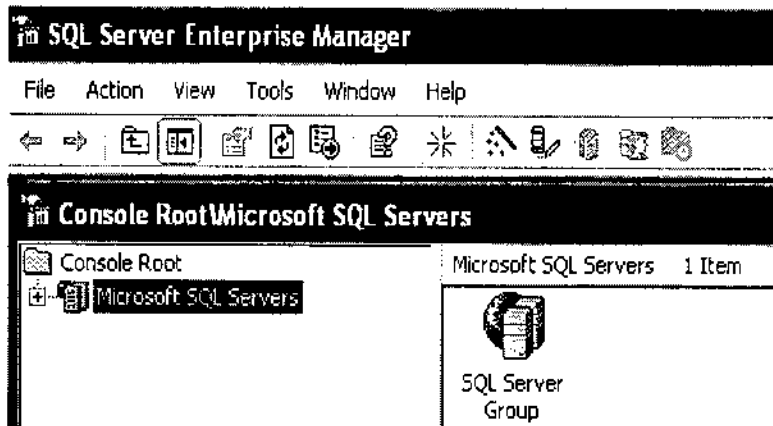


H254

Tạo cơ sở dữ liệu

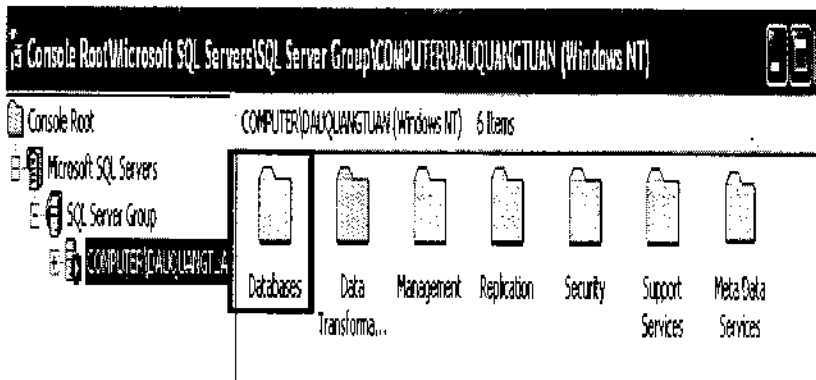
Có ba cách tạo CSDL trong SQL Server 2000, đó là dùng trình điều khiển Enterprise Manager (EM), bằng Wizards hoặc bằng phát biểu SQL. Trong bài này, chúng tôi hướng dẫn các bạn dùng cách thứ nhất, đó là dùng trình điều khiển Enterprise Manager.

◆ Vào menu **Start**, di chuyển vệt sáng đến **Programs**, đến **Microsoft Server 2000**, click chọn **Enterprise Manager**



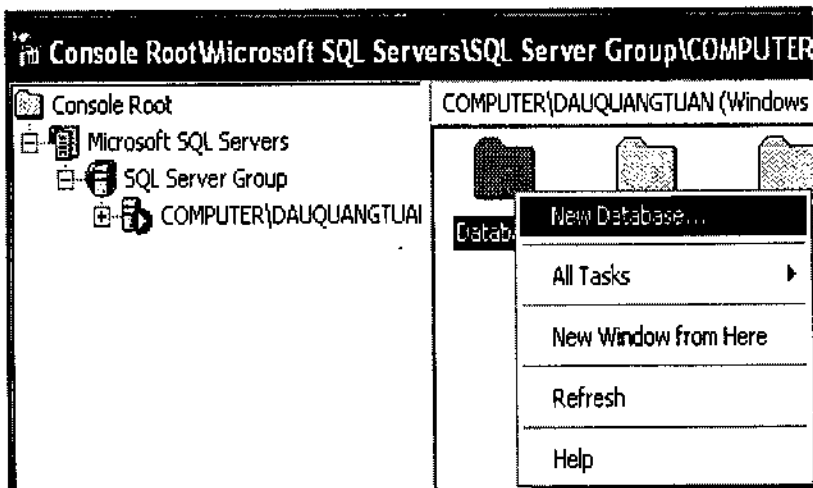
H255

◆ Click dấu cộng (+) trước **Microsoft SQL Server**, click chọn tên **COMPUTER\DAUQUANGTUAN**, bạn thấy hình như sau:



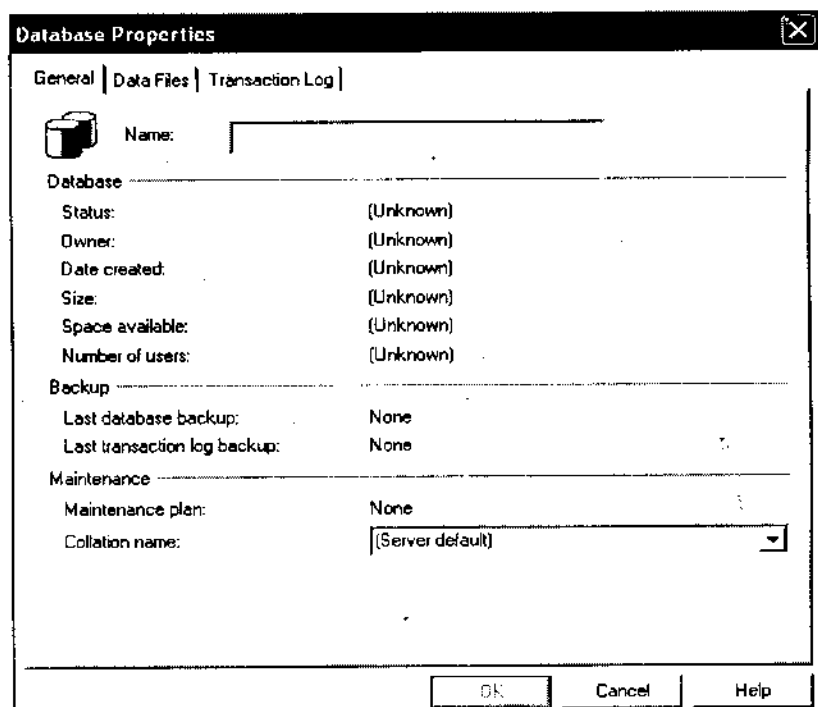
H256

◆ Click nút phải chuột tại **Databases**



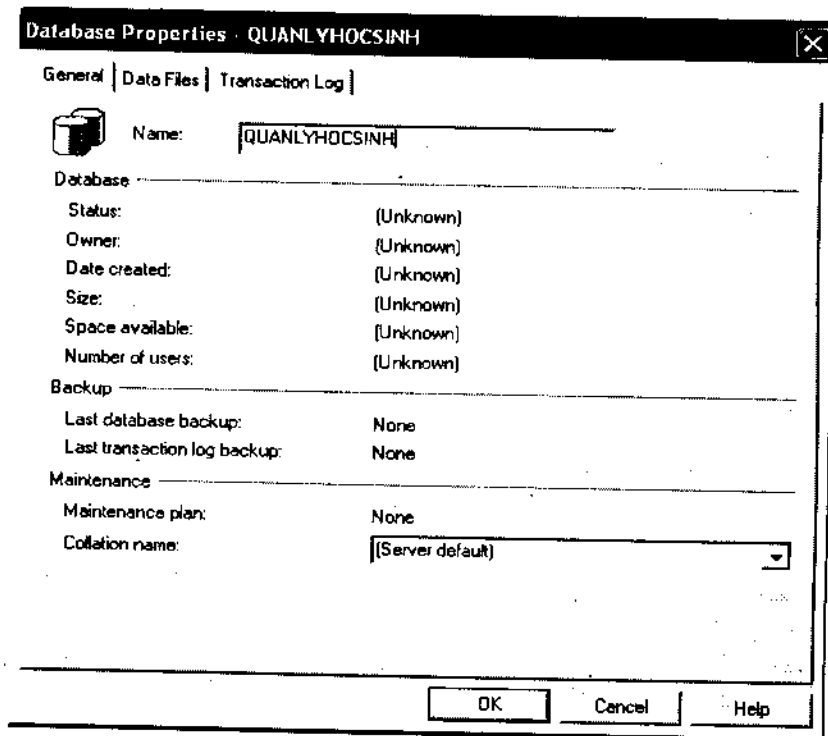
H257

◆ Chọn New Database bạn sẽ thấy hình như sau:



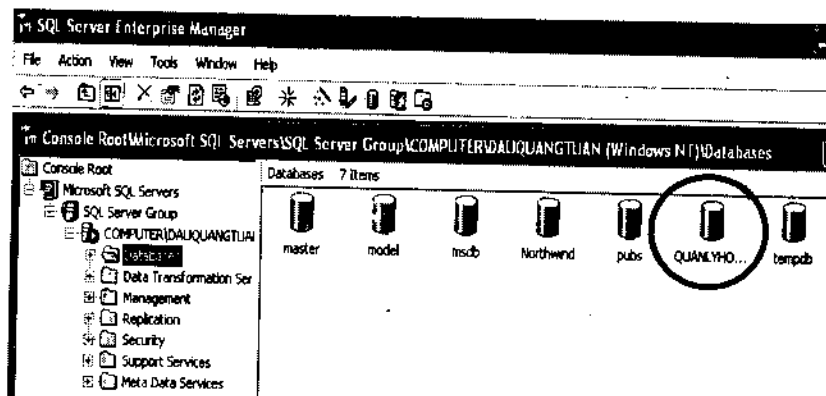
H258

◆ Nhập tên **QUANLYHOCSINH** trong khung Name như hình sau:



H259

◆ Click OK bạn thấy hình như sau:



H260

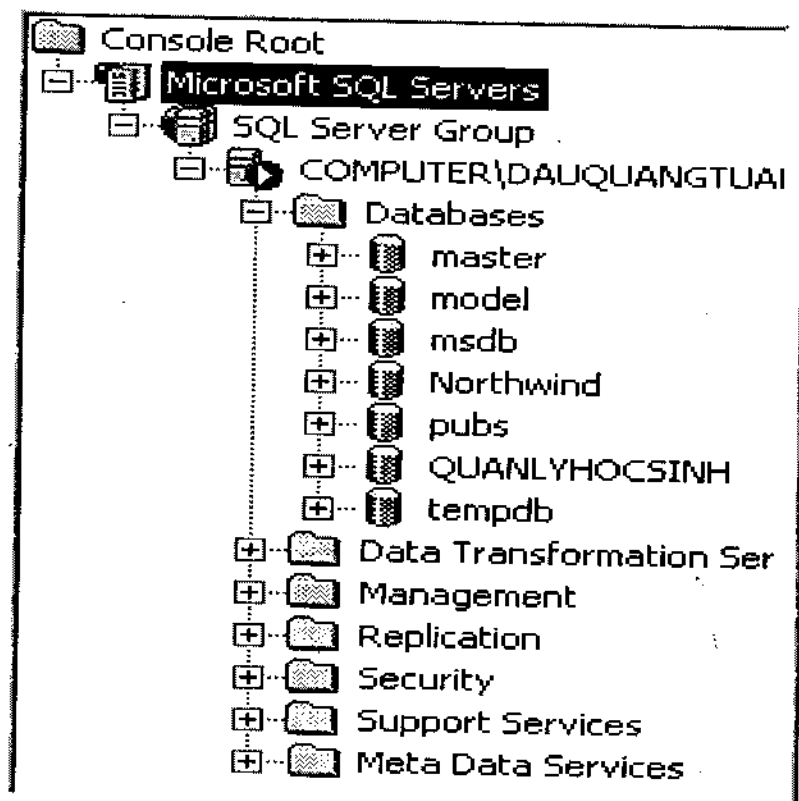
Bạn thấy có CSDL QUANLYHOCSINH như hình trên.

✎ Xóa cơ sở dữ liệu

Để xóa một cơ sở dữ liệu trong SQL Server 2000, bạn có thể sử dụng một trong hai cách đó là dùng phát biểu SQL hay dùng trình điều khiển Enterprise Manager (EM), các bạn nên dùng trình điều khiển EM cho tiện bằng cách:

◆ Vào menu Start, di chuyển vệt sáng đến Programs, di chuyển đến Microsoft SQL Server, chọn Enterprise Manager.

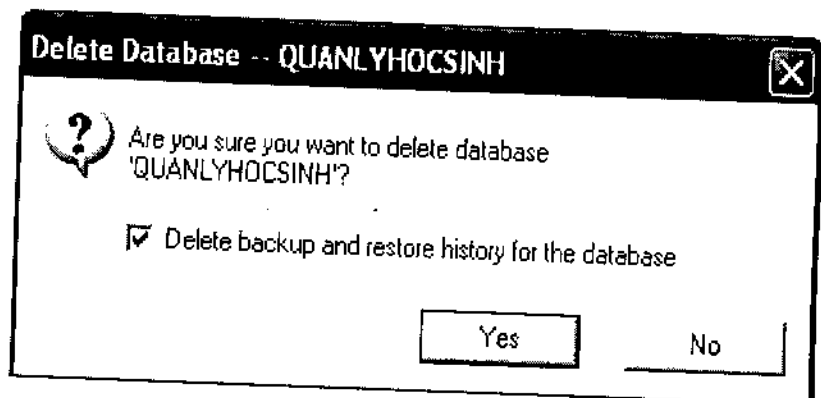
◆ Click dấu (+) trước các mục có hiện dấu (-) hiện thời trong hình dưới để thấy tên cơ sở dữ liệu cần xóa, trong trường hợp này là QUANLYHOCSINH



H261

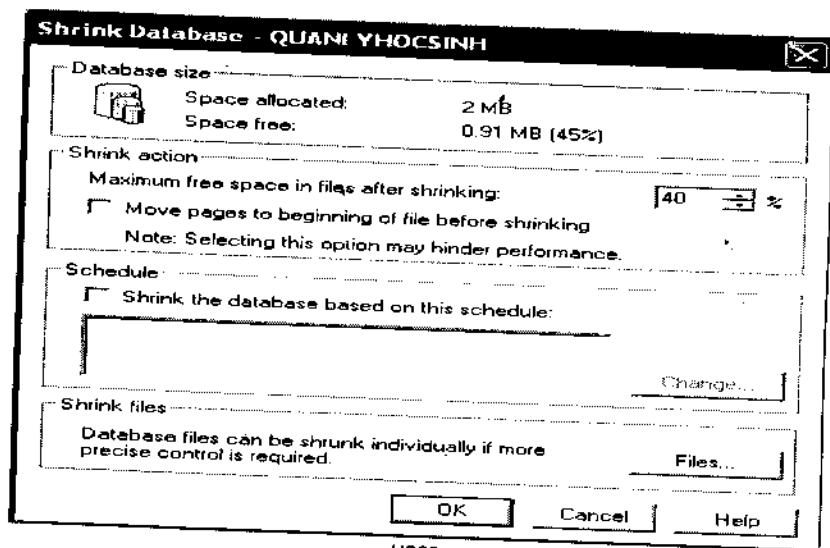
◆ Click nút phải chuột trên tên CSDL

◆ Chọn Delete trong menu hiện ra, bạn thấy hộp thoại sau:



H262

- ◆ Chọn Yes để xóa tất cả những tập tin dự phòng của CSDL
- ◆ Giảm dung lượng dư thừa của cơ sở dữ liệu
- ◆ Vào menu **Start**, di chuyển vệt sáng đến **Programs**, di chuyển đến **Microsoft SQL Server**, chọn **Enterprise Manager**.
- ◆ Chọn tên CSDL cần xóa
- ◆ Click nút phải chuột trên tên CSDL
- ◆ Di chuyển vệt sáng đến mục **All Tasks**, click chọn **Shrink Database**, bạn sẽ thấy hình sau:



H263

Bạn nhập giá trị phần trăm dung lượng lớn nhất cho phép sau khi giảm dung lượng trong phần Shrink action như hình trên.

🔧 Tạo bảng dữ liệu

Khi tạo một bảng, các bạn cần xác định:

- ◆ Tên bảng
- ◆ Kiểu dữ liệu của các vùng, vùng nào chấp nhận giá trị Null
- ◆ Khóa chính
- ◆ Khóa ngoại
- ◆ Các ràng buộc

Trước khi tạo bảng các bạn cần biết những điều sau đây:

① SQL Server 2000 gồm một số kiểu dữ liệu như sau:

● **Dữ liệu ký tự (character):** lưu trữ các chữ, kí hiệu và các ký tự số

Ví dụ:

"Laptrinh", "Tin hoc123", "234", "T*A145"

Thường dùng ba loại đó là Kiểu char, varchar và text

◆ Kiểu dữ liệu **char**: Kích thước tính bằng byte không quá 8.000 byte, chiều dài khai báo là cố định, bộ nhớ luôn cấp đúng dung lượng như khi bạn khai báo, dù khi nhập ít hơn, nó vẫn cung cấp đủ. Ví dụ bạn khai báo là 50, nhưng nhập 6 thì bộ nhớ vẫn cấp là 50 byte. Do đó khi xử lý dữ liệu bạn cần loại bỏ những ký tự trắng phía bên phải.

◆ Kiểu dữ liệu **varchar**: chiều dài thay đổi cũng tính bằng byte, không vượt quá 8.000 byte. Khác với kiểu char là bộ nhớ sẽ cấp đúng như giá trị bạn nhập. Ví dụ bạn khai báo là 50, nhưng nhập 6 thì bộ nhớ chỉ cấp 6 byte.

◆ Kiểu dữ liệu **text**: Nếu dữ liệu lớn vượt quá hai loại trên, bạn dùng kiểu này.

● **Dữ liệu kiểu nhị phân (binary):** gồm các số hệ thập lục phân (hệ 16) còn gọi là hệ hexadecimal

Ví dụ:

Số 118 hệ 10 là 76 hệ thập lục phân

$76 \text{ (Hệ 16)} = 7 \cdot 16^1 + 6 \cdot 16^0 = 112 + 6 = 118 \text{ (Hệ 10)}$

Dữ liệu kiểu binary gồm binary, varbinary và image

◆ Kiểu dữ liệu **Binary**: có chiều dài cố định là 8KB

◆ Kiểu dữ liệu **varbinary**: lưu trữ con số hệ 16 có chiều dài cố định là 8KB

◆ Kiểu dữ liệu **image**: lưu trữ chiều dài thay đổi 8KB, như văn bản Word, bảng tính Excel và các tập tin hình, đồ họa.

● Dữ liệu **Date** và **Time**: gồm dữ liệu kiểu Date hoặc kết hợp cả hai Date và Time.

Ví dụ:

"12/12/04 10:14:00:00:00 AM", "10/24/04"

Gồm hai kiểu **datetime** và **smalldatetime**

◆ Kiểu **datetime**: lưu trữ dữ liệu date từ January 1, 1753 đến December 31, 9999 (phải nhập đủ 8 byte)

◆ Kiểu **smalldatetime**: lưu trữ dữ liệu từ January 1, 1900 đến June 6, 2079 (yêu cầu 4 byte cho mỗi giá trị)

● Dữ liệu số (**Numeric**): chỉ bao gồm số âm và số dương, số lẻ, thập phân và số nguyên

◆ Dữ liệu kiểu **integer** bao gồm số nguyên âm và dương và được lưu trữ dạng: **bigint**, **int**, **smallint**, **tinyint** (xếp theo thứ tự giảm dần)

■ Kiểu **bigint**: từ -9,223,372,036,854,775,808

đến 9,223,372,036,854,775,808 (trong 8 byte)

■ Kiểu **int**: từ -2,147,483,648 đến 2,147,483,647 (trong 4 byte)

■ Kiểu **smallint**: từ -32,768 đến 32,767 (trong 2 byte)

■ Kiểu **tinyint**: từ 0 đến 255 (trong 1 byte)

◆ Dữ liệu **Decimal**: bao gồm phần nguyên và phần các số lẻ. Số bytes yêu cầu để lưu trữ số decimal hay numeric phụ thuộc vào tổng số con số trước và sau dấu phân cách số lẻ.

● Dữ liệu kiểu **Unicode**: lưu trữ các chữ, ký hiệu và ký tự số dạng Unicode với các kiểu **nchar**, **nvarchar** và **ntext** (n viết tắt chữ National)

◆ Kiểu dữ liệu **nchar**: kích thước tính bằng ký tự không quá 4.000 ký tự, chiều dài khai báo là cố định, bộ nhớ luôn cấp đúng dung lượng như khi bạn khai báo, dù khi nhập ít hơn, nó vẫn cung cấp đủ. Ví dụ bạn khai báo là 50, nhưng nhập 6 thì bộ nhớ vẫn cấp là 50 ký tự. Do đó khi xử lý dữ liệu bạn cần loại bỏ những ký tự trắng phía bên phải.

◆ Kiểu dữ liệu **nvarchar**: chiều dài thay đổi cũng tính bằng ký tự, không vượt quá 4.000 ký tự. Khác với kiểu **nchar** là bộ nhớ sẽ cấp đúng như giá trị bạn nhập. Ví dụ bạn khai báo là 50, nhưng nhập 6 thì bộ nhớ chỉ cấp 6 ký tự.

◆ Kiểu dữ liệu **ntext**: Nếu dữ liệu lớn vượt quá hai loại trên, bạn dùng kiểu này.

Ghi chú:

Mỗi ký tự Unicode cần 2 bytes để mã hóa.

● Dữ liệu kiểu tiền tệ (Monetary): Dùng để trình bày số dạng tiền tệ với số âm hay dương bao gồm hai kiểu dữ liệu là money và smallmoney

◆ Kiểu dữ liệu money: lưu trữ số tiền tệ từ -922,337,203,685,477.5808 đến +922,337,203,685,477.5807 (8 byte cho mỗi giá trị)

◆ Kiểu dữ liệu smallmoney: Dùng để lưu trữ số tiền tệ khoảng từ -214,748.3648 đến 214,748.3647 (4 byte cho mỗi giá trị)

② Số tự động trong SQL Server 2000

Trong một bảng, chúng ta có thể khai báo một cột duy nhất có giá trị là số tự động (identity), tăng dần theo số mẫu tin thêm vào.

Ghi chú:

Cần quan tâm đến số bắt đầu (identity seed) và mỗi lần nhảy bao nhiêu giá trị (identity increament), giá trị mặc nhiên, giá trị ban đầu là 1 và giá trị nhảy là 1. Khi khai báo cột tự động bạn cần lưu ý những điều sau:

◆ Kiểu dữ liệu cho cột này phải thuộc một trong các kiểu: decimal, int, numeric, smallint, bigint hay tinyint.

◆ Phải chỉ định seed, increment và giá trị mặc định của hai tham số này là 1

◆ Cột khai báo số tự động không cho phép Null

◆ Hàm Objectproperty có thể sử dụng trong quá trình lập trình nếu bảng có khai báo cột Identity và hàm Columnproperty có thể sử dụng để biết tên của cột khai báo Identity

③ Khóa và các ràng buộc

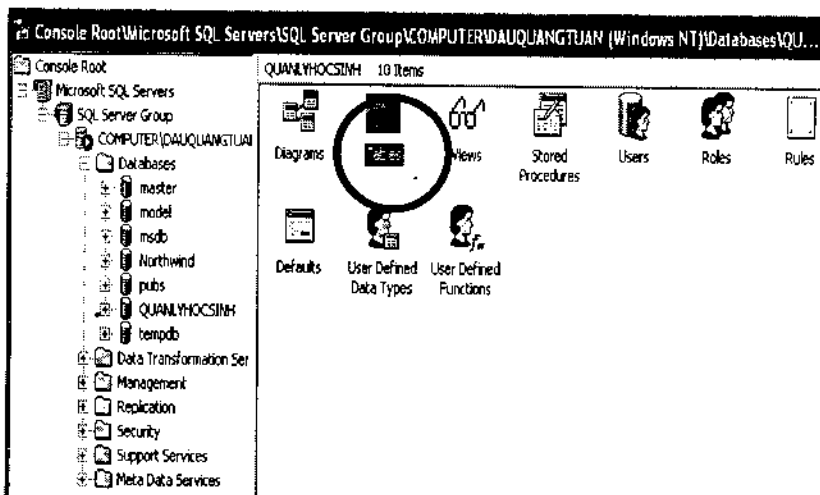
Phần này các bạn có thể xem lại ở phần CSDL Access 2003 ở trên.

☐ Đến đây bạn có thể bắt tay vào tạo bảng dữ liệu, chúng tôi hướng dẫn các bạn tạo bảng bằng cách dùng trình điều khiển Enterprise Manager.

☐ Vào menu Start, di chuyển vệt sáng đến Programs, di chuyển vệt sáng đến Microsoft SQL Server 2000, click chọn Enterprise Manager.

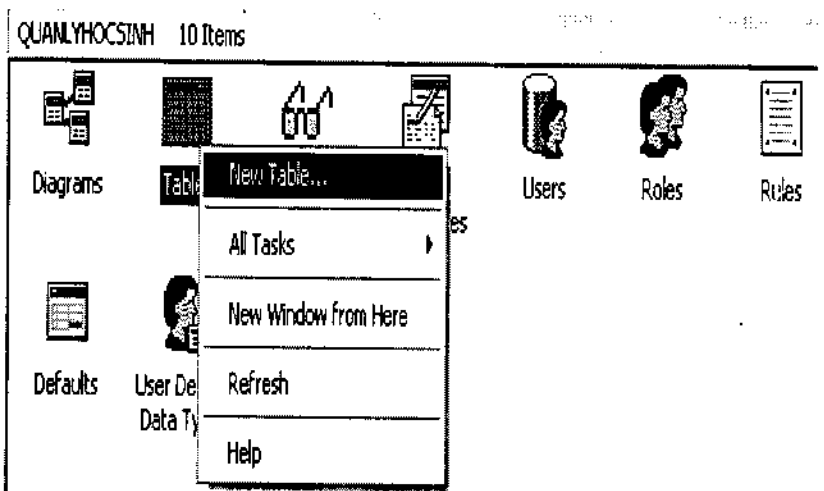
☐ Trong cửa sổ hiện ra, nhìn phía bên trái, bạn click dấu "+" lần lượt trước các mục có dấu "-" như hình bên (các mục có dấu "-" khi chưa click sẽ là dấu "+") cho đến khi thấy tên cơ sở dữ liệu QUANLYHOCSINH mà bạn đã tạo ở trên.

☐ Click chọn tên SCDL QUANLYHOCSINH, bạn thấy thông tin có sẵn như hình sau:



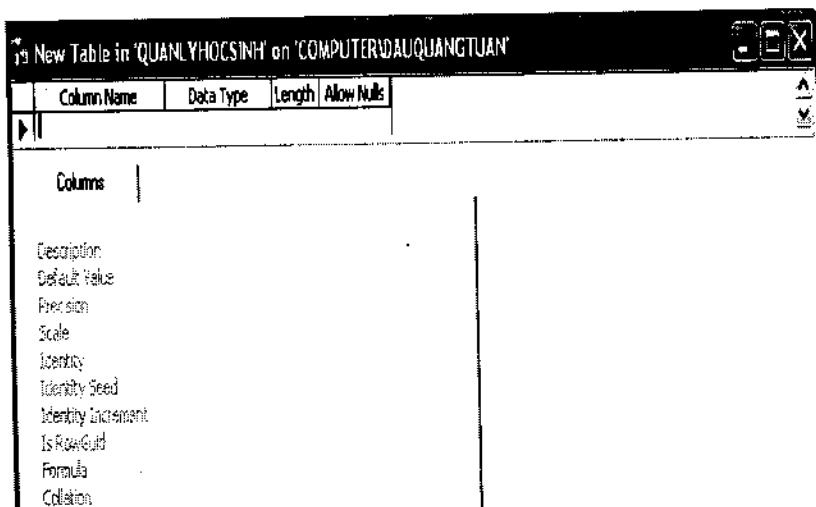
H264

☐ Trong các đối tượng có bên trong CSDL QUANLYHOCSINH, có Tables. Click nút phải chuột trên đối tượng này



H265

☐ Click chọn New Table, các bạn sẽ thấy cửa sổ thiết kế bảng như hình sau:



H266

◆ Bạn bắt đầu thiết kế cho bảng HOCSINH, ví dụ bảng học sinh có cấu trúc như sau:

Cấu trúc của bảng HOSINH

Column Name	Data type	Length	Allow Nulls
MSHS	Char	5	
Holot	Varchar	30	
Ten	Char	7	
Lop	Char	4	
Nam	Char	3	✓
Ngaysinh	Datetime	8	✓
Noisinh	Varchar	40	✓
Diachi	Varchar	40	✓
Tencha	Varchar	40	✓
Tenme	Varchar	40	✓
DT	Char	12	✓

Bạn nhập giá trị tương ứng trong các cột vào bảng để có như hình sau:

Design Table 'HOCSINH' in 'QUANLYHOCSINH' on 'COMPUTERIDAUQUANGTUAN'

Column Name	Data Type	Length	Allow Nulls
MSHS	char	5	
Holot	varchar	30	
Ten	char	7	
Lop	char	4	
Nam	char	3	
Ngaysinh	datetime	8	
Noisinh	varchar	40	
Diachi	varchar	40	
Tenchu	varchar	40	
Tenme	varchar	40	
DT	char	12	

H267

◆ Trong bảng này vùng khóa là MSHS, bạn click chuột tại ô bên trái của MSHS để chọn dòng này

Design Table 'HOCSINH' in 'QUANLYHOCSINH' on 'COMPUTERIDAUQUANGTUAN'

Column Name	Data Type	Length	Allow Nulls
MSHS	char	5	
Holot	varchar	30	
Ten	char	7	
Lop	char	4	
Nam	char	3	
Ngaysinh	datetime	8	
Noisinh	varchar	40	
Diachi	varchar	40	
Tenchu	varchar	40	
Tenme	varchar	40	
DT	char	12	

H268

◆ Click chọn icon có hình chiếc chìa khóa trên thanh công cụ, bạn sẽ được như hình sau:

	Column Name	Data Type	Length	Allow Nulls
	MSHS	char	5	
	Holot	varchar	30	
	Ten	char	7	
	Lop	char	4	
	Nam	char	3	
	Ngaysinh	datetime	8	
	Moisinh	varchar	40	
	Diachi	varchar	40	
	Tencha	varchar	40	
	Tenme	varchar	40	
	QT	char	12	

H269

Bạn thấy hình chiếc chìa khóa xuất hiện bên cạnh MSHS, nếu cột Allow Nulls của vùng MSHS có dấu chọn thì bây giờ sẽ biến mất. Ngoài những cột bắt buộc có giá trị như vùng khóa chính, vùng khóa ngoại, ..., những vùng khác có thể cho phép giá trị Null mỗi khi mẫu tin được thêm vào.

Nếu bạn muốn diễn giải cho vùng này, bạn có thể nhập vào khung có tiêu đề Description ở bên dưới, bạn có thể nhập giá trị mặc định cho vùng trong khung Default Value. Nếu muốn khai báo vùng dữ liệu là số tự động, bạn chọn Yes cho ô Identity và giá trị khởi tạo, giá trị tăng cho ô Identity Seed và Identity Inreament. Bạn có thể dùng kết quả của các vùng khác qua công thức tính toán trong ô Formula.

Lưu ý:

- ◆ Những vùng khai báo Identity thì không cho phép nhập liệu
- ◆ Vùng khai báo giá trị mặc định Default, bạn có thể nhập hoặc không
- ◆ Nếu khóa chính là tổ hợp của nhiều vùng, bạn nhấn phím Shift và click chọn trước các vùng, sau đó click chọn hình chiếc chìa khóa trên thanh công cụ.

Để bỏ khóa, bạn click chọn vùng chứa hình chiếc chìa khóa, click icon chứa hình chiếc chìa khóa trên thanh công cụ.

Giả sử bạn click chọn vùng MSHS, click hình chiếc chìa khóa trên thanh công cụ để bỏ vùng khóa. Nhấn phím Shift và click chuột chọn ba vùng MSHS, Holot và Ten như hình sau:

Design Table 'HOCSINH' in 'QUANLYHOCSINH' on 'COMPUTERDAUQUANGTUAN'

	Column Name	Data Type	Length	Allow Nulls
PK	MSHS	char	5	
	Holot	varchar	30	
	Ten	char	7	
	Lop	char	4	
	Nam	char	3	
	Ngaysinh	datetime	8	
	Noisinh	varchar	40	
	Diachi	varchar	40	
	Tencha	varchar	40	
	Tenme	varchar	40	
	DT	char	12	

H270

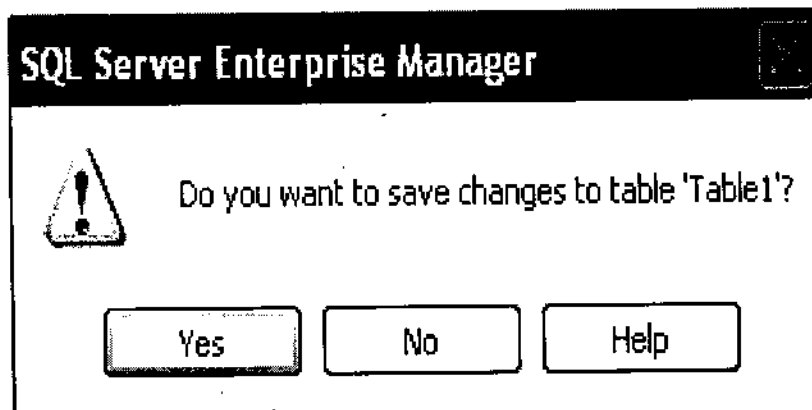
Click hình chiếc chìa khóa trên thanh công cụ, bạn sẽ có hình như sau:

Design Table 'HOCSINH' in 'QUANLYHOCSINH' on 'COMPUTERDAUQUANGTUAN'

	Column Name	Data Type	Length	Allow Nulls
PK	MSHS	char	5	
	Holot	varchar	30	
	Ten	char	7	
	Lop	char	4	
	Nam	char	3	
	Ngaysinh	datetime	8	
	Noisinh	varchar	40	
	Diachi	varchar	40	
	Tencha	varchar	40	
	Tenme	varchar	40	
	DT	char	12	

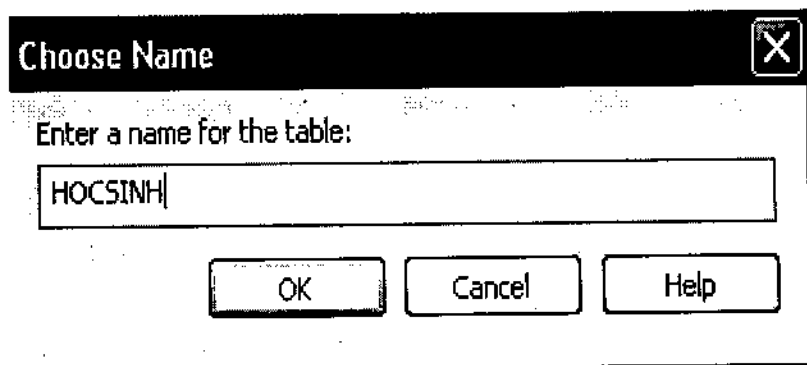
H271

◆ Bạn đổi lại vì bảng HOCSINH này có khóa chính là MSHS. Khi hoàn thành việc thiết kế, bạn click nút X để đóng cửa sổ thiết kế, hộp thoại hiện ra hỏi bạn, click nút Yes



B2H23

☒ Nhập vào HOCSINH trong khung



H272

◆ Click nút OK

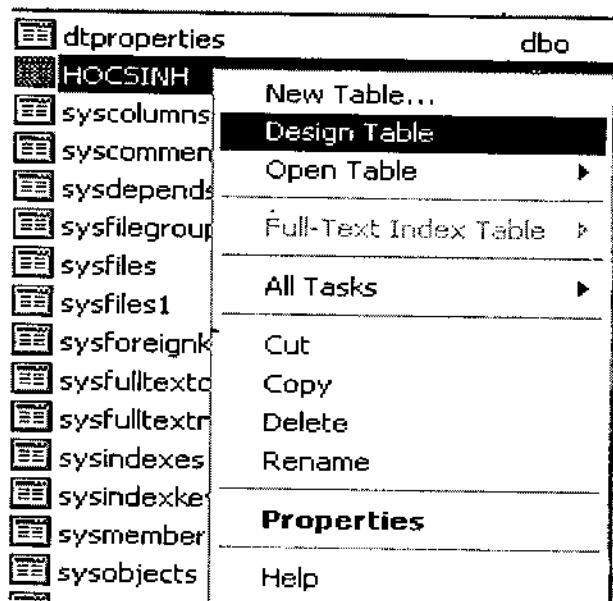
◆ Bạn vào lại CSDL QUANLYHOCSINH, click dấu + phía trước, chọn Tables, bạn sẽ thấy có bảng HOCSINH như hình sau:

Console Root		Tables: 21 items			
		Name /	Owner	Type	Create Date
Microsoft SQL Servers					
SQL Server Group					
COMPUTER\DAUQUANGTUAN					
Databases					
master		dtproperties	dbo	System	4/22/2005 5:23:55 PM
model		HOCSINH	dbo	User	4/22/2005 6:11:40 PM
model		syscolumns	dbo	System	8/6/2000 1:29:12 AM
model		syscomments	dbo	System	8/6/2000 1:29:12 AM
model		sysdepends	dbo	System	8/6/2000 1:29:12 AM
Northwind		sysfilegroups	dbo	System	8/6/2000 1:29:12 AM
pubs		sysfiles	dbo	System	8/6/2000 1:29:12 AM
QUANLYHOCSINH		sysfilest	dbo	System	8/6/2000 1:29:12 AM
Diagrams		sysforeignkeys	dbo	System	8/6/2000 1:29:12 AM
		sysfulltextcatalogs	dbo	System	8/6/2000 1:29:12 AM
Views		sysfulltextnotify	dbo	System	8/6/2000 1:29:12 AM
Stored Procedure		sysindexes	dbo	System	8/6/2000 1:29:12 AM
Users		sysindexkeys	dbo	System	8/6/2000 1:29:12 AM
Roles		sysmembers	dbo	System	8/6/2000 1:29:12 AM
Rules		sysobjects	dbo	System	8/6/2000 1:29:12 AM
Defaults		syspermissions	dbo	System	8/6/2000 1:29:12 AM
User Defined Dal		sysproperties	dbo	System	8/6/2000 1:29:12 AM
User Defined Fur		sysprotects	dbo	System	8/6/2000 1:29:12 AM

H273

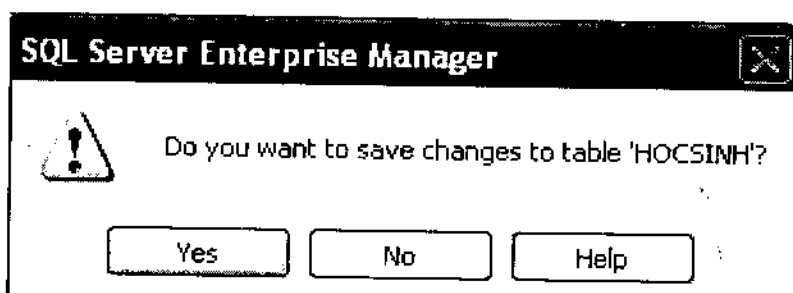
Sửa cấu trúc của bảng

Muốn sửa lại cấu trúc của bảng HOCSINH, bạn vào để thấy bảng HOCSINH như hình trên, click nút phải chuột trên tên HOCSINH, chọn Design Table trong menu hiện ra



H274

Cửa sổ thiết kế hiện ra, bạn chỉnh sửa, thực hiện xong, bạn click dấu X để đóng cửa sổ thiết kế lại, bạn sẽ thấy hộp thoại hiện ra hỏi chúng ta có muốn lưu những gì vừa thay đổi không, nếu click Yes thì sẽ lưu với nội dung mới, nếu click No thì vẫn giữ những giá trị cũ trước đó.

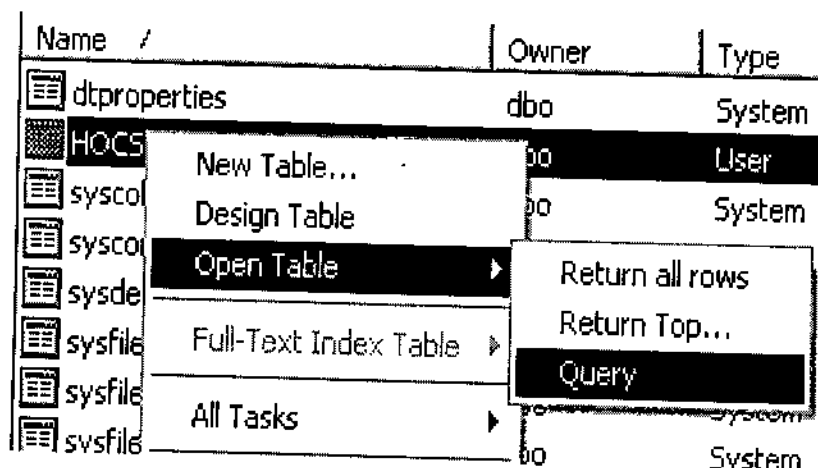


H275

➤ Nhập dữ liệu cho bảng

Để nhập dữ liệu vào cho bảng HOCSINH, bạn cho tên bảng hiện ra như hình trên, click nút phải chuột tại tên HOCSINH, đưa vệt sáng đến

Open Table có ba lựa chọn hiện ra, đó là **Return all rows**, **Return Top** hoặc **Query**



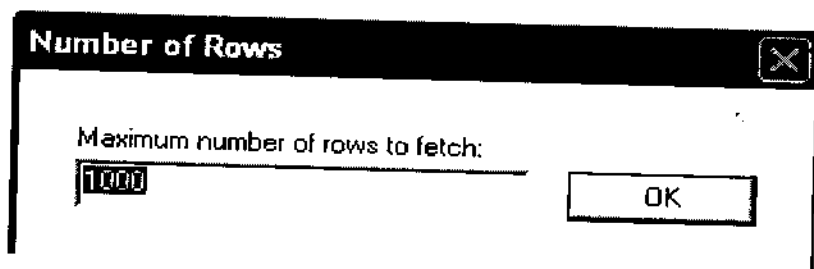
H276

◆ **Return all rows:** Chọn mục này sẽ mở bảng với tất cả mẫu tin đang có

◆ **Return Top:** Nếu bảng có số mẫu tin nhiều, bạn có thể chọn mục này để trình bày một số mẫu tin chỉ định

◆ **Query:** Sử dụng khung nhìn truy vấn để lọc một số mẫu tin theo tiêu chuẩn cho trước.

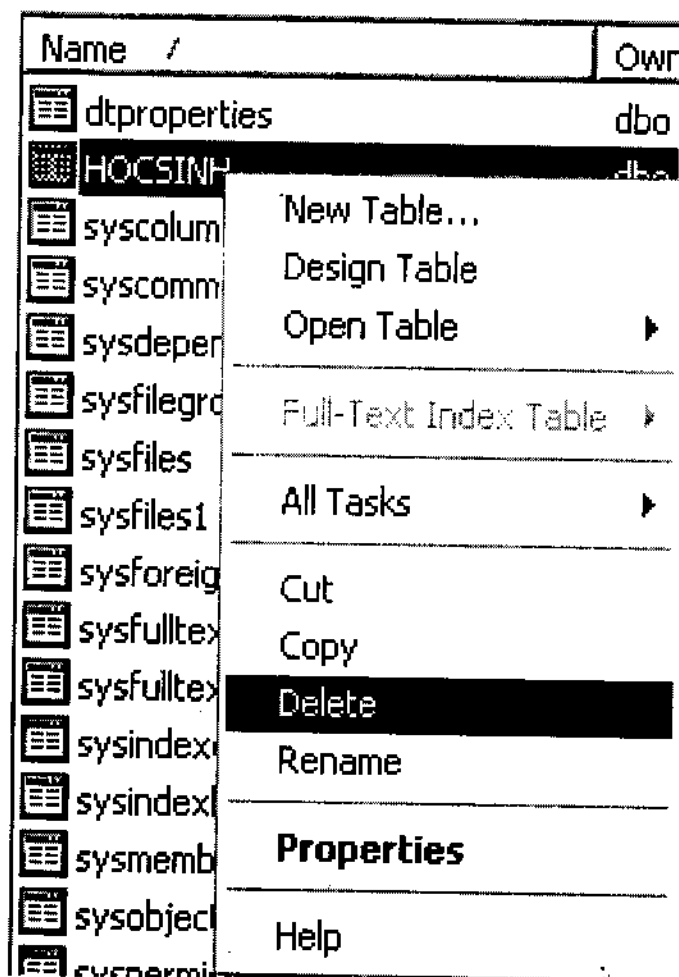
Nếu bạn nhập mục thứ hai **Return Top**, hộp thoại hiện ra với giá trị mặc nhiên là 1000, bạn nhập số mẫu tin cần xuất hiện trong khung này



H277

✶ Xóa bảng dữ liệu

◆ Vào Enterprise Manager, click nút chuột phải trên tên bảng, trong menu hiện ra chọn **Delete**



H278

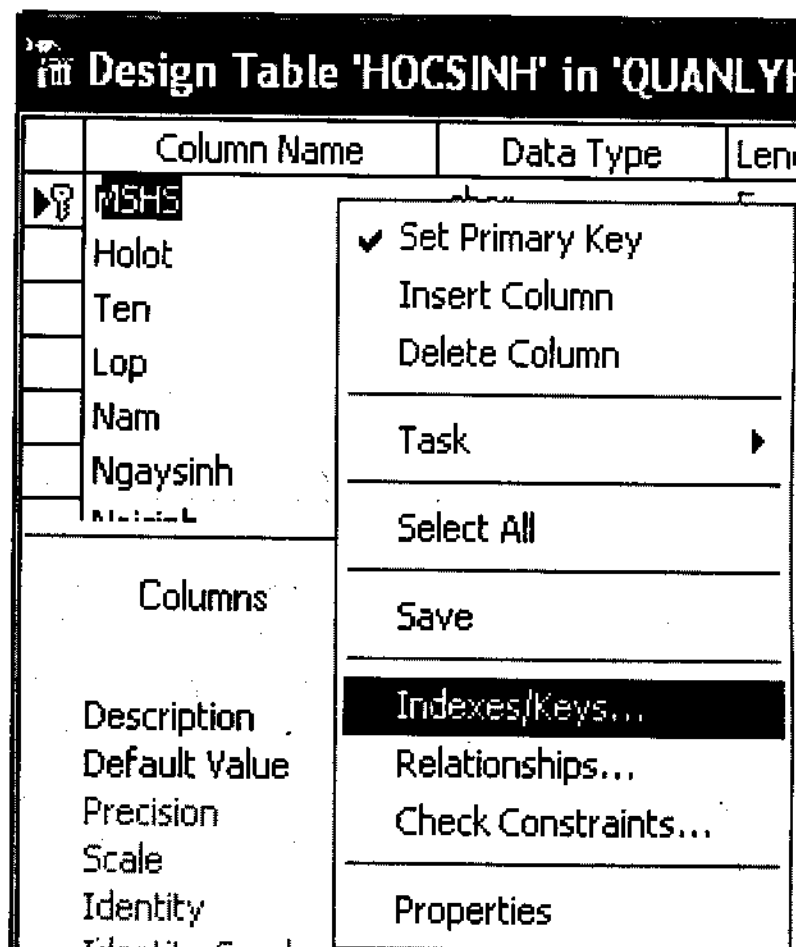
☛ Đổi tên bảng dữ liệu

◆ Vào Enterprise Manager, click nút chuột phải trên tên bảng; trong menu hiện ra chọn Rename (xem hình trên)

☛ Tạo chỉ mục (Indexes)

Trong CSDL, một chỉ mục cho phép chương trình CSDL tìm kiếm dữ liệu trong bảng mà không cần duyệt qua các mẫu tin đang có, cách tạo index bằng Enterprise Manager như sau:

◆ Vào cửa sổ thiết kế (Design view) của bảng, click nút phải chuột trên vùng cần tạo chỉ mục, trong menu hiện ra chọn Indexes/Keys như hình sau:



H279

◆ Click xong bạn sẽ thấy hộp thoại hiện ra như hình sau:

Properties

Tables | Relationships | Indexes/Keys | Check Constraints

Table name:

Selected index:

Type:

Index name:

Column name	Order
MSHS	Ascending

Index Filegroup:

- ☒ Create UNIQUE
- ☐ Constraint
- ☐ Index ☐ Ignore duplicate key

Fill factor:

%

☐ Pad Index

☒ Create as CLUSTERED

☐ Do not automatically recompute statistics

Close

Help

H280 Chọn Index cho MSHS

◆ Tương tự như vậy, các bạn tạo các bảng KHENTHUONG, KYLUAT, DIEMTB và DIEMTHILAI cho CSDL QUANLYHOCSINH như cấu trúc sau:

Cấu trúc của bảng KHENTHUONG

Column Name	Data type	Length	Allow Nulls
MSKT	Char	5	
Ndkhen	Varchar	50	
NgKhen	DateTime	8	
Phanthuong	Varchar	50	
KTMSHS	Char	5	

New Table in 'QUANLYHOC SINH' on 'COMPUTER\DAUQUANGTUAN'

Column Name	Data Type	Length	Allow Nulls
MSKT	char	5	
Ngkhen	varchar	50	
Ngkhen	datetime	8	
Phanthuong	varchar	50	
KTMSHS	char	5	

Columns

Description

Default Value

Precision 0

Scale 0

Identity No

Identity Seed

Identity Increment

Is RowGuid No

Formula

Collation <database default>

H281

SQL Server Enterprise Manager

Do you want to save changes to table 'Table1'?

Yes No Help

H282

Click nút Yes

Choose Name

Enter a name for the table:

KHENTHUONG

OK Cancel Help

H283 Đặt tên là KHENTHUONG

- ◆ Click nút OK
- ◆ Tiếp tục tạo bảng KYLUAT theo cấu trúc sau:

Cấu trúc của bảng KYLUAT

Column Name	Data type	Length	Allow Nulls
MSKL	Char	5	
Vipham	Varchar	50	
Ngvipham	DateTime	8	
Xuly	Varchar	50	
KyluatMSHS	Char	5	

New Table in 'QUANLYHOCSINH' on 'COMPUTERDAUQUANGTUAN'

	Column Name	Data Type	Length	Allow Nulls
☒	MSVP	char	5	
☐	Vipham	varchar	50	
☐	Ngvipham	datetime	8	
☐	Xuly	varchar	50	
☐	KyluatMSHS	char	5	

Columns

Description	
Default Value	
Precision	0
Scale	0
Identity	No
Identity Seed	
Identity Increment	
Is RowGuid	No
Formula	

H284

- ◆ Click dấu X đóng bảng, đặt tên cho bảng là KYLUAT như hình sau:

Choose Name

Enter a name for the table:

KYLUAU

OK

Cancel

Help

H285

Cấu trúc của bảng DIEMTB

Column Name	Data type	Length	Allow Nulls
MSKQ	Char	5	
Van1	Real	4	
Van2	Real	4	
Vancn	Real	4	
Toan1	Real	4	
Toan2	Real	4	
Toancn	Real	4	
Ngoaing1	Real	4	
Ngoaing2	Real	4	
Ngoaingcn	Real	4	
Vatly1	Real	4	
Vatly2	Real	4	
Vatlycn	Real	4	
Hoa1	Real	4	
Hoa2	Real	4	
Hoacn	Real	4	
Sinh1	Real	4	
Sinh2	Real	4	
Sinhcn	Real	4	
Congnghe1	Real	4	
Congnghe2	Real	4	
Congnghecn	Real	4	

Su1	Real	4	
Su2	Real	4	
Sucn	Real	4	
Dia1	Real	4	
Dia2	Real	4	
Diacn	Real	4	
GDCD1	Real	4	
GDCD2	Real	4	
GDCDcn	Real	4	
Theduc1	Real	4	
Theduc2	Char	2	
Theduccn	Char	2	
Nhac1	Char	2	
Nhac2	Char	2	
Nhaccn	Char	2	
Mythuat1	Char	2	
Mythuat2	Char	2	
Mythuatcn	Char	2	
HoclucHK1	Char	10	
HoclucHK2	Char	10	
Hoclucn	Char	10	
HanhkiemHK1	Char	10	
HanhkiemHK2	Char	10	
Hanhkiemcn	Char	10	
SongayngghiHK1	Tinyint	1	
SongayngghiHK2	Tinyint	1	
Songayngghicn	Tinyint	1	
DanhhieuhHK1	Char	10	
DanhhieuhHK2	Char	10	
Danhhieucn	Char	10	
Ketquacuoinam	Char	20	
DiemMSHS	Char	5	

Design Table 'DIEMTB' in 'QUANLYHOCSINH' on 'COMPUTERIDAUQUANGTUAN'

Column Name	Data Type	Length	Allow Nulls
MSKQ	char	5	
Van1	real	4	
Van2	real	4	
Vanch	real	4	
Toan1	real	4	
Toan2	real	4	
Toanch	real	4	
Ngoaingiu1	real	4	
Ngoaingiu2	real	4	
Ngoaingiuch	real	4	
Vatly1	real	4	
Vatly2	real	4	
Vatlych	real	4	
Hoa1	real	4	
Hoa2	real	4	

H286

Click nút đóng bảng và lưu với tên DIEMTB như hình sau:

Choose Name

Enter a name for the table:

DIEMTB

OK

Cancel

Help

H287

Cấu trúc của bảng DIEMTHILAI

Column Name	Data type	Length	Allow Nulls
<u>MSTL</u>	Char	5	

VanT	Real	4	
VanS	Real	4	
ToanT	Real	4	
ToanS	Real	4	
NgoainguiT	Real	4	
NgoainguiS	Real	4	
VatlyT	Real	4	
VatlyS	Real	4	
HoaT	Real	4	
HoaS	Real	4	
SinhT	Real	4	
SinhS	Real	4	
CongngheT	Real	4	
CongngheS	Real	4	
SuT	Real	4	
SuS	Real	4	
ĐiaT	Real	4	
ĐiaS	Real	4	
GDCDT	Real	4	
GDCDS	Real	4	
TheducT	Char	2	
TheducS	Char	2	
NhacT	Char	2	
NhacS	Char	2	
MythuatT	Char	2	
MythuatS	Char	2	
HoclucT	Char	10	
HanhkiemT	Char	10	
HoclucS	Char	10	
HanhkiemS	Char	10	
ThilaiMSHS	Char	5	

New Table in 'QUANLYHOCSINH' on 'COMPUTERDAUQUANGTUAN'

	Column Name	Data Type	Length	Allow Nulls
☒	MSTL	char	5	
☐	YanT	real	4	
☐	YanS	real	4	
☐	ToanT	real	4	
☐	ToanS	real	4	
☐	Ngoaingiut	real	4	
☐	Ngoaingus	real	4	
☐	VatlyT	real	4	
☐	VatlyS	real	4	
☐	...	...	...	

Columns

Description	Default Value	Precision	Scale	Identity
		0	0	No

H288

Choose Name

Enter a name for the table:

DIEMTHILAI

OK

Cancel

Help

H289

◆ Khi tạo xong, trong CSDL QUANLYHOCSINH, chúng ta sẽ thấy trong DataBase có năm bảng như hình sau:

Console Root\Microsoft SQL Servers\SQL Server

Console Root	Tables 25 Items
Microsoft SQL Servers	Name /
SQL Server Group	
COMPUTER\DAUQUANGTUAI	
Databases	
master	DIEMTB
model	DIEMTHILAI
msdb	dtproperties
Northwind	HOC SINH
pubs	KHENTHUONG
QUANLYHOC SINH	KYLUAT
Diagrams	syscolumns
Tables	syscomments
	sysdepends
	sysfilegroups

H290

Đối tượng DIAGRAM

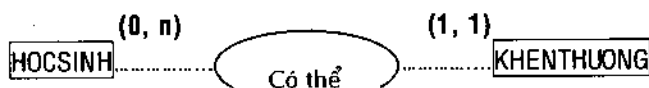
Dùng để tạo mối quan hệ giữa các bảng trong cơ sở dữ liệu, trước hết bạn cần xét mối quan hệ giữa các bảng hiện có trong CSDL QUANLYHOC SINH

Xét hai bảng HOC SINH và KHENTHUONG

Một học sinh có thể không được khen thưởng lần nào hoặc được khen thưởng nhiều lần.

Một khen thưởng nào đó thì chỉ thuộc một học sinh nào đó được nhận mà thôi.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:



Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ KHENTHUONG qua HOCSINH.

Khi có mối liên hệ một nhiều giữa hai table thì bạn nhớ là khoá ngoại sẽ nằm trong table có giá trị là (0, 1) hoặc (1, 1).

Trong hai Table trên, bạn thấy khoá ngoại sẽ nằm trong table KHENTHUONG, đó là field KTMSHS.

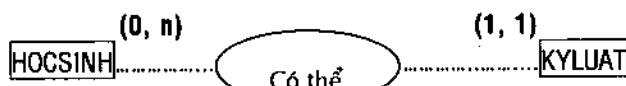
Bạn có thể tìm hiểu thêm trong các sách phân tích thiết kế hệ thống và cơ sở dữ liệu.

▣ Xét hai bảng HOCSINH và KYLUAT

☞ Một học sinh có thể không bị kỷ luật lần nào hoặc bị kỷ luật nhiều lần.

☞ Một vi phạm kỷ luật nào đó thì chỉ thuộc một học sinh nào đó mà thôi.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:



Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ KYLUAT qua HOCSINH.

Khi có mối liên hệ một nhiều giữa hai table thì bạn nhớ là khoá ngoại sẽ nằm trong table có giá trị là (0, 1) hoặc (1, 1).

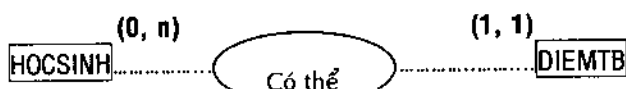
Trong hai Table trên, bạn thấy khoá ngoại sẽ nằm trong table KYLUAT, đó là field KyluatMSHS.

▣ Xét hai bảng HOCSINH và DIEMTB

☞ Một học sinh có thể có điểm trung bình của một hoặc nhiều môn.

☞ Một điểm trung bình môn nào đó thì chỉ thuộc một học sinh nào đó mà thôi.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:



Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ DIEMTB qua HOCSINH.

Khi có mối liên hệ một nhiều giữa hai table thì bạn nhớ là khoá ngoại sẽ nằm trong table có giá trị là (0, 1) hoặc (1, 1).

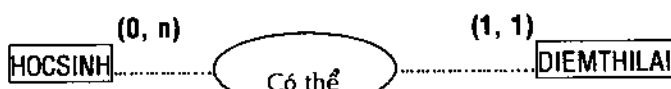
Trong hai Table trên, bạn thấy khoá ngoại sẽ nằm trong table DIEMTB, đó là field DiemMSHS.

☐ Xét hai bảng HOCSINH và DIEMTHILAI

☞ Một học sinh có thể thi lại một hoặc nhiều môn.

☞ Điểm một môn thi lại nào đó thì chỉ thuộc một học sinh nào đó mà thôi.

Bạn sẽ thấy có mối liên hệ giữa hai Table như sau:



Vậy giữa hai table trên sẽ có một mối liên kết. Liên kết là $n-1$ từ DIEMTHILAI qua HOCSINH.

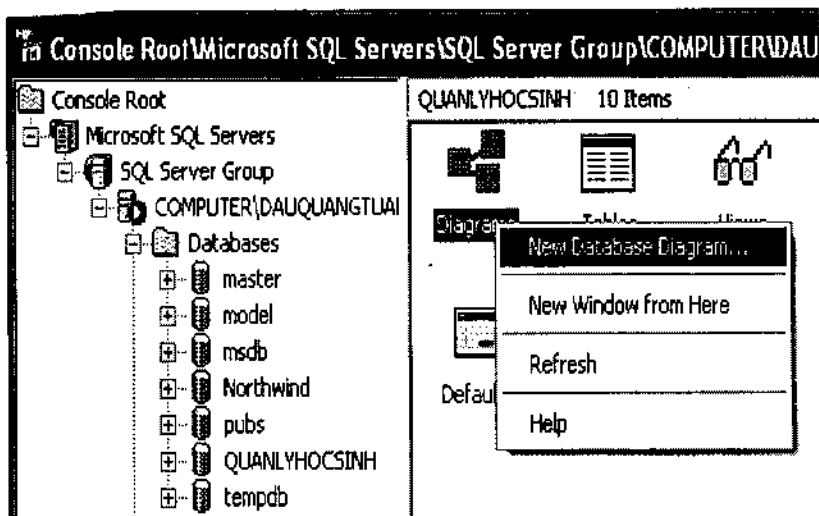
Khi có mối liên hệ một nhiều giữa hai table thì bạn nhớ là khoá ngoại sẽ nằm trong table có giá trị là (0, 1) hoặc (1, 1).

Trong hai Table trên, bạn thấy khoá ngoại sẽ nằm trong table DIEMTHILAI, đó là field ThilaiMSHS.

Bây giờ chúng ta bắt đầu vào Diagram như sau:

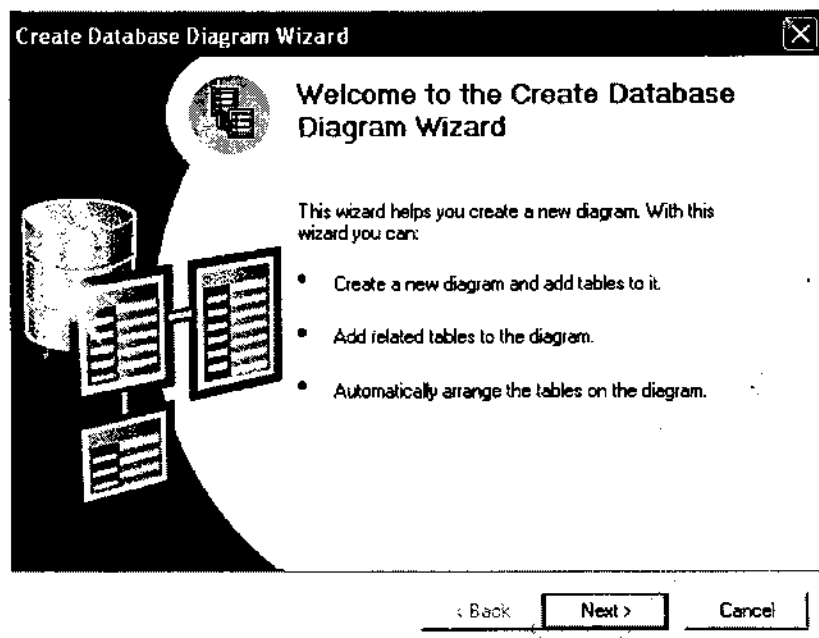
☑ Click chọn tên CSDL QUANLYHOCSINH

☑ Trong các đối tượng của CSDL có đối tượng Diagram, click nút phải chuột trên đối tượng này



H291

◆ Chọn mục **New Database Diagram** trong menu hiện ra, bạn sẽ có hình như sau:



H292

◆ Click nút Next, ta có cửa sổ sau hiện ra



Available tables

DIEMTE
DIEMTHILAI
dtproperties
HOCSINH
KHENTHUONG
KYLUA
syscolumns
syscomments
sysdepends
sysfilegroups
sysfiles

Add >

< Remove

Tables to add to diagram

☐ Add related tables automatically

How many levels of related tables?



< Back

Next >

Cancel

H293

◆ Bạn click chọn từng bảng và nhấn nút Add, nếu muốn bỏ bảng ra khỏi khung vừa đưa qua, bạn click chọn tên bảng và click nút Remove, khi đưa năm bảng qua khung bên phải, bạn sẽ có hình như sau:

Create Database Diagram Wizard

Select Tables to be Added

Select one or more tables from available tables box. You may also choose to add related tables.

Available tables

diproperties
syscolumns
syscomments
sysdepends
sysfilegroups
sysfiles
sysfiles1
sysforeignkeys
sysfulltextcatalogs
sysfulltextnotifi
sysindexes

Add >

< Remove

Tables to add to diagram

DIEMTB
DIEMTHILAI
HOCSINH
KHENTHUONG
KYLUAU

☐ Add related tables automatically

How many levels of related tables?

1

< Back

Next >

Cancel

H294

◆ Click nút Next, bạn sẽ có hình như sau:

Create Database Diagram Wizard

Completing the Create Database Diagram Wizard

You have completed the steps required to create a database diagram. You have chosen the following tables to be added and automatically arranged on the diagram:

DIEMTB
DIEMTHILAI
HOCSINH
KHENTHUONG
KYLUAU

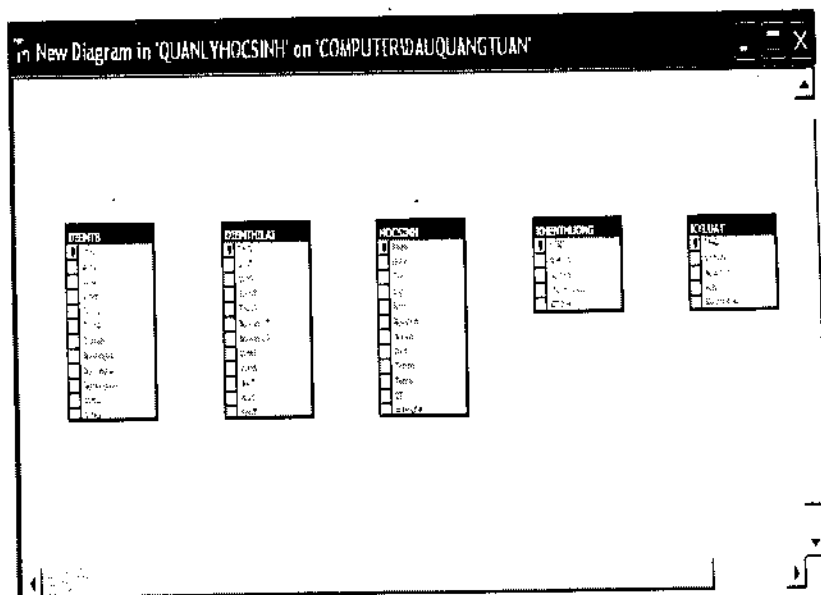
< Back

Finish

Cancel

H295

◆ Click nút Finish để kết thúc, bạn sẽ có kết quả như sau:



H296

◆ Click và drag chuột tại tiêu đề của mỗi bảng để kéo đến vị trí thích hợp, bạn nhớ để bảng HOCSINH nằm ở vị trí trung tâm.

Chú ý:

Khi vùng khóa của các bảng đang được chọn trong cửa sổ Diagram, nếu click chuột trong bảng, bạn sẽ thấy vùng này biến mất, bạn click bên ngoài bảng thì không mất.

Khi tạo mối liên hệ từ vùng của bảng này đến một vùng của bảng khác, các bạn phải làm sao để thấy được vùng này. Trong bài trên vì hai bảng DIEMTB và DIEMTHILAI có nhiều vùng, mà khóa ngoại của hai bảng này để phía sau cùng, nên hơi bất tiện, khi thực hiện, các bạn phải kéo cửa sổ Diagram lớn ra, kéo hai bảng này dài ra về phía bên dưới để thấy được vùng cần liên kết. Trong thực tế, bạn có thể tạo mục cần liên kết lên phía trên để thuận tiện cho công việc này.

① Tạo mối liên hệ cho hai bảng HOCSINH và KHENTHUONG

Bạn cho thấy cả hai vùng cần liên hệ. Trong bảng HOCSINH là vùng MSHS, trong bảng KHENTHUONG là vùng KTMSHS. Thực hiện như sau:

◆ Bạn click chọn vùng MSHS là khóa chính của bảng HOCSINH, Drag và kéo sang vùng KTMSHS khi biến thành hình chữ nhật nhỏ, thả nút chuột, bạn sẽ có bảng sau:

Create Relationship

Relationship name: FK_KHENTHUONG_HOCSINH

Primary key table: HOCSINH Foreign key table: KHENTHUONG

MSHS KTMSHS

☒ Check existing data on creation

☒ Enforce relationship for replication

☒ Enforce relationship for INSERTs and UPDATES

☐ Cascade Update Related Fields

☐ Cascade Delete Related Records

OK **Cancel** **Help**

H297

Bạn thấy khung trên cùng (Relationship name) có ghi tên của quan hệ mà SQL Server 2000 sẽ quản lý (tên này không được lặp lại trong CSDL), chúng ta có thể thay đổi tên khác

Phía dưới ghi chú MSHS của bảng HOCSINH là khóa chính (Primary key table) và KTMSHS của bảng KHENTHUONG là khóa ngoại (Foreign key table)

◆ Mục **Check existing data on creation**: Kiểm tra dữ liệu trong hai bảng có đúng với ràng buộc này không, nếu không đúng thì quan hệ này sẽ không được thiết lập khi bạn chọn và thông báo sẽ xuất hiện

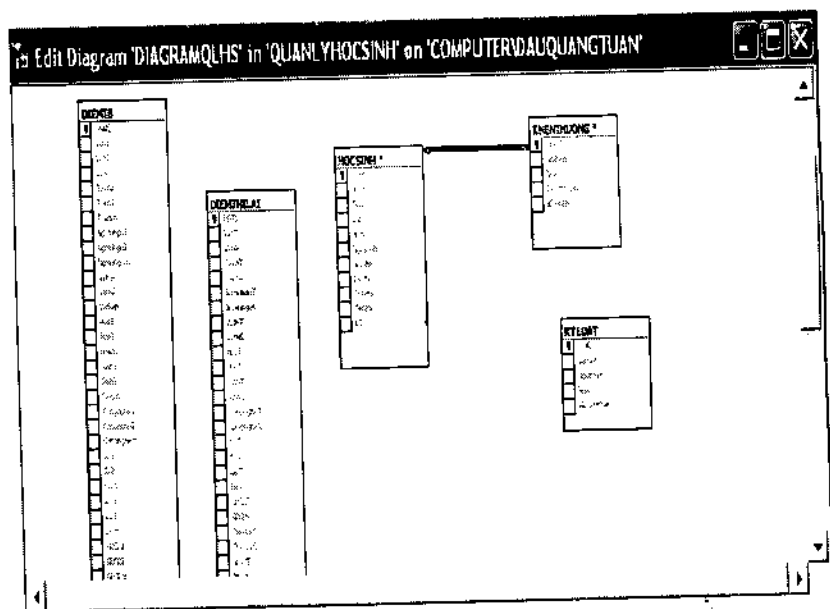
◆ Mục **Enforce relationship for replication**: Ràng buộc trọn vẹn với bảng dữ liệu được replication trong CSDL khác.

◆ Mục **Enforce relationship for INSERTs and UPDATES**: Ràng buộc trọn vẹn dữ liệu với bảng dữ liệu mỗi khi phát biểu Insert hay Update xảy ra.

◆ Mục **Cascade Update Related Fields**: Sẽ cập nhật giá trị trong cột khóa ngoại (foreign key) trong bảng KHENTHUONG mỗi khi giá trị trong cột khóa chính (primary key) của bảng HOCSINH thay đổi.

◆ Mục **Cascade delete related fields**: Xóa tất cả các mẫu tin trong bảng quan hệ foreign key mỗi khi mẫu tin trong bảng có quan hệ primary key bị xóa.

Trong bài này, giả sử chúng ta không chọn hai mục này. Click nút OK, bạn thấy có mối quan hệ giữa hai bảng



H298

② Tạo mối liên hệ cho hai bảng HOCSINH và KYLUAT

Bạn cho thấy cả hai vùng cần liên hệ. Trong bảng HOCSINH là vùng MSHS, trong bảng KYLUAT là vùng KyluatMSHS. Thực hiện như sau:

◆ Bạn click chọn vùng MSHS là khóa chính của bảng HOCSINH, Drag và kéo sang vùng KyluatMSHS khi biến thành hình chữ nhật nhỏ, thả nút chuột, bạn sẽ có bảng sau:

Create Relationship

Relationship name: FK_KYLUAT_HOCSINH

Primary key table: HOCSINH Foreign key table: KYLUAT

MSHS Ky luatMSHS

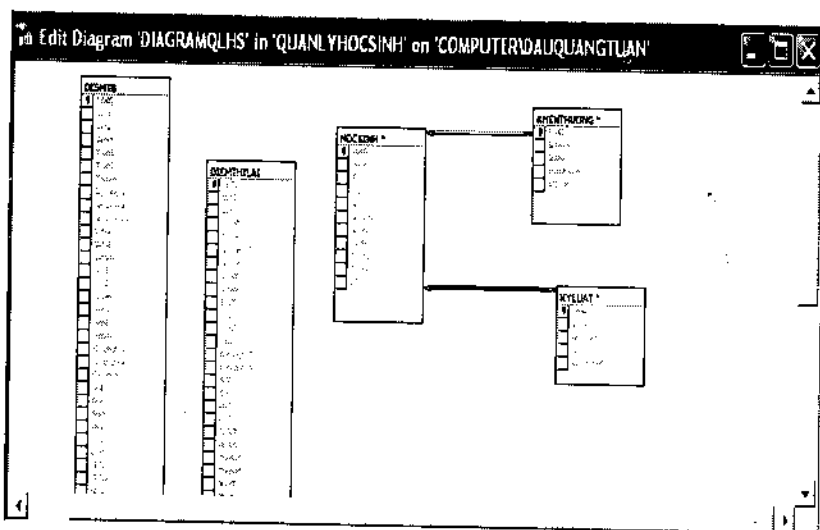
☒ Check existing data on creation
☒ Enforce relationship for replication
☒ Enforce relationship for INSERTs and UPDATES
☐ Cascade Update Related Fields
☐ Cascade Delete Related Records

OK Cancel Help

H299

◆ Các thành phần trong hộp thoại này cũng có ý nghĩa như hộp thoại trên.

◆ Click nút OK, chúng ta có mối quan hệ giữa hai bảng HOCSINH và KYLUAT



H300

③ Tạo mối liên hệ cho hai bảng HOCSINH và DIEMTB

Bạn cho thấy cả hai vùng cần liên hệ. Trong bảng HOCSINH là vùng MSHS, trong bảng DIEMTB là vùng KyluatMSHS. Thực hiện như sau:

◆ Bạn click chọn vùng MSHS là khóa chính của bảng HOCSINH, Drag và kéo sang vùng DiemMSHS khi biến thành hình chữ nhật nhỏ, thả nút chuột, bạn sẽ có bảng sau:

Create Relationship

Relationship name:

Primary key table: Foreign key table:

☒ Check existing data on creation

☒ Enforce relationship for replication

☒ Enforce relationship for INSERTs and UPDATEs

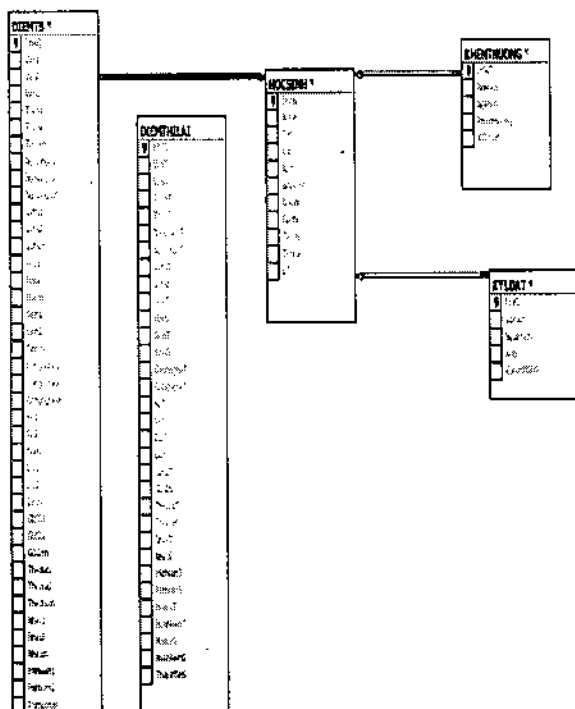
☐ Cascade Update Related Fields

☐ Cascade Delete Related Records

H301

◆ Các thành phần trong hộp thoại này cũng có ý nghĩa như hộp thoại trên.

◆ Click nút OK, chúng ta có mối quan hệ giữa hai bảng HOCSINH và DIEMTB



H302

Chúng tôi phải kéo khung Diagram lớn ra, kéo bảng DIEMTB lớn ra cho đến khi thấy được vùng DiemMSHS (điều này chúng tôi đã ghi chú ở trên, nên khi thiết kế, các bạn có thể tạo vùng DiemMSHS ở phía trên để dễ tạo mối quan hệ). Mặc dầu nối đúng vùng nhưng khi click nút OK, bạn thấy đường nối có thể nằm ở vị trí khác, bạn kiểm tra khi hộp thoại kết nối hiện ra, kiểm tra tên các vùng kết nối có đúng không?

④ Tạo mối liên hệ cho hai bảng HOCSINH và DIEMTHILAI

Bạn cho thấy cả hai vùng cần liên hệ. Trong bảng HOCSINH là vùng MSHS, trong bảng DIEMTHILAI là vùng ThilaiMSHS. Thực hiện như sau:

Bạn click chọn vùng MSHS là khóa chính của bảng HOCSINH, Drag và kéo sang vùng ThilaiMSHS khi biến thành hình chữ nhật nhỏ, thả nút chuột, bạn sẽ có bảng sau:

Create Relationship

Relationship name:

FK_DIEMTHILAI_HOCSINH

Primary key table

HOCSINH

Foreign key table

DIEMTHILAI

MSHS ThilaimSHS

- ☒ Check existing data on creation
- ☒ Enforce relationship for replication
- ☒ Enforce relationship for INSERTs and UPDATES
 - ☐ Cascade Update Related Fields
 - ☐ Cascade Delete Related Records

OK

Cancel

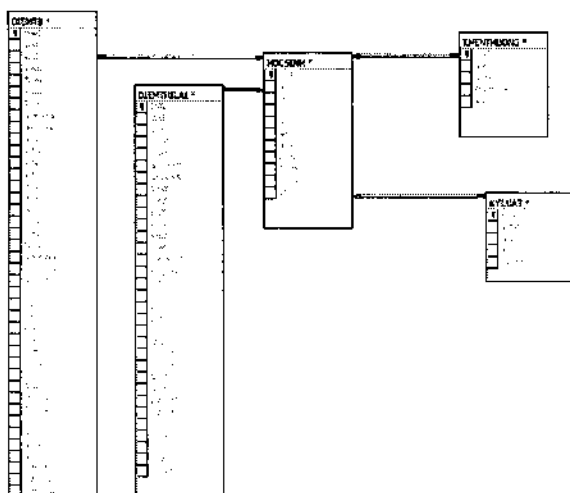
Help

H303

◆ Các thành phần trong hộp thoại này cũng có ý nghĩa như hộp thoại trên.

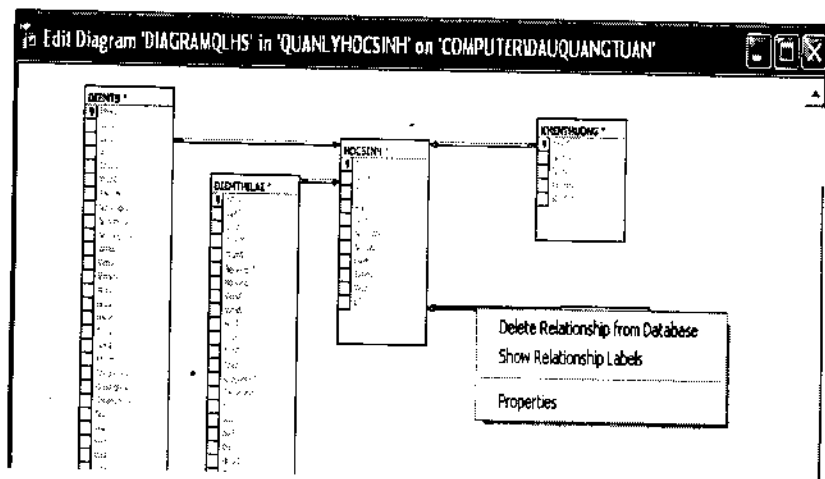
◆ Click nút OK, chúng ta có mối quan hệ giữa hai bảng HOCSINH và DIEMTHILAI

Edit Diagram 'DIAGRAMQLHS' in 'QUANLYHOCSINH' on 'COMPUTER\DAUQUANGTUAN'



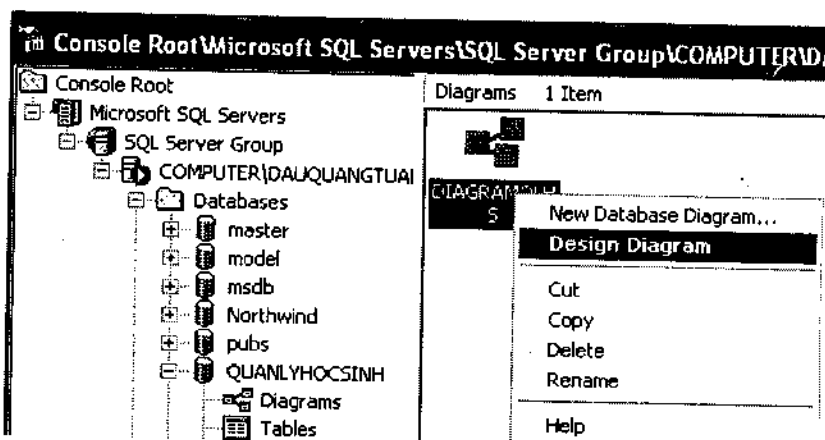
H304

◆ Nếu tạo mối liên hệ nhầm lẫn, hoặc nối sai, chúng ta có thể cắt bỏ kết nối bằng cách click nút chuột phải trên đường nối quan hệ giữa hai bảng, chọn Delete Relationship from Database



H305

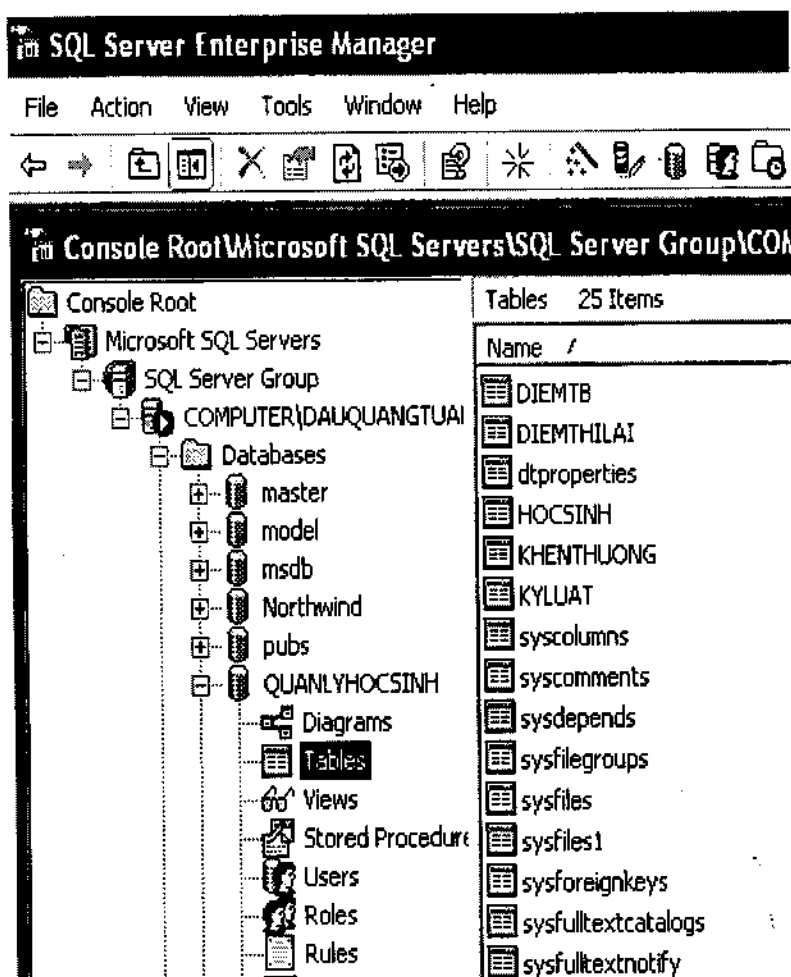
◆ Nếu cửa sổ Diagram chưa có, bạn click các dấu "+" trước các mục để hiện mục Diagram trong Tables, click chọn tên Diagram bạn thấy mục Diagram hiện ra trong khung bên phải. Click nút phải chuột tại biểu tượng Diagram, chọn Design Diagram như hình sau:



H306

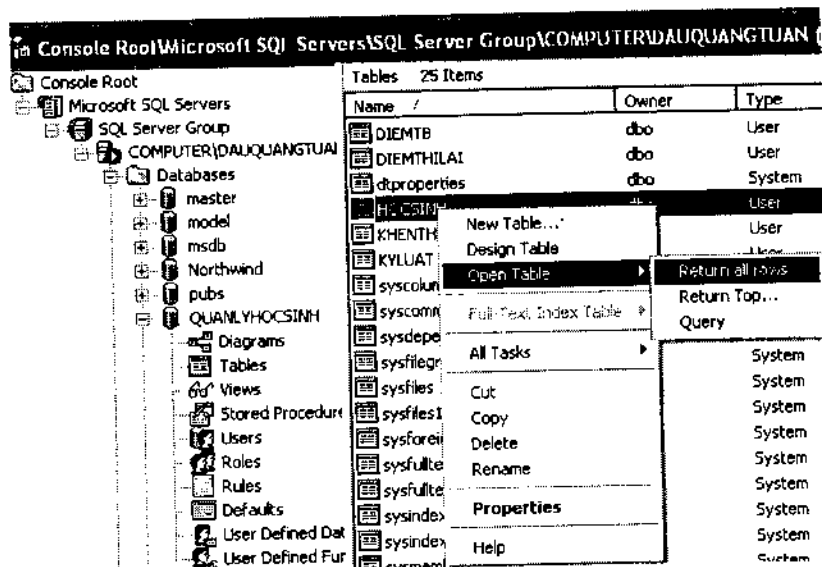
Nhập thông tin cho các bảng

◆ Bạn vào Enterprise Manager, click các dấu "+" trước các mục, sao cho Tables hiện ra, click chọn Tables, bạn thấy tên các bảng hiện ra bên khung bên phải



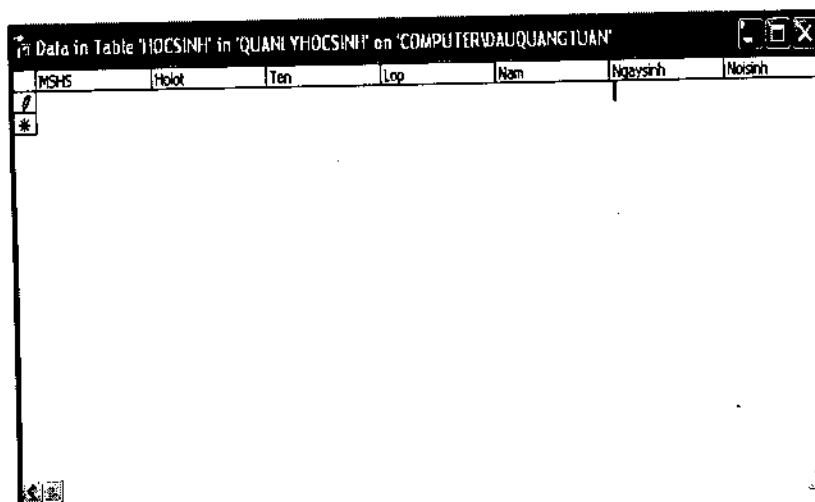
H307

◆ Bạn click nút phải trên bảng học sinh, di chuột vệt sáng đến Open table, click chọn mục Return all rows như hình sau:



H308

◆ Click xong bạn thấy bảng hiện ra cho chúng ta nhập thông tin



H309

◆ Bạn nhập thông tin cho bảng HOCSINH, tiếp theo là cho các bảng KHENTHUONG, KYLUAT, DIEMTB và DIEMTHILAI. Bạn nhớ nhập đúng cho khóa ngoại của các bảng sau, thông tin này (MSMS) phải có trong bảng HOCSINH. Giả sử bạn dựa vào quy tắc như sau:

Trước khi nhập thông tin cho các bảng để viết thử chương trình, các bạn cần lưu ý để thực hiện đúng.

◆ Đối với bảng HOCSINH

Trong bảng HOCSINH, **MSHS** là khóa và có kiểu **Text** (chiều dài tối đa là 5 kí tự), học sinh đầu tiên sẽ là **00001**, học sinh thứ hai sẽ có **MSHS** là **00002**, và tiếp tục như vậy cho đến hết, nếu bạn nhập dài quá 5 kí tự, hoặc nhập trùng với mã số đã có thì sẽ nhận thông báo là không được.

◆ Đối với bảng KHENTHUONG

Trong bảng KHENTHUONG, **MSKT** sẽ là khóa và có kiểu **Text** (chiều dài tối đa là 5 kí tự), học sinh khen thưởng đầu tiên sẽ là **KT001**, học sinh kế tiếp là **KT002** ... và cứ tiếp tục như vậy cho đến hết. Quy định nhập cũng như trên, nếu trùng hoặc nhập quá dài, SQL Server 2000 sẽ báo lỗi.

◆ Đối với bảng KYLUAT

Trong bảng KYLUAT, **MSKL** sẽ là khóa và có kiểu **Text** (chiều dài tối đa là 5 kí tự), học sinh vi phạm kỷ luật đầu tiên sẽ là **KL001**, học sinh kế tiếp là **KL002** ... và cứ tiếp tục như vậy cho đến hết. Quy định nhập cũng như trên, nếu trùng hoặc nhập quá dài, SQL Server 2000 sẽ báo lỗi.

◆ Đối với bảng DIEMTB

Trong bảng DIEMTB, **MSKQ** sẽ là khóa và có kiểu **Text** (chiều dài tối đa là 7 kí tự), học sinh đầu tiên sẽ là **KQ00001**, học sinh kế tiếp là **KQ00002** ... và cứ tiếp tục như vậy cho đến hết. Quy định nhập cũng như trên, nếu trùng hoặc nhập quá dài, SQL Server 2000 sẽ báo lỗi.

◆ Đối với bảng DIEMTHILAI

Trong bảng DIEMTHILAI, **MSTL** sẽ là khóa và có kiểu **Text** (chiều dài tối đa là 5 kí tự), học sinh đầu tiên sẽ là **TL001**, học sinh kế tiếp là **TL002** ... và cứ tiếp tục như vậy cho đến hết. Quy định nhập cũng như trên, nếu trùng hoặc nhập quá dài, SQL Server 2000 sẽ báo lỗi.

Vì trong một số bảng có nhiều vùng nên chúng tôi không thể giới thiệu đến các bạn được, chúng tôi sẽ nhập một số thông tin cho các bảng, khi thực tập bài này, các bạn đọc kỹ các kiểu của vùng, kích thước của vùng, nhất là lưu ý vùng mã số để nhập cho đúng. Chỉ cần chú ý một chút, các bạn sẽ hiểu và nhập đúng ngay.

Chúng tôi tạm ngừng giới thiệu SQL Server 2000 ở đây, từng này cũng đủ để các bạn viết chương trình.

Bài 1

Chương trình Quản lý học sinh dùng SQL Server 2000 và Visual C#

Bạn tạo một Project có tên KHENTHUONGQLHS gồm 4 Form như sau:

① Form thứ nhất có tên **Form1.cs** chứa *hai điều khiển* **DataGrid**, mỗi **DataGrid** sẽ thể hiện thông tin của từng bảng **HOCSINH** và **KHENTHUONG**. Trên khay công cụ của Form này sẽ có 5 đối tượng

◆ Các đối tượng **OleDbDataAdapter1** và **DataSetHOCSINHKT1** dùng cho bảng **HOCSINH**.

◆ Các đối tượng **OleDbDataAdapter2** và **DataSetKHENTHUONG** dùng cho bảng **KHENTHUONG**.

◆ Đối tượng **OleDbConnection1** dùng chung cho cả hai bảng.

② Form thứ hai có tên **FormKYLAT.cs** chứa *hai điều khiển* **DataGrid**, mỗi **DataGrid** sẽ thể hiện thông tin của từng bảng **HOCSINH** và **KYLAT**. Trên khay công cụ của Dialog này sẽ có 5 đối tượng

◆ Các đối tượng **OleDbDataAdapter1** và **DataSetHOCSINHKL1** dùng cho bảng **HOCSINH**.

◆ Các đối tượng **OleDbDataAdapter2** và **DataSetKYLAT** dùng cho bảng **KYLAT**.

◆ Đối tượng **OleDbConnection1** dùng chung cho cả hai bảng.

③ Form thứ ba có tên **FormDIEMTB.cs** chứa *hai điều khiển* **DataGrid**, mỗi **DataGrid** sẽ thể hiện thông tin của từng bảng **HOCSINH** và **DIEMTB**. Trên khay công cụ của Dialog này sẽ có 5 đối tượng

◆ Các đối tượng **OleDbDataAdapter1** và **DataSetHOCSINHBT1** dùng cho bảng **HOCSINH**.

◆ Các đối tượng **OleDbDataAdapter2** và **DataSetDIEMTB** dùng cho bảng **DIEMTB**.

◆ Đối tượng **OleDbConnection1** dùng chung cho cả hai bảng.

④ Form thứ tư có tên **FormDIEMTHILAI.cs** chứa *hai điều khiển* **DataGrid**, mỗi **DataGrid** sẽ thể hiện thông tin của từng bảng **HOCSINH** và **DIEMTHILAI**. Trên khay công cụ của Form này sẽ có 5 đối tượng

◆ Các đối tượng **OleDbDataAdapter1** và **DataSetHOCSINHDTL1** dùng cho bảng **HOCSINH**.

◆ Các đối tượng **OleDbDataAdapter2** và **DataSetDIEMTHILAI** dùng cho bảng **DIEMTHILAI**.

◆ Đối tượng **OleDbConnection1** dùng chung cho cả hai bảng.

* Yêu cầu

* Các Form cần thiết kế

▣ Form1.cs cần thiết kế

NAM TINH HINH CUA NHUNG HOC SINH DUOC KH...

MSXS	Holot	Ten	Lop	Nam
*				

MSKT	Ndkhen	Ngkhen	Phanhuo	KTMS
*				

XEM THÔNG TIN CẬP NHẬT

OleDbDataAdapter1

OleDbConnection1

DataSetHOCSINHDTL1

OleDbDataAdapter2

DataSetKHENTHONG1

H310

Bảng các thuộc tính của Form1.cs

TT	Đối tượng	Các thuộc tính	Giá trị
1	Dialog	Font Text WindowState	VNI-Times, size 12 Bold NAM TINH HINH CUA NHUNG HOC SINH DUOC KHEN THUONG Maximized
2	DataGrid	Name Font Anchor Datasource Datamember	DataGrid1 VNI-Times, size 12 Bold Top, Left, Right, Bottom DatasetHOCSINHKT1 HOCSINH
3	DataGrid	Name Font Anchor Datasource Datamember	DataGrid2 VNI-Times, size 12 Bold Top, Left, Right, Bottom DatasetKHENTHUONG1 KHENTHUONG
4	Button	Name Font Text	buttonXem VNI-Times, size 12 Bold XEM THÔNG TIN
5	Button	Name Font Text	buttonCapnhat VNI-Times, size 12 Bold CAPNHAT

☐ FormKYLUAT.cs cần thiết kế

MSHS	Holot	Ten	Lop	Nam
*				
MSKL	Vipham	Ngvipha	Xuly	KyluatM
*				
XEM THÔNG TIN			CẬP NHẬT	

oleDbDataAdapter1

oleDbConnection1

dataSetHOCSINHKL1

oleDbDataAdapter2

dataSetKYLUA1

H311

Bảng các thuộc tính của FormKYLUA1.cs

TT	Đối tượng	Các thuộc tính	Giá trị
1	Dialog	Font Text WindowState	VNI-Times, size 12 Bold NAM TINH HINH NHUNG HOC SINH VI PHAM KY LUAT Maximized
2	DataGrid	Name Font Anchor Datasource Datamember	DataGrid1 VNI-Times, size 12 Bold Top, Left, Right, Bottom DataSetHOCSINHKL1 HOCSINH
3	DataGrid	Name Font Anchor Datasource Datamember	DataGrid2 VNI-Times, size 12 Bold Top, Left, Right, Bottom DataSetKYLUA1 KYLUA1
4	Button	Name	buttonXem VNI-Times, size 12 Bold

		Font Text	XEM THÔNG TIN
5	Button	Name Font Text	buttonCapnhat VNI-Times, size 12 Bold CAPNHAT

▣ FormKETQUA.cs cần thiết kế

H312

Bảng các thuộc tính của FormDIEMTB.cs

TT	Đối tượng	Các thuộc tính	Giá trị
1	Dialog	Font Text WindowState	VNI-Times, size 12 Bold NAM TINH HINH KET QUA HOC TAP CUA HOC SINH Maximized
2	DataGrid	Name Font	DataGrid1 VNI-Times, size 12 Bold

3	DataGrid	Name Font Anchor Datasource Datamember	DataGrid2 VNI-Times, size 12 Bold Top, Left, Right, Bottom DatasetDIEMTB1 DIEMTB
4	Button	Name Font Text	buttonXem VNI-Times, size 12 Bold XEM THÔNG TIN
5	Button	Name Font Text	buttonCapnhat VNI-Times, size 12 Bold CAPNHAT

▣ FormDIEMTHILAI.cs cần thiết kế

NAM TINH HINH KET QUẢ THI LẠI CỦA HỌC SINH

MSHS	Holot	Ten	Lop	Nam										
* 														
<table border="1"> <thead> <tr> <th>MSTL</th> <th>VanT</th> <th>VanS</th> <th>ToanT</th> <th>ToanS</th> </tr> </thead> <tbody> <tr> <td colspan="5">* </td> </tr> </tbody> </table>					MSTL	VanT	VanS	ToanT	ToanS	* 				
MSTL	VanT	VanS	ToanT	ToanS										
* 														
XEM THÔNG TIN			CẬP NHẬT											

oleDbDataAdapter1

oleDbConnection1

dataSetHOC SINHDTL1

oleDbDataAdapter2

dataSetDIEMTHILAI1

H313

Bảng các thuộc tính của FormDIEMTHILAI.cs

TT	Đối tượng	Các thuộc tính	Giá trị
1	Dialog	Font Text WindowState	VNI-Times, size 12 Bold NAM TINH HINH KET QUA THI LAI CUA HOC SINH Maximized
2	DataGrid	Name Font Anchor Datasource Datamember	DataGrid1 VNI-Times, size 12 Bold Top, Left, Right, Bottom DatasetHOCSINHDTL1 HOCSINH
3	DataGrid	Name Font Anchor Datasource Datamember	DataGrid2 VNI-Times, size 12 Bold Top, Left, Right, Bottom DatasetDIEMTHILAI1 DIEMTHILAI
4	Button	Name Font Text	buttonXem VNI-Times, size 12 Bold XEM THÔNG TIN
5	Button	Name Font Text	buttonCapnhat VNI-Times, size 12 Bold CAPNHAT

① Thiết kế các Form trong Project

 **Thiết kế Form thứ nhất có tên Form1.cs**

◆ Vào Visual Studio .NET

◆ Chọn menu File, di chuyển vệt sáng đến New, chọn Project

Trong hộp thoại New Project hiện ra, chọn Visual C# Projects trong khung Project Types bên trái, chọn Windows Application trong khung Templates bên phải, nhập tên Project KHENTHUONGQLHS trong khung Name, nhập E:\ trong khung Location như hình sau:



Project Types:

- Visual Basic Projects
- Visual C# Projects
- Visual J# Projects
- Visual C++ Projects
- Setup and Deployment Projects
- Other Projects
- Visual Studio Solutions

Templates:

- Windows Application
- Class Library
- Windows Control Library
- Smart Device Application
- ASP.NET Web Application
- ASP.NET Web Service

A project for creating an application with a Windows user interface

Name:

KHENTHUONGQLHS

Location:

E:\

Browse...

☐ Add to Solution☒ Close Solution

Project will be created at E:\KHENTHUONGQLHS.

▼ Morg

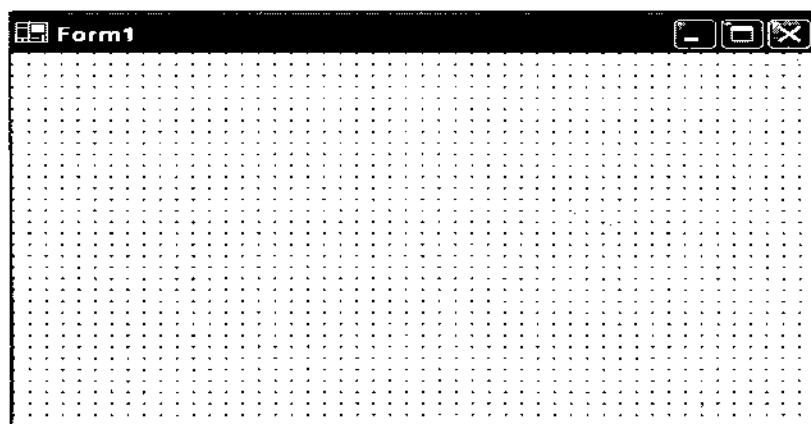
OK

Cancel

Help

H314

◆ Click nút OK, bạn sẽ thấy Form thiết kế hiện ra như hình sau:

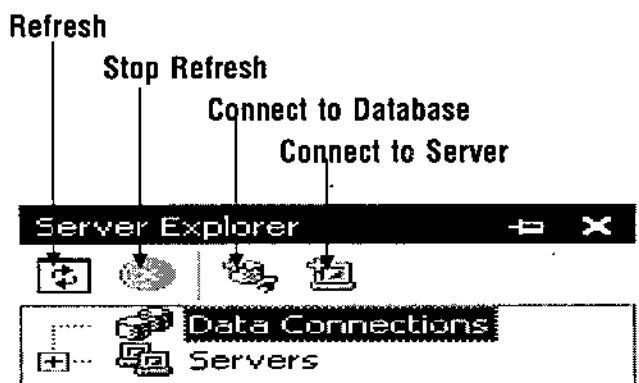


H315

◆ Click nút phải chuột trên Form, xác lập các thuộc tính Font và Text như trong bảng các thuộc tính của Form đã cho ở trên.

☐ Form sẽ kết nối đến CSDL QUANLYHOCSINH trong server có tên COMPUTER\DAUQUANTUAN của SQL Server 2000 nên chúng ta phải kết nối Dialog này với CSDL nằm trong SQL Server 2000. Bạn thực hiện kết nối như sau:

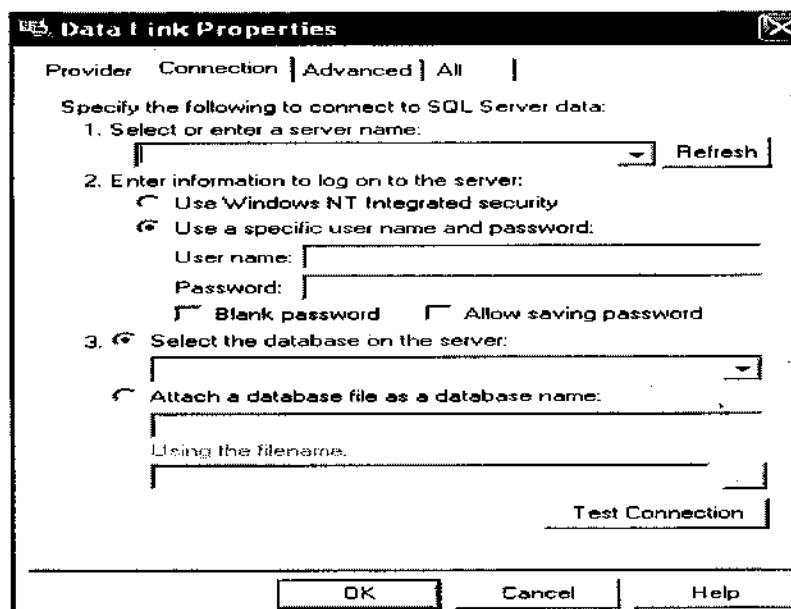
☐ Vào menu View, chọn **Server Explorer**



H316

Đây là công cụ cho phép bạn thiết lập kết nối đến CSDL.

☐ Click nút **Connect to Database** trong cửa sổ **Server Explorer**, hộp thoại kết nối hiện ra (hộp thoại **Data Link Properties**).



H317

☐ Chọn tab **Provider**, có dạng như hình sau:

Data Link Properties

Provider | Connection | Advanced | All

Select the data you want to connect to:

OLE DB Provider(s)

- Connectivity Service Provider
- MediaCatalogDB OLE DB Provider
- MediaCatalogMergedDB OLE DB Provider
- MediaCatalogWebDB OLE DB Provider
- Microsoft ISAM 1.1 OLE DB Provider
- Microsoft Jet 3.51 OLE DB Provider
- Microsoft Jet 4.0 OLE DB Provider
- Microsoft OLE DB Provider For Data Mining Services
- Microsoft OLE DB Provider for DTS Packages
- Microsoft OLE DB Provider for Indexing Service
- Microsoft OLE DB Provider for Internet Publishing
- Microsoft OLE DB Provider for ODBC Drivers
- Microsoft OLE DB Provider for Olap Services 8.0
- Microsoft OLE DB Provider for Oracle
- Microsoft OLE DB Provider for Outlook Search
- Microsoft OLE DB Provider for SQL Server**
- Microsoft OLE DB Simple Provider

Next >>

OK

Cancel

Help

H318 Tab Provider, chọn Microsoft OLE DB Provider for SQL Server

☐ Click chọn **Microsoft OLE DB Provider for SQL Server**

Ghi chú:

Microsoft OLE DB Provider for SQL Server là bộ máy cho phép truy xuất đến CSDL của **Microsoft SQL Server**. Nếu bạn dùng CSDL **Oracle** thì bạn sẽ chọn **Microsoft OLE DB Provider for Oracle**. Nếu bạn dùng **Access** thì bạn sẽ chọn **Microsoft jet 4.0 OLE DB Provider**

☐ Click nút **Next**, sẽ chuyển sang tab **Connection** và hộp thoại như hình sau hiện ra

Data Link Properties

Provider | Connection | Advanced | All

Specify the following to connect to SQL Server data:

- Select or enter a server name: Refresh
- Enter information to log on to the server:
 - ☐ Use Windows NT Integrated security
 - ☒ Use a specific user name and password:
 - User name:
 - Password:
 - ☐ Blank password ☐ Allow saving password
- ☒ Select the database on the server:
 - ☐ Attach a database file as a database name:
 - Using the filename:

Test Connection

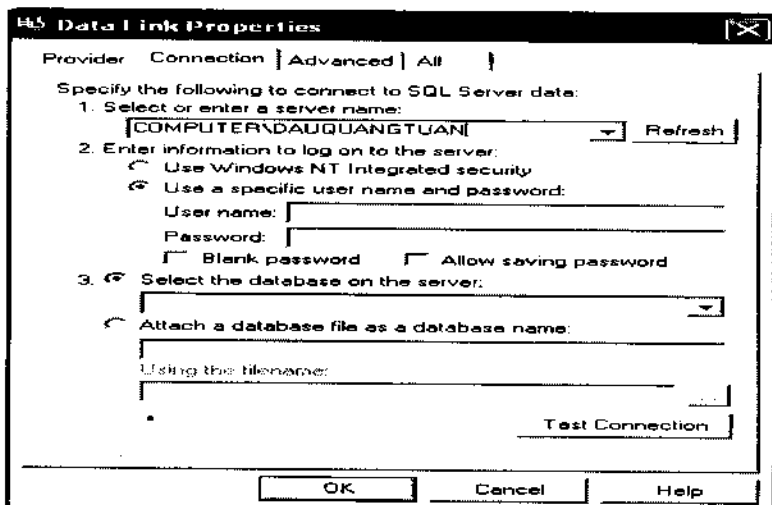
OK

Cancel

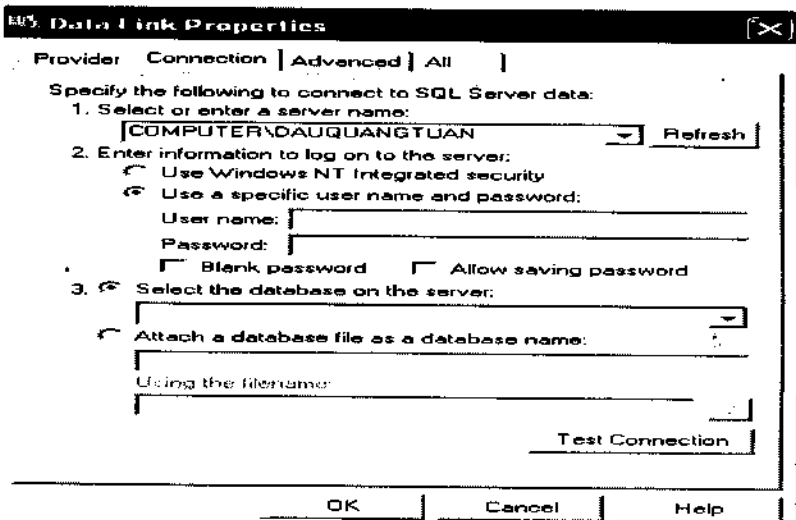
Help

H319 Hộp thoại khai báo Database cần kết nối

⊠ Nhập COMPUTER\DAUQUANGTUAN trong khung Select or enter a server name như hình sau:

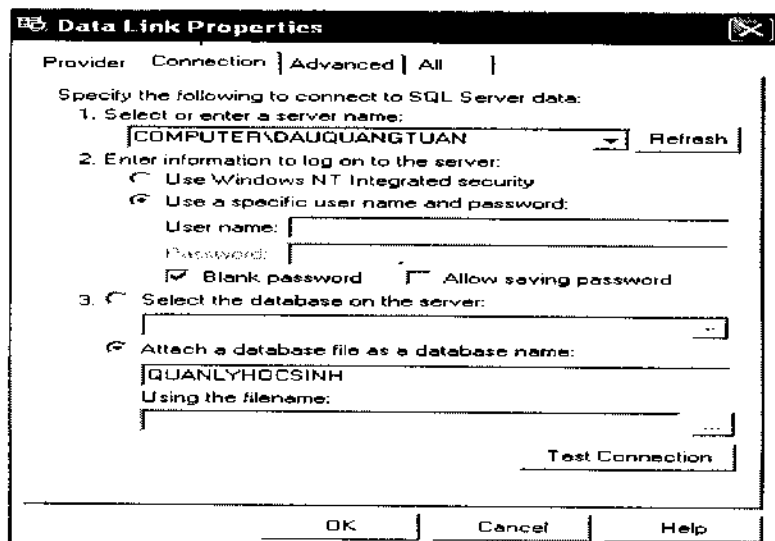


H320



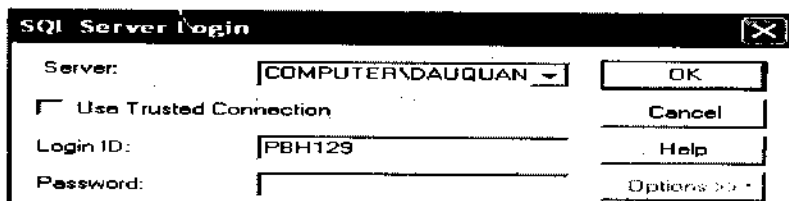
H321

◆ Click chọn mục Blank password và mục Attach a database file as a database name, nhập QUANLYHOCSINH trong khung Attach a database file as a database name như hình sau:



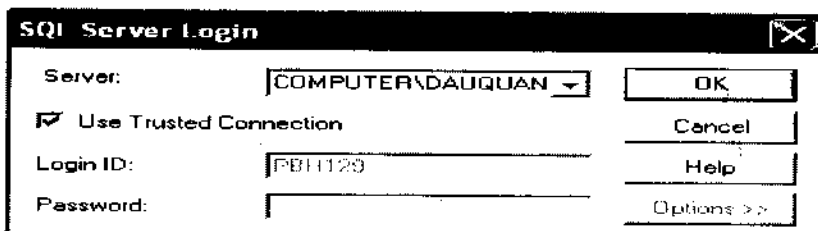
H322

- Click nút OK, màn hình sau hiện ra



H323

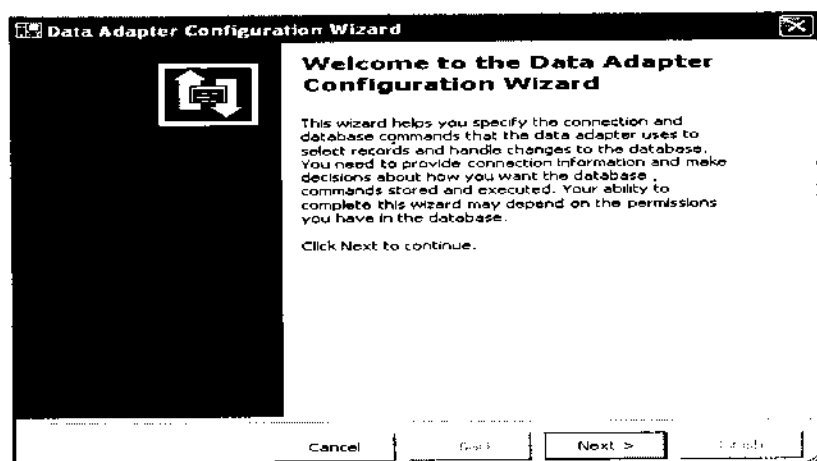
- Click chọn mục Use Trusted Connection



H324

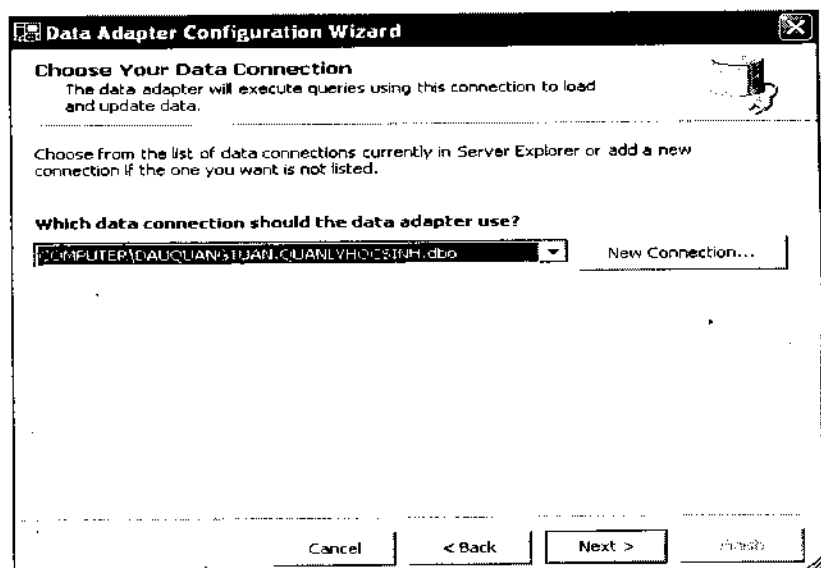
- Click nút OK để kết thúc việc kết nối. Tiếp theo là đưa đối tượng OleDbDataAdapter vào khay công cụ

◆ Vào hộp công cụ, click tab có tên Data, click chọn OleDbDataAdapter và vẽ vào Form, bạn có màn hình sau:



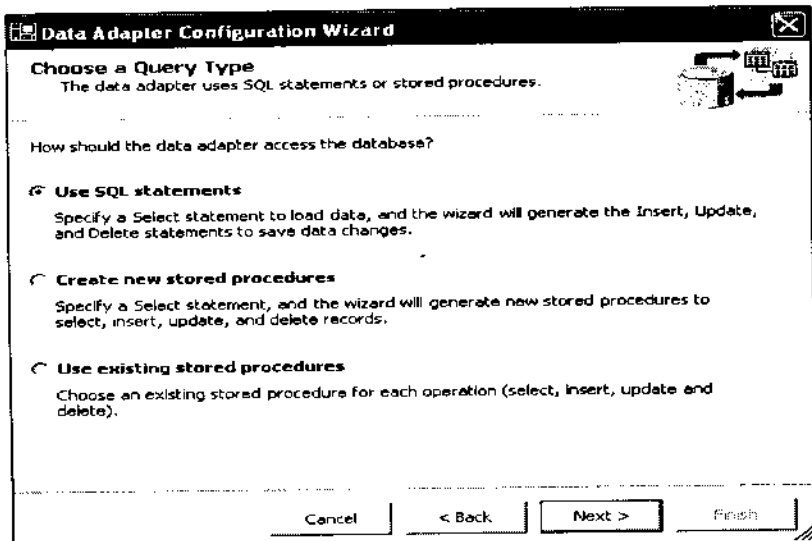
H325

◆ Click nút Next, chọn CSDL có tên QUANLYHOCSINH.dbo như màn hình sau:



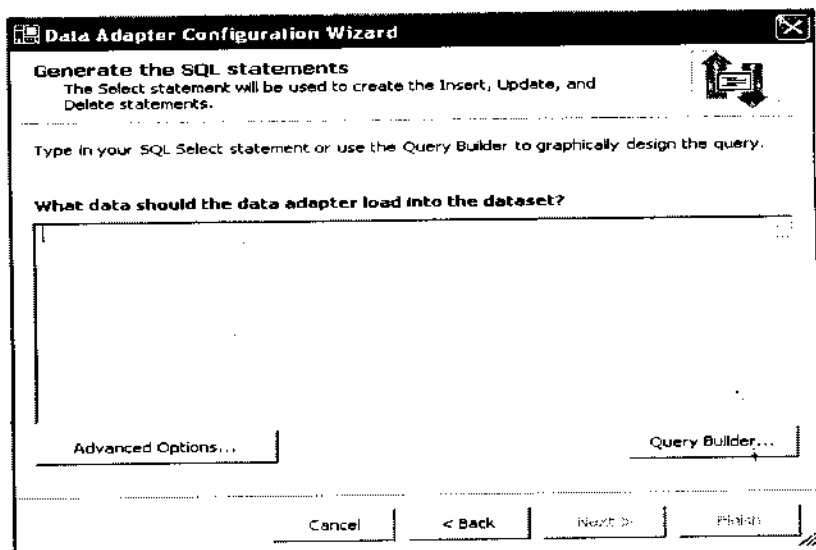
H326

◆ Click nút Next, màn hình sau hiện ra



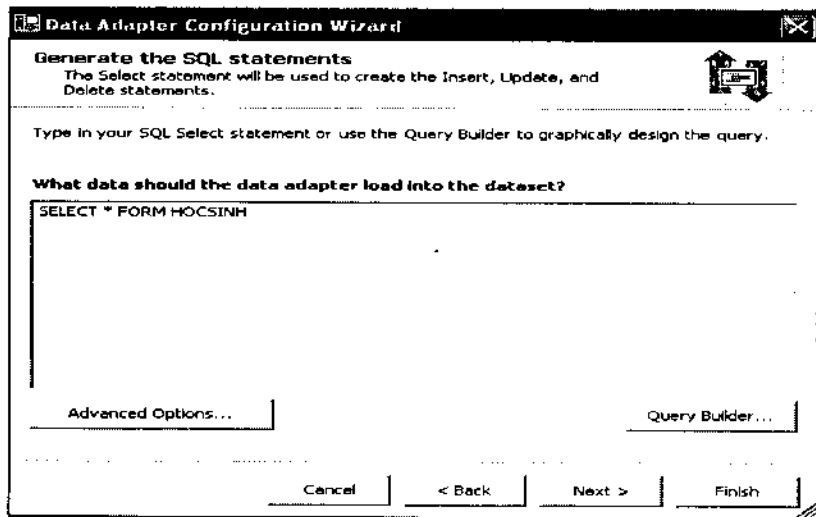
H327

◆ Click nút Next, màn hình sau hiện ra



H328

Nhập vào câu lệnh SQL như sau:
SELECT * FROM HOCSINH

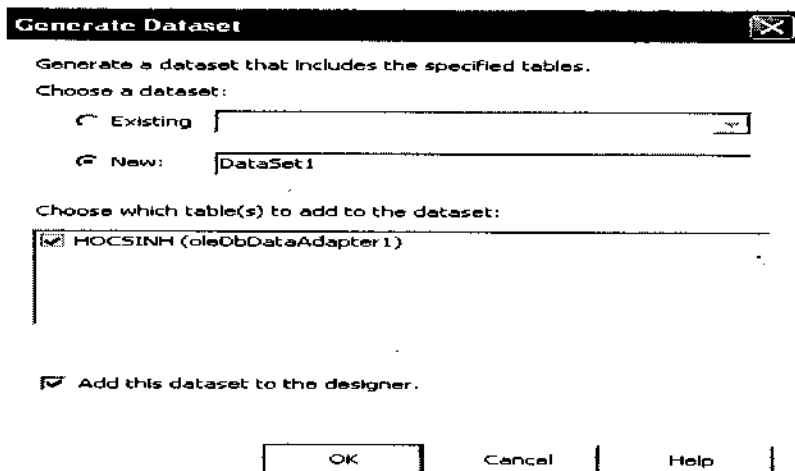


H329

◆ Click nút Finish để kết thúc việc đưa đối tượng OleDbDataAdapter vào khay công cụ. Tiếp theo là đưa đối tượng Dataset vào Form.

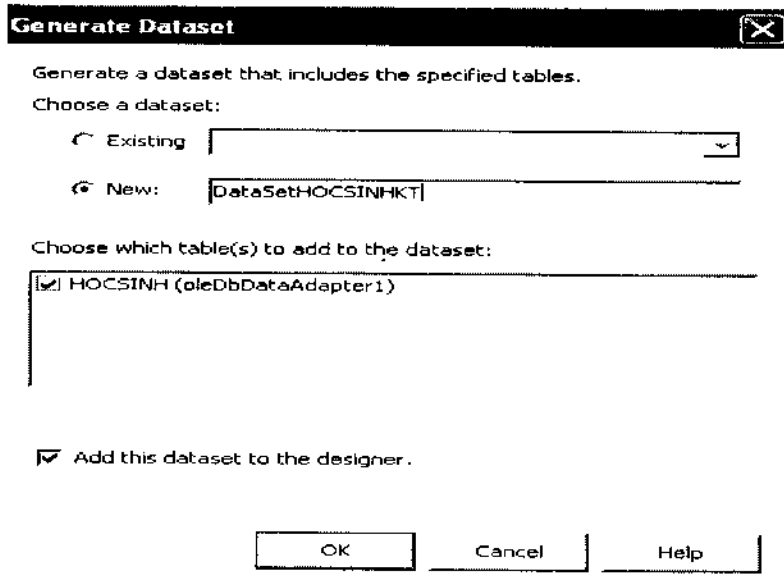
◆ Click chọn Form

◆ Vào menu Data, chọn Generate Dataset, hộp thoại sau hiện ra



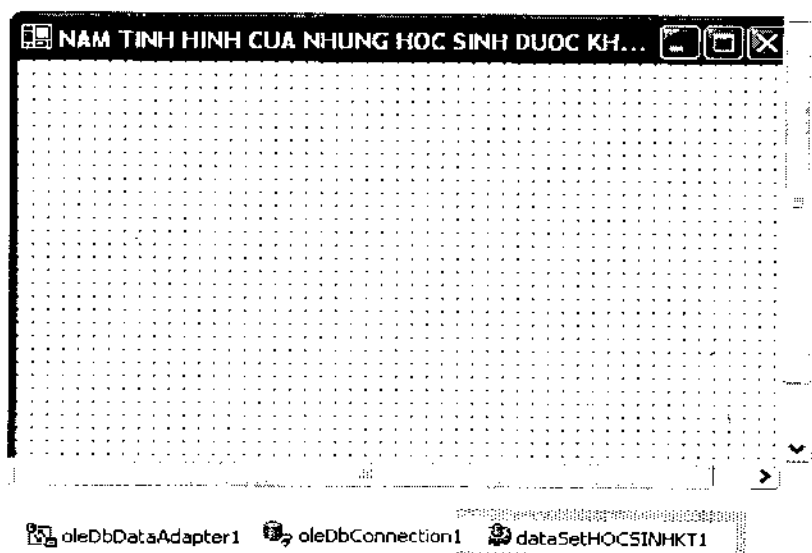
H330

Gõ tên DataSetHOCSINHKT trong khung New



H331

◆ Click nút OK, bạn thấy Form lúc này như sau:

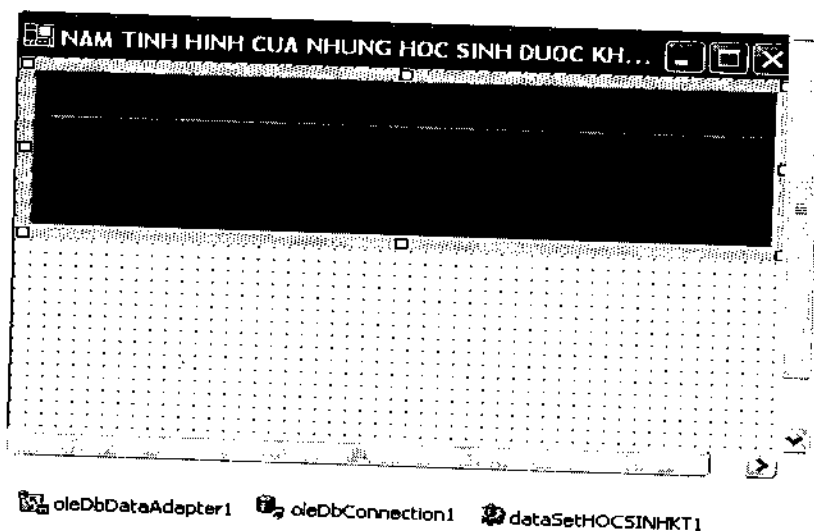


H332

Tiếp theo là đưa điều khiển DataGrid vào Form

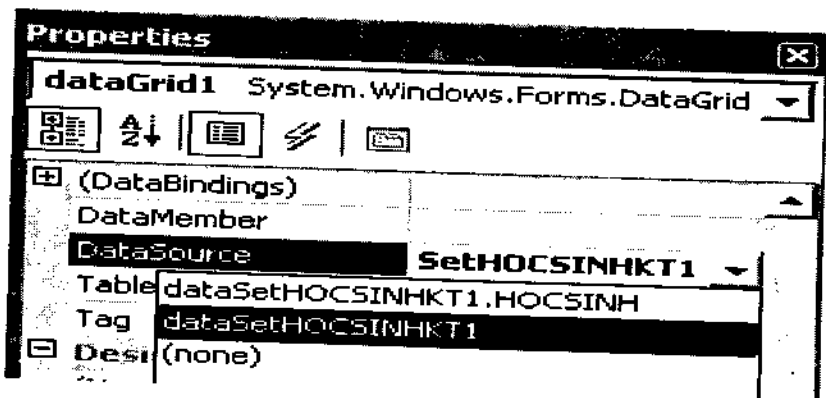
◆ Click chọn tab Windows Forms

- Click chọn điều khiển DataGrid và vẽ vào Form như hình sau:



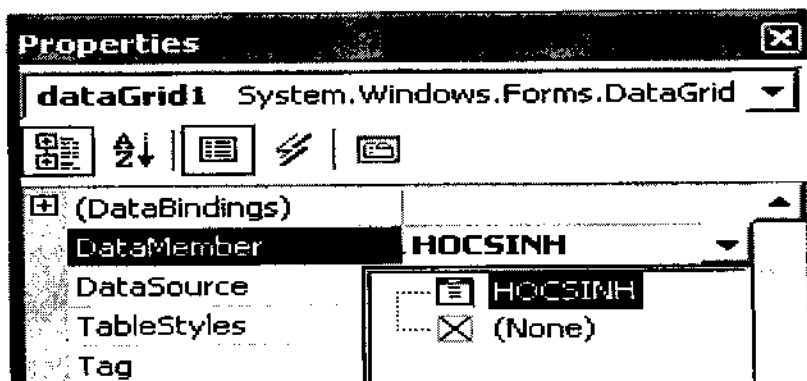
H333

- Click nút phải chuột trên DataGrid, chọn Properties để xác lập các thuộc tính như trong bảng đã cho ở trên
- Khi xác lập cho thuộc tính DataSource, click đầu mũi tên ở khung bên phải, chọn DataSetHOCSINHKT1 như sau:



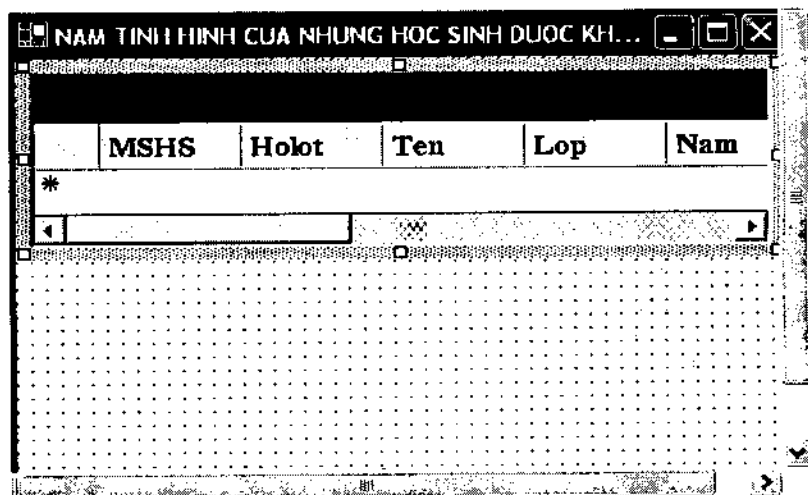
H334

- Click chọn khung bên phải của thuộc tính Datamember, sau đó click vào đầu mũi tên, chọn bảng HOCSINH trong khung như hình sau:



H335

Thực hiện xong, bạn có hình như sau:



oleDbDataAdapter1

oleDbConnection1

dataSetHOCSINHKT1

H336

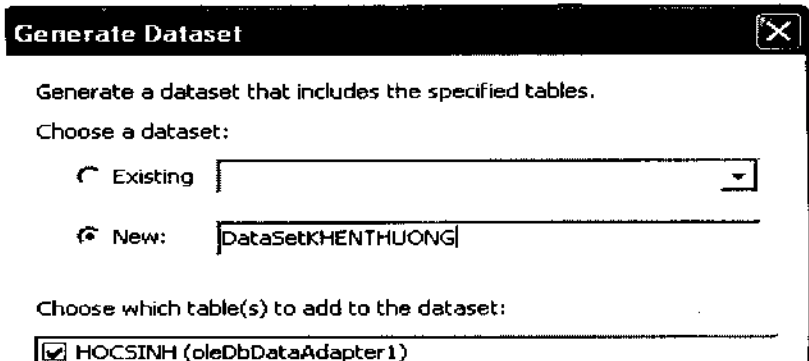
Bạn tiếp tục bước hai như những thao tác trước đây để đưa bảng KHENTHUONG vào DataGrid thứ hai.

Vì đã kết nối với CSDL QUANLYHOCSINH.dbo rồi nên không cần kết nối nữa.

◆ Click Data trong hộp công cụ, chọn OleDbDataAdapter và đưa vào Form. Trong mục này, chọn CSDL QUANLYHOCSINH.db, khi bấm nhập các câu lệnh SQL hiện ra, nhập vào câu lệnh sau:

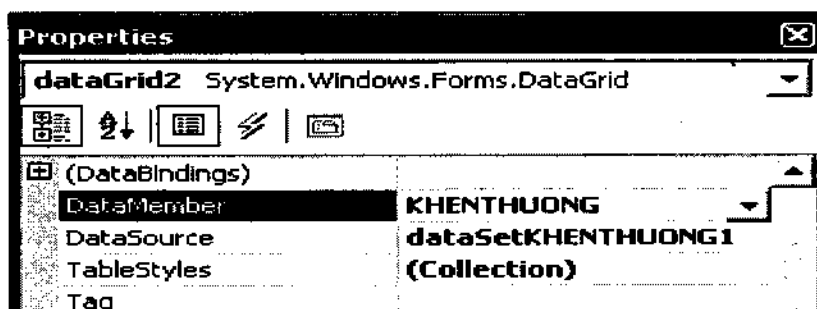
```
SELECT * FROM KHENTHUONG
```

◆ Click chọn Form, vào menu Data, chọn Generate Dataset, chọn mục New trong hộp thoại hiện ra và đặt tên là DataSetKHENTHUONG.



H337

◆ Click mục Windows Forms trong hộp công cụ, đưa điều khiển lưới DataGrid vào trong Form. Click nút phải chuột trên điều khiển lưới này, chọn Properties, xác lập các thuộc tính cho DataGrid này, ngoài các thuộc tính đơn giản, bạn cần lưu ý phải chọn đúng cho hai thuộc tính Datasource và Datamember. Thuộc tính Datasource bạn chọn DataSetKHENTHUONG1, thuộc tính Datamember bạn phải chọn bảng KHENTHUONG.



H338

Thực hiện xong, bạn sẽ có hình như sau:

oleDbDataAdapter1

oleDbConnection1

dataSetHOCSINHKT1

oleDbDataAdapter2

dataSetKHENTHUONG1

H339

♦ Vào hộp công cụ, chọn lần lượt điều khiển button đưa vào form và xác lập các thuộc tính đã cho trong bảng. Cuối cùng bạn có hình như sau:

oleDbDataAdapter1

oleDbConnection1

dataSetHOCSINHKT1

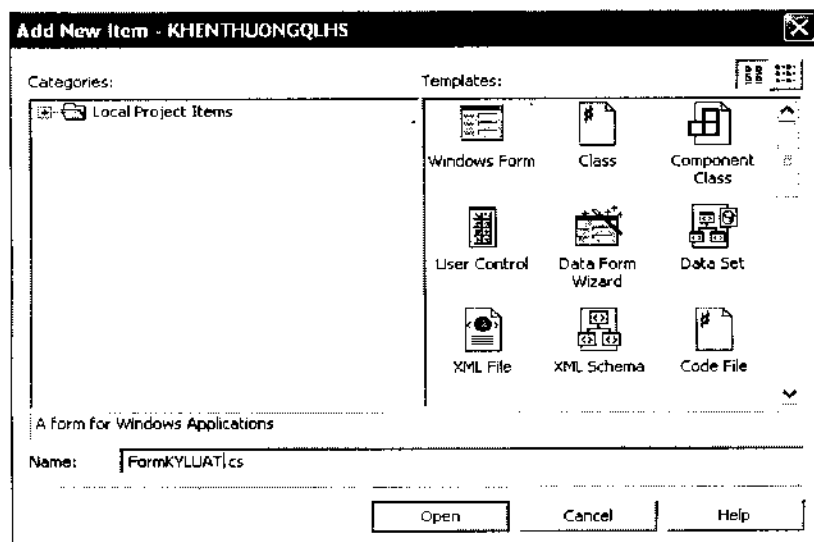
oleDbDataAdapter2

dataSetKHENTHUONG1

H340

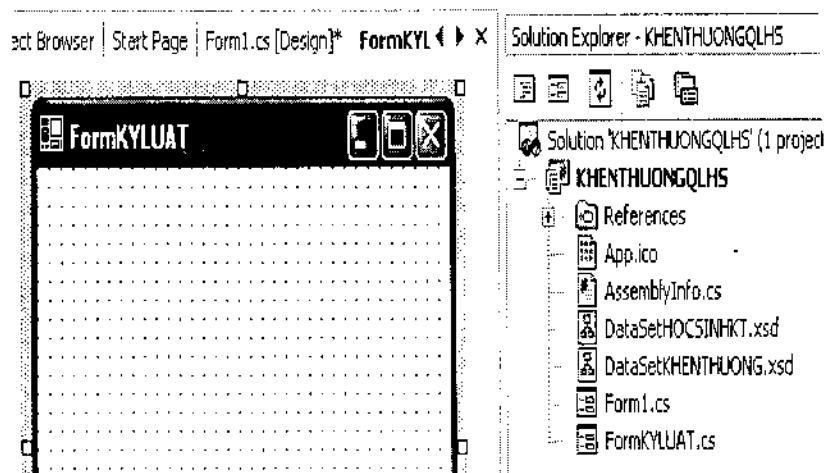
Thiết kế Form thứ hai có tên FormKYLUAT.cs

Để tạo Form thứ hai, bạn vào menu Project, chọn Add Windows Form, chọn Windows Form bên khung bên phải, đặt tên FormKYLUAT.cs như hình sau:



H341

Click nút Open, màn hình sau hiện ra



H342

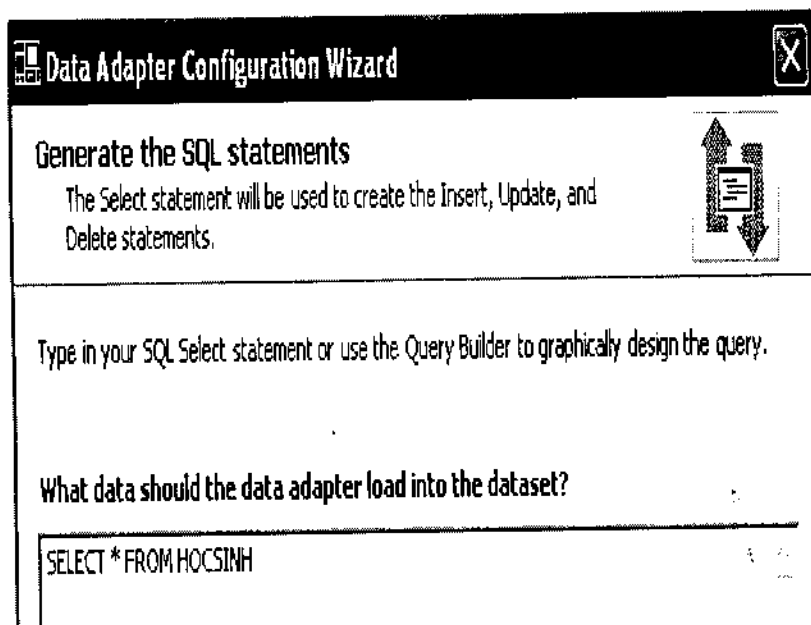
Bạn thấy Form mới hiện ra cho chúng ta thiết kế, nhìn vào cửa sổ Solution Explorer bên phải, bạn thấy có thêm FormKYLUAT.cs và Form1.cs trước đó.

Click nút phải chuột trên Form, chọn Properties, xác lập thuộc tính Font là VNI-Times, size 12, bold. Thuộc tính Text là năm "NAM TINH HINH NHUNG HOC SINH VI PHAM KY LUAT"

Tiếp theo bạn thực hiện những thao tác để đưa bảng HOCSINH và KYLUAT vào các DataGrid.

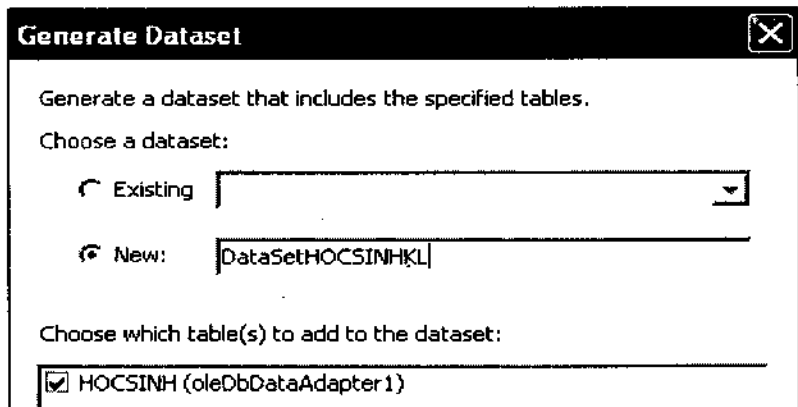
Vì đã kết nối với CSDL QUANLYHOCSINH.dbo rồi nên không cần kết nối nữa.

◆ Click Data trong hộp công cụ, chọn OleDbDataAdapter và đưa vào Form. Trong mục này, chọn CSDL QUANLYHOCSINH.dbo, khi bảng nhập các câu lệnh SQL hiện ra, nhập vào câu lệnh sau:
`SELECT * FROM HOCSINH`



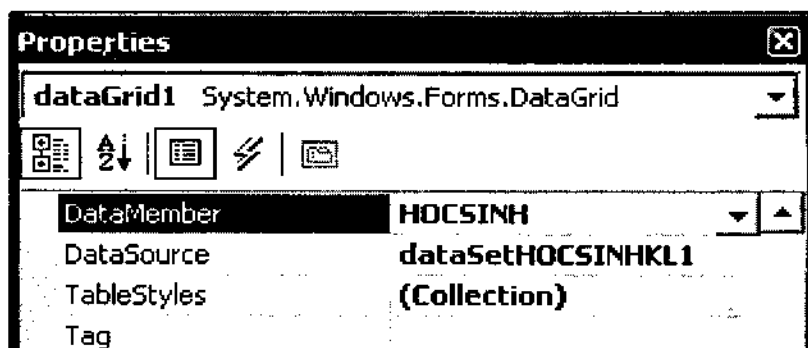
H343

◆ Click chọn Form, vào menu Data, chọn Generate Dataset, chọn mục New trong hộp thoại hiện ra và đặt tên là DatasetHOCSINHKL.



H344

◆ Click mục Windows Forms trong hộp công cụ, đưa điều khiển lưới DataGrid vào trong Form. Click nút phải chuột trên điều khiển lưới này, chọn Properties, xác lập các thuộc tính cho DataGrid này, ngoài các thuộc tính đơn giản, bạn cần lưu ý phải chọn đúng cho hai thuộc tính Datasource và Datamember. Thuộc tính Datasource bạn chọn DataSetHOCSINHKL1, thuộc tính Datamember bạn phải chọn bảng HOCSINH.

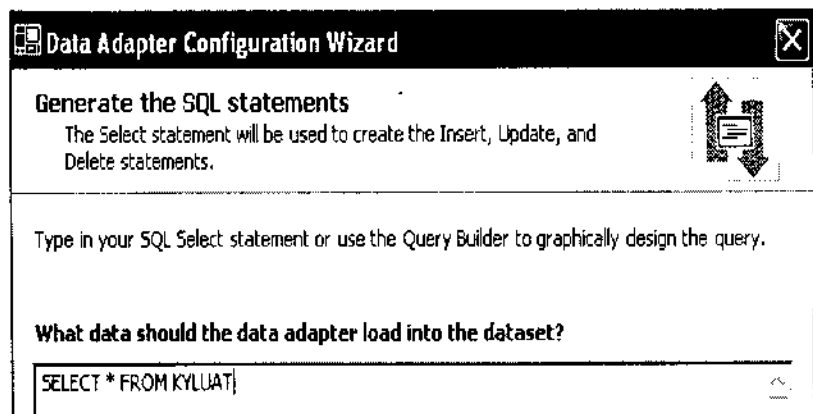


H345

Tiếp theo bạn thực hiện những thao tác để đưa bảng KYLUAT vào DataGrid.

Vì đã kết nối với CSDL QUANLYHOCSINH.dbo rồi nên không cần kết nối nữa.

◆ Click Data trong hộp công cụ, chọn OleDbDataAdapter và đưa vào Form. Trong mục này, chọn CSDL QUANLYHOCSINH.dbo, khi bảng nhập các câu lệnh SQL hiện ra, nhập vào câu lệnh sau:
SELECT * FROM KYLUAT



Data Adapter Configuration Wizard

Generate the SQL statements

The Select statement will be used to create the Insert, Update, and Delete statements.

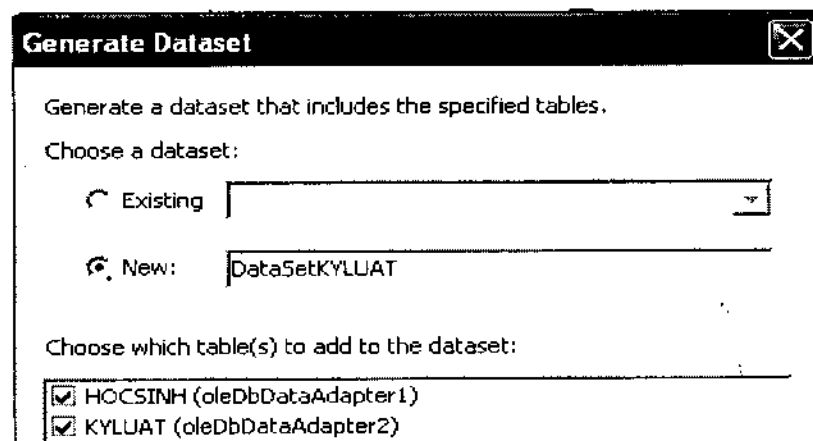
Type in your SQL Select statement or use the Query Builder to graphically design the query.

What data should the data adapter load into the dataset?

SELECT * FROM KYLUAT

H346

◆ Click chọn Form, vào menu Data, chọn Generate Dataset, chọn mục New trong hộp thoại hiện ra và đặt tên là DataSetKYLUT.



Generate Dataset

Generate a dataset that includes the specified tables.

Choose a dataset:

☐ Existing

☒ New: DataSetKYLUT

Choose which table(s) to add to the dataset:

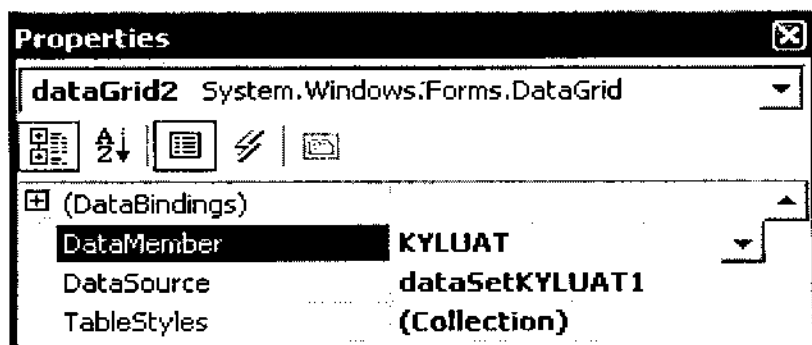
☒ HOCSINH (oleDbDataAdapter1)

☒ KYLUAT (oleDbDataAdapter2)

H347

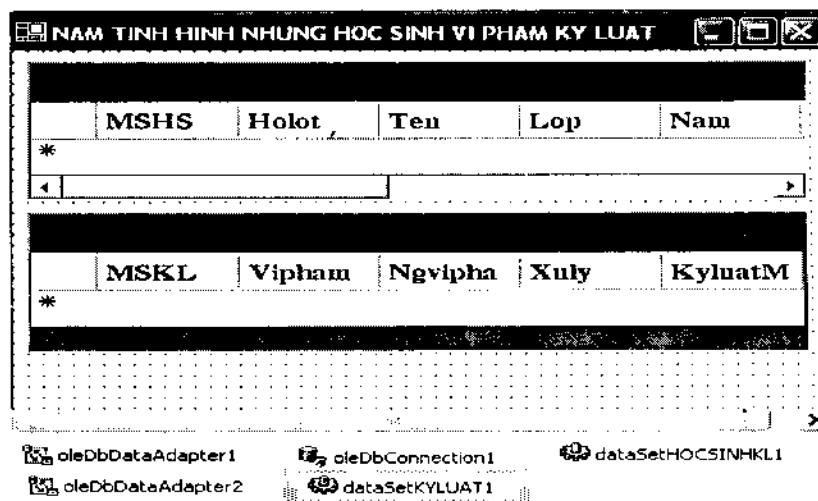
◆ Click mục Windows Forms trong hộp công cụ, đưa điều khiển lưới DataGrid vào trong Form. Click nút phải chuột trên điều khiển lưới này, chọn Properties, xác lập các thuộc tính cho DataGrid này, ngoài

các thuộc tính đơn giản, bạn cần lưu ý phải chọn đúng cho hai thuộc tính Datasource và Datamember. Thuộc tính Datasource bạn chọn DataSetKYLUA1, thuộc tính Datamember bạn phải chọn bảng KYLUAT.



H348

Lúc này bạn sẽ có Form như hình sau:



H349

Đưa hai điều khiển button vào và xác lập các thuộc tính cho chúng như bảng đã cho ở trên. Bạn có kết quả như hình sau:

NAM TÍNH HÌNH NHƯNG HỌC SINH VI PHẠM KỶ LUẬT

MSHS	Holot	Ten	Lop	Nam
*				

MSKL	Vipham	Ngvipha	Xuly	KyluatM
*				

XEM THÔNG TIN

CẬP NHẬT

oleDbDataAdapter1
oleDbDataAdapter2

oleDbConnection1
dataSetKYLUAU1

dataSetHOCSINHKL1

H350

Thiết kế Form thứ ba có tên FormDIEMTB.cs

Để tạo Form thứ ba, bạn vào menu Project, chọn Add Windows Form, chọn Windows Form bên khung bên phải, đặt tên FormDIEMTB.cs như hình sau:

Add New Item - KHENTHƯƠNGQLHS

Categories: Local Project Items

Templates:

Windows Form	Class	Component Class
User Control	Data Form Wizard	Data Set
XML File	XML Schema	Code File

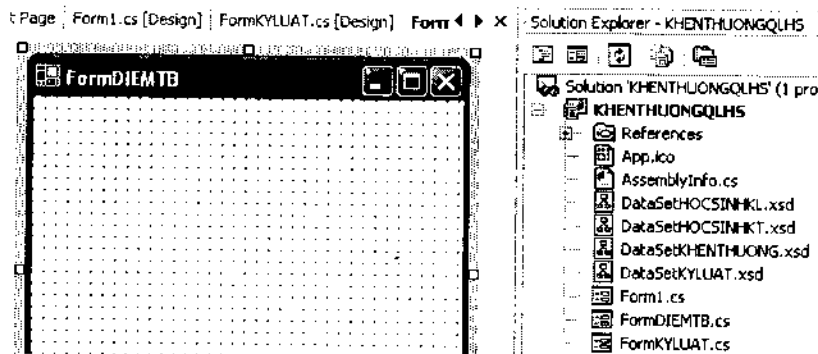
A Form for Windows Applications

Name:

Open Cancel Help

H351

Click nút Open, màn hình sau hiện ra



H352

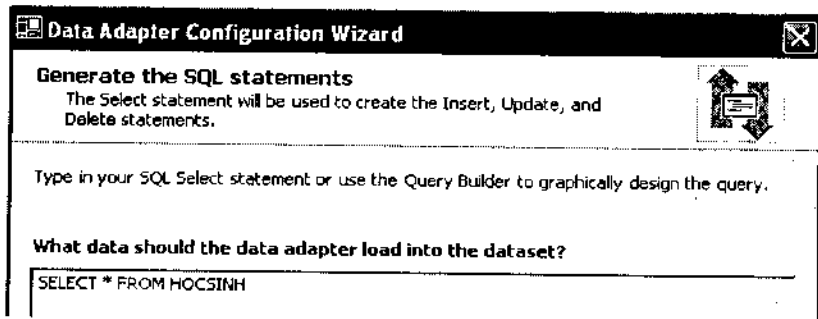
Bạn thấy Form mới hiện ra cho chúng ta thiết kế, nhìn vào cửa sổ Solution Explorer bên phải, bạn thấy có thêm FormDIEMTB.cs và Form1.cs, FormKYLUAU.cs trước đó.

◆ Click nút phải chuột trên Form, chọn Properties, xác lập thuộc tính Font là VNI-Times, size 12, bold. Thuộc tính Text là năm "NAM TINH HINH KET QUA HOC TAP CUA HOC SINH"

Tiếp theo bạn thực hiện những thao tác để đưa bảng HOCSINH và DIEMTB vào các DataGrid.

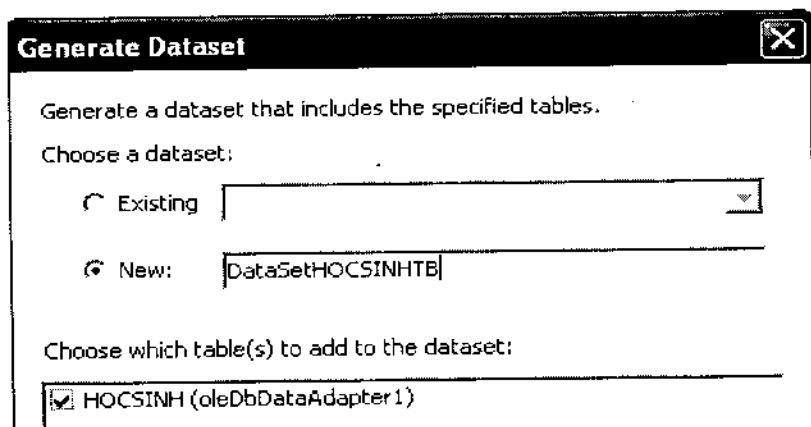
Vì đã kết nối với CSDL QUANLYHOCSINH.dbo rồi nên không cần kết nối nữa.

◆ Click Data trong hộp công cụ, chọn OleDbDataAdapter và đưa vào Form. Trong mục này, chọn CSDL QUANLYHOCSINH.dbo, khi bảng nhập các câu lệnh SQL hiện ra, nhập vào câu lệnh sau:
SELECT * FROM HOCSINH



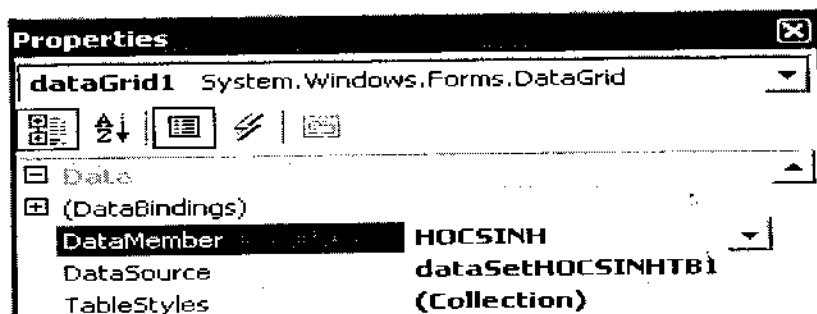
H353

◆ Click chọn Form, vào menu Data, chọn Generate Dataset, chọn mục New trong hộp thoại hiện ra và đặt tên là DataSetHOCSINHTB.



H354

◆ Click mục Windows Forms trong hộp công cụ, đưa điều khiển lưới DataGrid vào trong Form. Click nút phải chuột trên điều khiển lưới này, chọn Properties, xác lập các thuộc tính cho DataGrid này, ngoài các thuộc tính đơn giản, bạn cần lưu ý phải chọn đúng cho hai thuộc tính Datasource và Datamember. Thuộc tính Datasource bạn chọn DataSetHOCSINHTB1, thuộc tính Datamember bạn phải chọn bảng HOCSINH.



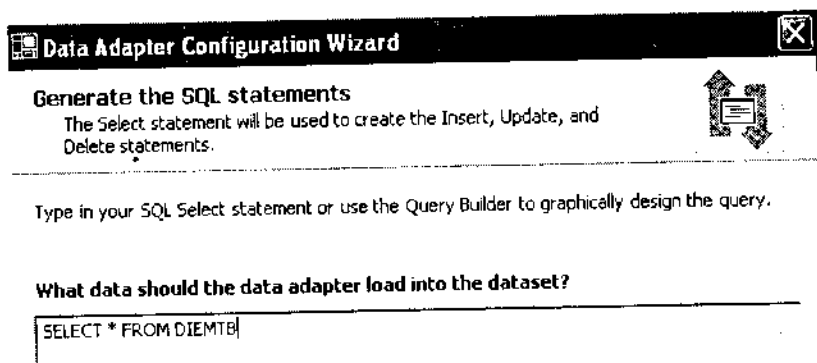
H355

Tiếp theo bạn thực hiện những thao tác để đưa bảng DIEMTB vào DataGrid.

Vì đã kết nối với CSDL QUANLYHOCSINH.dbo rồi nên không cần kết nối nữa.

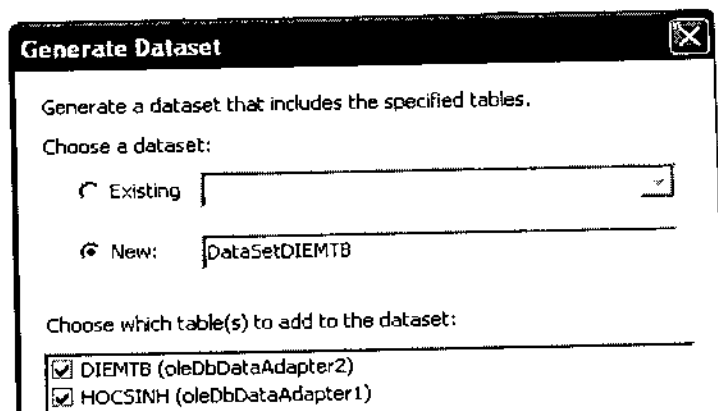
◆ Click Data trong hộp công cụ, chọn OleDbDataAdapter và đưa vào Form. Trong mục này, chọn CSDL QUANLYHOCSINH.dbo, khi bảng nhập các câu lệnh SQL hiện ra, nhập vào câu lệnh sau:

```
SELECT * FROM DIEMTB
```



H356

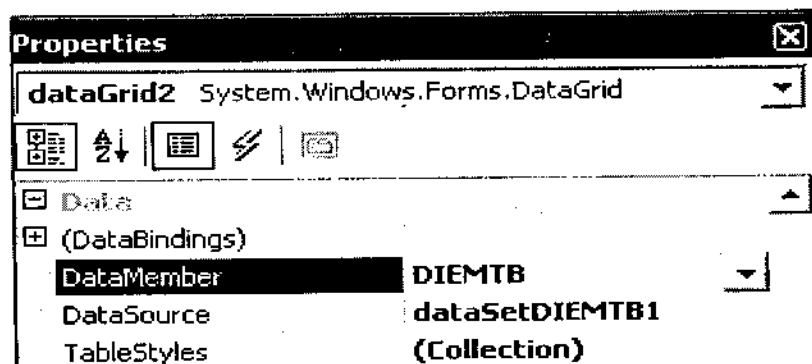
◆ Click chọn Form, vào menu Data, chọn Generate Dataset, chọn mục New trong hộp thoại hiện ra và đặt tên là DataSetDIEMTB.



H357

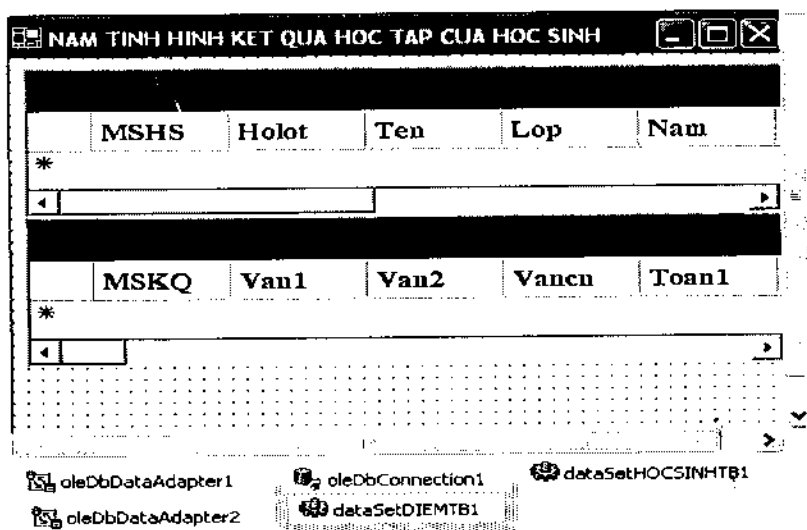
◆ Click mục Windows Forms trong hộp công cụ, đưa điều khiển lưới DataGrid vào trong Form. Click nút phải chuột trên điều khiển lưới này, chọn Properties, xác lập các thuộc tính cho DataGrid này, ngoài các thuộc tính đơn giản, bạn cần lưu ý phải chọn đúng cho hai thuộc

tính Datasource và Datamember. Thuộc tính Datasource bạn chọn DataSetDIEMTB1, thuộc tính Datamember bạn phải chọn bảng DIEMTB.



H358

Lúc này bạn sẽ có Form như hình sau:



H359

Đưa hai điều khiển button vào và xác lập các thuộc tính cho chúng như bảng đã cho ở trên. Bạn có kết quả như hình sau:

NAM TINH HINH KET QUẢ HỌC TẬP CỦA HỌC SINH

MSHS	Holot	Ten	Lop	Nam
*				

MSKQ	Van1	Van2	Vanen	Toan1
*				

XEM THÔNG TIN **CẬP NHẬT**

oleDbDataAdapter1

oleDbConnection1

dataSetHOCSINHBT1

oleDbDataAdapter2

dataSetDIEMTB1

H360

Thiết kế Form thứ tư có tên FormDIEMTHILAI.cs

Để tạo Form thứ tư, bạn vào menu Project, chọn Add Windows Form, chọn Windows Form bên khung bên phải, đặt tên FormDIEMTHILAI.cs như hình sau:

Add New Item - KHENTHUONGQLHS

Categories: Local Project Items

Templates:

Windows Form	Class	Component Class
User Control	Data Form Wizard	Data Set
XML File	XML Schema	Code File

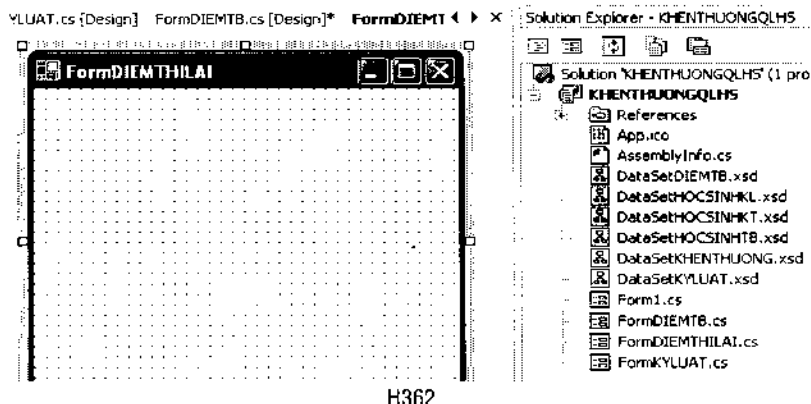
A form for Windows Applications

Name:

Open **Cancel** **Help**

H361

◆ Click nút Open, màn hình sau hiện ra



H362

Bạn thấy Form mới hiện ra cho chúng ta thiết kế, nhìn vào cửa sổ Solution Explorer bên phải, bạn thấy có thêm FormDIEMTHILAI.cs và Form1.cs, FormKYLUAU.cs, FormDIEMTB trước đó.

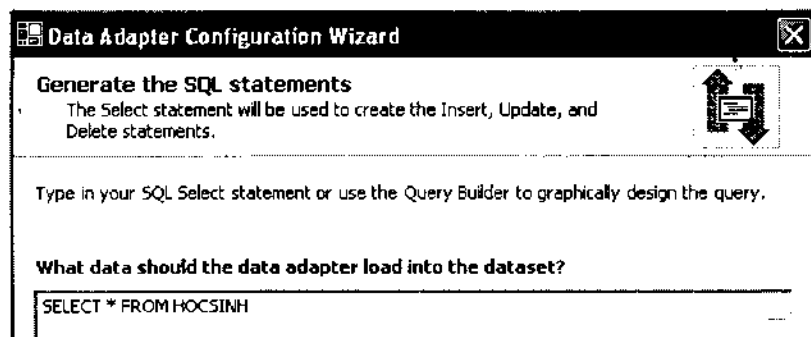
Click nút phải chuột trên Form, chọn Properties, xác lập thuộc tính Font là VNI-Times, size 12, bold. Thuộc tính Text là năm "NAM TINH HINH KET QUA THI LAI CUA HOC SINH"

Tiếp theo bạn thực hiện những thao tác để đưa bảng HOCSINH và DIEMTHILAI vào các DataGrid.

Vì đã kết nối với CSDL QUANLYHOCSINH.dbo rồi nên không cần kết nối nữa.

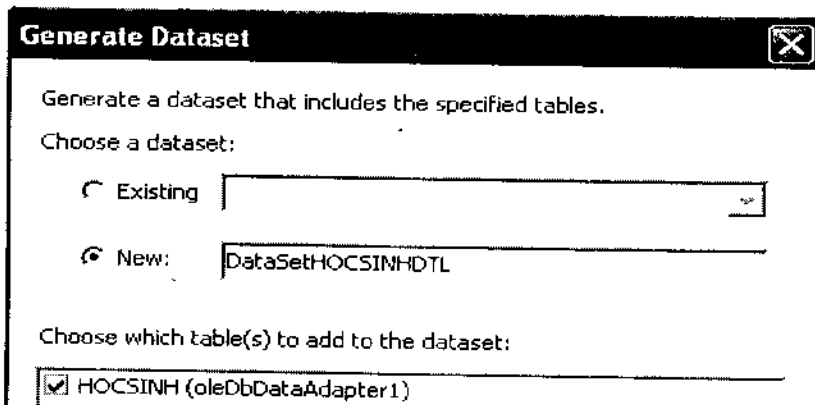
◆ Click Data trong hộp công cụ, chọn OleDbDataAdapter và đưa vào Form. Trong mục này, chọn CSDL QUANLYHOCSINH.dbo, khi bảng nhập các câu lệnh SQL hiện ra, nhập vào câu lệnh sau:

SELECT * FROM HOCSINH



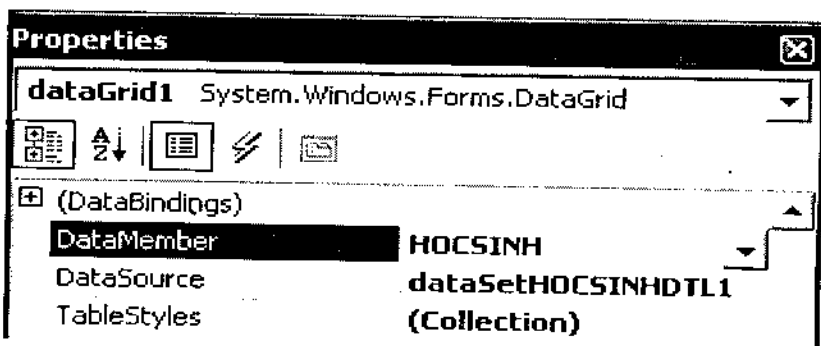
H363

◆ Click chọn Form, vào menu Data, chọn Generate Dataset, chọn mục New trong hộp thoại hiện ra và đặt tên là DataSetHOCSINHDTL.



H364

◆ Click mục Windows Forms trong hộp công cụ, đưa điều khiển lưới DataGrid vào trong Form. Click nút phải chuột trên điều khiển lưới này, chọn Properties, xác lập các thuộc tính cho DataGrid này, ngoài các thuộc tính đơn giản, bạn cần lưu ý phải chọn đúng cho hai thuộc tính Datasource và Datamember. Thuộc tính Datasource bạn chọn DataSetHOCSINHDTL1, thuộc tính Datamember bạn phải chọn bảng HOCSINH.



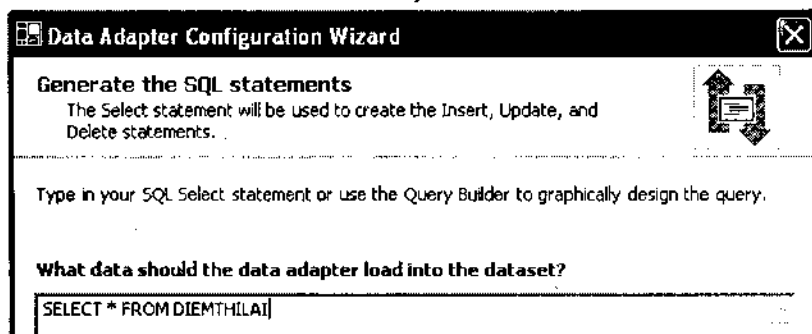
H365

Tiếp theo bạn thực hiện những thao tác để đưa bảng DIEMTHILAI vào DataGrid.

Vì đã kết nối với CSDL QUANLYHOCSINH.dbo rồi nên không cần kết nối nữa.

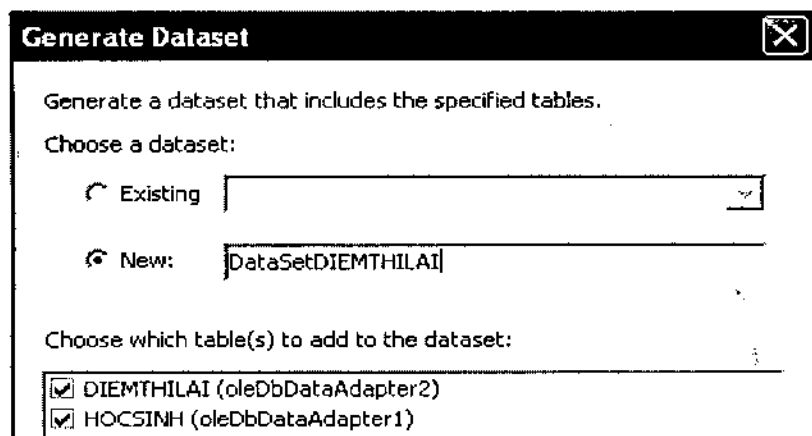
◆ Click Data trong hộp công cụ, chọn OleDbDataAdapter và đưa vào Form. Trong mục này, chọn CSDL QUANLYHOCSINH.dbo, khi bảng nhập các câu lệnh SQL hiện ra, nhập vào câu lệnh sau:

```
SELECT * FROM DIEMTHILAI
```



H366

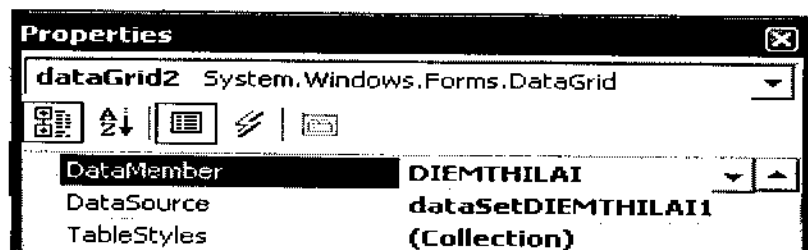
◆ Click chọn Form, vào menu Data, chọn Generate Dataset, chọn mục New trong hộp thoại hiện ra và đặt tên là DataSetDIEMTHILAI.



H367

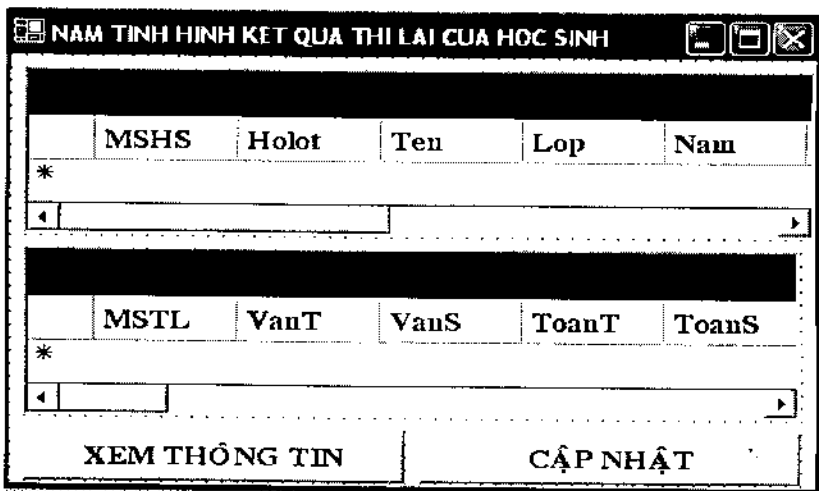
◆ Click mục Windows Forms trong hộp công cụ, đưa điều khiển lưới DataGrid vào trong Form. Click nút phải chuột trên điều khiển lưới này, chọn Properties, xác lập các thuộc tính cho DataGrid này, ngoài

các thuộc tính đơn giản, bạn cần lưu ý phải chọn đúng cho hai thuộc tính Datasource và Datamember. Thuộc tính Datasource bạn chọn DataSetDIEMTHILAI1, thuộc tính Datamember bạn phải chọn bảng DIEMTHILAI.



H368

Đưa hai điều khiển button vào và xác lập các thuộc tính cho chúng như bảng đã cho ở trên. Bạn có kết quả như hình sau:



oleDbDataAdapter1 oleDbConnection1 dataSetHOCSINHDTL1
oleDbDataAdapter2 dataSetDIEMTHILAI1

H369

Đến đây bạn đã hoàn thành việc thiết kế các Form cho chương trình, bạn còn khâu viết code để lấy được thông tin của các bảng từ cơ sở dữ liệu QUANLYHOCSINH.dbo.

② Viết code cho các Form



Viết code cho Form1.cs

Để viết code cho các Form, bạn vào cửa sổ Solution Explorer, click chọn tên Form1.cs. Click nút View Design để Form1.cs thiết kế hiện ra.

■ Điều khiển Buttonxem

Nhấp đúp vào button có tên "XEM THÔNG TIN". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttonxem_Click(object sender, System.EventArgs e)
{
    dataSetHOCSINHKT1.Clear();
    dataSetKHENTHUONG1.Clear();
    oleDbDataAdapter1.Fill(dataSetHOCSINHKT1);
    oleDbDataAdapter2.Fill(dataSetKHENTHUONG1);
}
```

Phần này chúng tôi đã giải thích ở những bài trước.

■ Điều khiển Buttoncapnhat

Nhấp đúp vào button có tên "CẬP NHẬT". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttoncapnhat_Click(object sender, System.EventArgs e)
{
    oleDbDataAdapter1.Update(dataSetHOCSINHKT1);
    oleDbDataAdapter2.Update(dataSetKHENTHUONG1);
}
```

Phần viết code này các bạn cũng đã biết từ những bài trên. :



Viết code cho FormKYLAT.cs

Click chọn tên FormKYLAT.cs trong cửa sổ Solution Explorer. Click nút View Design để FormKYLAT.cs thiết kế hiện ra.

■ Điều khiển Buttonxem

Nhấp đúp vào button có tên "XEM THÔNG TIN". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttonxem_Click(object sender, System.EventArgs e)
{
    dataSetHOCSINHKL1.Clear();
    dataSetKYLUA1.Clear();
    oleDbDataAdapter1.Fill(dataSetHOCSINHKL1);
    oleDbDataAdapter2.Fill(dataSetKYLUA1);
}
```

■ Điều khiển Buttoncapnhat

Nhấn đúp vào button có tên "CẬP NHẬT". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttoncapnhat_Click(object sender, System.EventArgs e)
{
    oleDbDataAdapter1.Update(dataSetHOCSINHKL1);
    oleDbDataAdapter2.Update(dataSetKYLUA1);
}
```



Viết code cho FormDIEMTB.cs

Click chọn tên FormDIEMTB.cs trong cửa sổ Solution Explorer. Click nút View Design để FormDIEMTB.cs thiết kế hiện ra.

■ Điều khiển Buttonxem

Nhấn đúp vào button có tên "XEM THÔNG TIN". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttonxem_Click(object sender, System.EventArgs e)
{
    dataSetHOCSINHBT1.Clear();
    dataSetDIEMTB1.Clear();
    oleDbDataAdapter1.Fill(dataSetHOCSINHBT1);
    oleDbDataAdapter2.Fill(dataSetDIEMTB1);
}
```

■ Điều khiển Buttoncapnhat

Nhấn đúp vào button có tên "CẬP NHẬT". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttoncapnhat_Click(object sender, System.EventArgs e)
```

```
{
    oleDbDataAdapter1.Update(dataSetHOCSINHDTB1);
    oleDbDataAdapter2.Update(dataSetDIEMTB1);
}
```



Viết code cho FormDIEMTHILAI.cs

Click chọn tên FormDIEMTHILAI.cs trong cửa sổ Solution Explorer. Click nút View Design để FormDIEMTHILAI.cs thiết kế hiện ra.

■ Điều khiển Buttonxem

Nhấn đúp vào button có tên "XEM THÔNG TIN". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttonxem_Click(object sender, System.EventArgs e)
{
    dataSetHOCSINHDTL1.Clear();
    dataSetDIEMTHILAI1.Clear();
    oleDbDataAdapter1.Fill(dataSetHOCSINHDTL1);
    oleDbDataAdapter2.Fill(dataSetDIEMTHILAI1);
}
```

■ Điều khiển Buttoncapnhat

Nhấn đúp vào button có tên "CẬP NHẬT". Cửa sổ viết code hiện ra với hai câu lệnh đầu và cuối của thủ tục này. Bạn viết code để hoàn thành như sau:

```
private void buttoncapnhat_Click(object sender, System.EventArgs e)
{
    oleDbDataAdapter1.Update(dataSetHOCSINHDTL1);
    oleDbDataAdapter2.Update(dataSetDIEMTHILAI1);
}
```

Nhấn phím F5 chạy thử chương trình, bạn thấy kết quả hiện ra như hình sau:

 NAM TINH HINH CUA NHUNG HOC SINH DUOC KH...   

	MSHS	Holot	Ten	Lop	Nam
▶	(null)	(null)	(null)	(null)	(null)

◀

	MSKT	Ndkhen	Ngkhen	Phanthuo	KTMS
*					

◀

XEM THÔNG TIN

CẬP NHẬT

H370

Click nút có tiêu đề "XEM THÔNG TIN", bạn có kết quả như hình sau:

 NAM TINH HINH CUA NHUNG HOC SINH DUOC KH...   

	MSHS	Holot	Ten	Lop	Nam
▶	00001	Trần Mạ	An	10A1	X

◀

	MSKT	Ndkhen	Ngkhen	Phanthuo	KTMS
▶	KT001	Học sinh	4/20/2005	Một máy	0000

◀

XEM THÔNG TIN

CẬP NHẬT

H371

Lưu ý:

Vì trong Project Form1.cs sẽ là Form được chọn mặc nhiên khi chạy chương trình, nếu muốn Form nào là mặc nhiên, bạn có thể vào **code của Form1.cs**, sửa lại một chút về tên Form được gọi trong thủ tục static void Main().

Giả sử khi chạy chương trình, bạn muốn FormDIEMTB.cs thể hiện ra màn hình, thay vì:

```
static void Main()
{
    Application.Run(new Form1());
}
```

Bạn sửa lại như sau:

```
static void Main()
{
    Application.Run(new FormDIEMTB());
}
```

Nếu sửa lại như trên, chạy lại chương trình, bạn có kết quả như hình sau:

	MSHS	Holot	Ten	Lop	Nam
▶	(null)	(null)	(null)	(null)	(null)

	MSKQ	Van1	Van2	Vancn	Toan1
*					

XEM THÔNG TIN CẬP NHẬT

H372

◆ Click nút có tiêu đề "XEM THÔNG TIN" bạn thấy kết quả như hình sau:

NAM TINH HINH KET QUA HOC TAP CUA HOC SINH

	MSHS	Holot	Ten	Lop	Nam
	00001	Trần Mạ	An	10A1	X

	MSKQ	Van1	Van2	Vanch	Toan1
	KQ001	9	9	9	7

XEM THÔNG TIN

CẬP NHẬT

H373

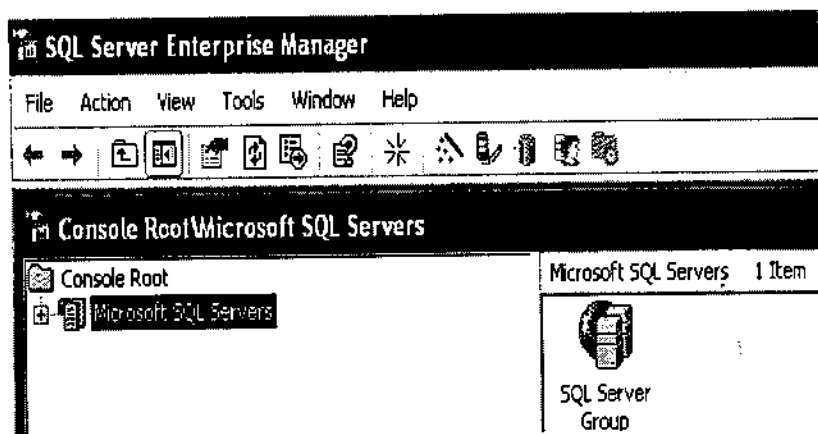
Bài 2

Chương trình quản lý học sinh Thiết kế bằng Data Form Wizard

Bài trên các bạn thực hiện thiết kế chương trình bằng chính mình, hơi lâu và chưa có hiệu quả cao, để thuận tiện và có hiệu quả bạn nên thiết kế bằng Data Form Wizard như ví dụ sau. Chúng tôi cũng dùng lại CSDL QUANLYHOCSINH.dbo ở trên.

Qua các bài trước, các bạn đã biết tạo cơ sở dữ liệu, tạo bảng, tạo các mối liên hệ giữa các bảng có liên quan với nhau, chúng tôi đã hướng dẫn bạn tạo một cơ sở dữ liệu có tên QUANLYHOCSINH trong SQL Server 2000. Bạn chú ý khi cài đặt SQL Server 2000, bạn đặt tên server là gì, các bạn cần nhớ, nếu quên, bạn có thể xem bằng cách:

☐ Click menu Start, di chuyển vệt sáng đến Programs, di chuyển vệt sáng đến Microsoft SQL Server 2000, chọn Enterprise Manager. Bạn sẽ thấy như hình sau:



H374

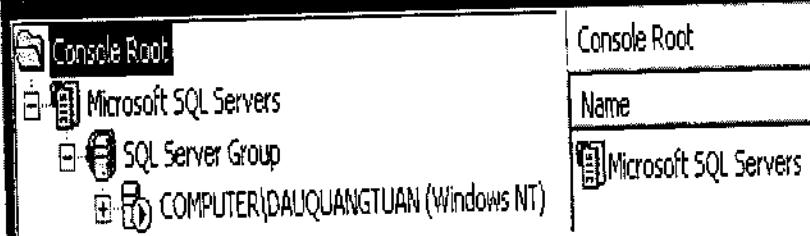
◆ Bạn click dấu "+" trước Microsoft SQL Server, sau đó click vào dấu "+" trước Microsoft SQL Group trong cây bung ra, bạn sẽ thấy tên Server

SQL Server Enterprise Manager

File Action View Window Help



Console Root



H375

Bạn thấy tên là COMPUTER\DAUQUANGTUAN, bạn đối chiếu ở bài 10 khi cài đặt, lúc nào chúng tôi đặt tên DAUQUANGTUAN. Sở dĩ chúng tôi nói điều này vì khi viết chương trình trên Visual C# .NET, trước tiên chúng ta phải kết nối với CSDL và trong bước kết nối này nó đòi hỏi chúng ta phải khai báo tên của Server.

Để viết một chương trình quản lý nào đó, trước hết các bạn phải tạo một cơ sở dữ liệu, sau đó tạo các bảng, xây dựng các mối liên hệ giữa các bảng có liên quan, nhập thông tin cho các bảng sau đó các bạn vào Visual C# .NET bắt đầu thiết kế chương trình.

Trong bài 2, chúng ta đã thiết kế cơ sở dữ liệu có tên QUANLYHOCSINH, trong CSDL này có năm bảng, và chúng ta cũng đã tạo các mối liên hệ, nếu chưa nhập thông tin, các bạn nhập thử, không cần nhiều, mỗi bảng nhập vài mẫu tin, nhớ nhập cho chính xác.

Với bài này, giả sử chúng ta phải dùng nhiều Form

Có 4 Form như sau:

① Form con thứ nhất có tên **QLHSKHENTHUONG.cs** thể hiện thông tin của hai bảng **HOCSINH** và **KHENTHUONG**.

② Form con thứ hai có tên **QLHSKYLUAT.cs** thể hiện thông tin của hai bảng **HOCSINH** và **KYLUAT**.

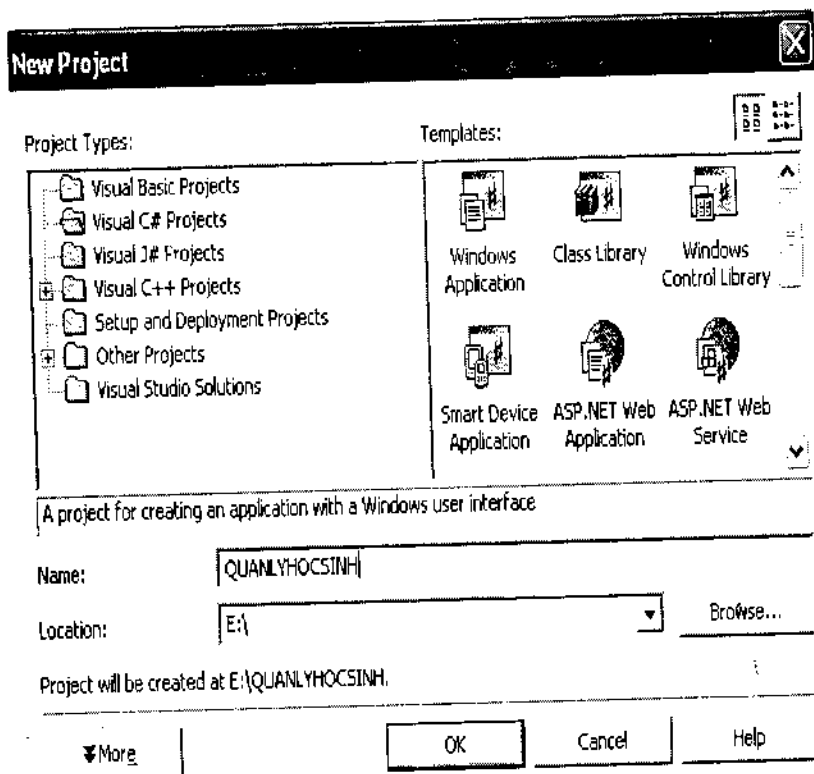
③ Form con thứ ba có tên **QLHSKETQUA.cs** thể hiện thông tin của hai bảng **HOCSINH** và **DIEMTB**.

④ Form con thứ tư có tên **QLHSDIEMTHILAI.cs** thể hiện thông tin của hai bảng **HOCSINH** và **DIEMTHILAI**.

Trong bài này, chúng ta dùng Data Form Wizard để thiết kế chương trình. Trong cả 4 bảng, chúng ta sẽ dùng cách 2 trong trang cuối cùng của Data Form Wizard.

◆ Vào visual Studio .NET

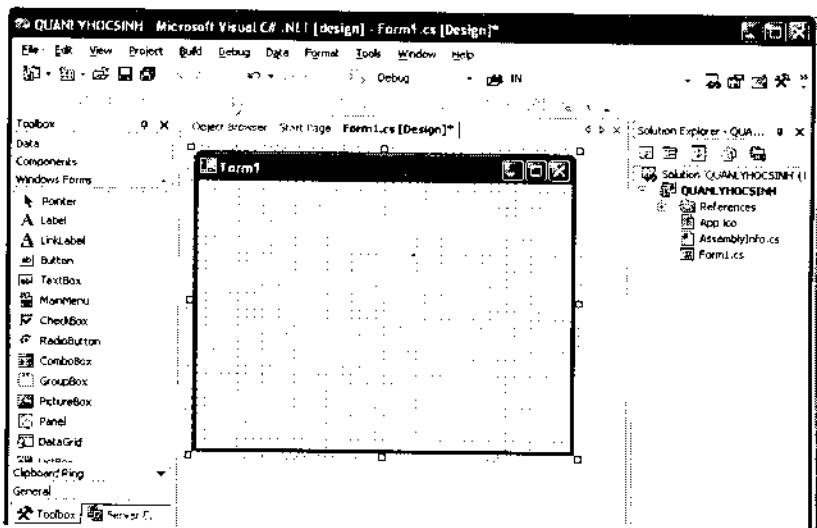
◆ Click menu File, chọn New, chọn Project để tạo một Project mới bằng Visual C# Projects và dùng Windows Application.



H376

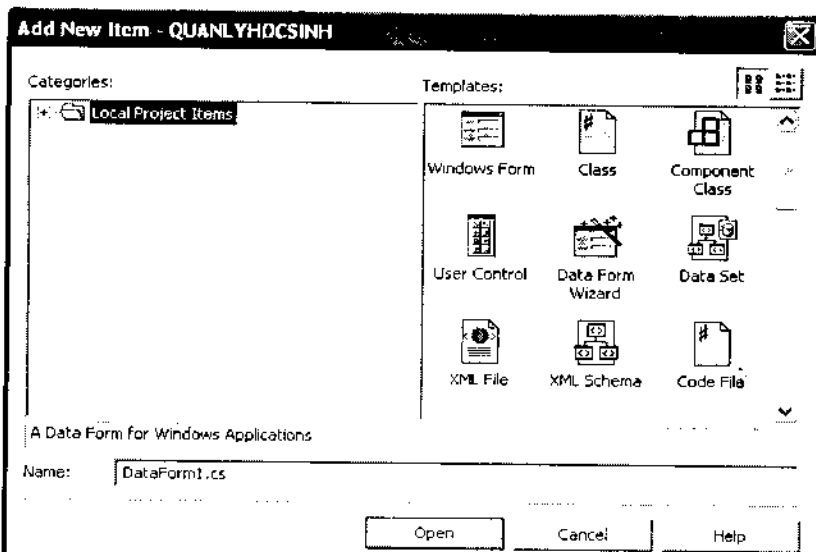
◆ Giả sử Project có tên **QUANLYHOCSINH**.

◆ Click nút OK, bạn sẽ có màn hình sau:



H377

◆ Vào menu Project, chọn Add New Item, chúng ta thấy màn hình sau:



H378

Chọn Data Form Wizard từ template có sẵn, Nhỏ vào khung Name tên QLHSCIENTHUONG.cs như hình sau:



Categories:

Templates:



Local Project Items

Windows Form

Class

Component
Class

User Control

Data Form
Wizard

Data Set

XML File

XML Schema

Code File

A Data Form for Windows Applications

Name: QLHSKHENTHUONG.cs

Open

Cancel

Help

H379

◆ Click nút Open. Màn hình sau hiện ra

Data Form Wizard



Welcome to the DataForm Wizard

This wizard will create a new form with data-bound controls to display information from the dataset you specify.

Click Next to continue.

Cancel

Back <

Next >

Finish

H380

◆ Click nút Next, màn hình sau hiện ra

Data Form Wizard

Choose the dataset you want to use
The dataset contains the tables and columns to work with in your form.

The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

☐ Use the following dataset:

Cancel < Back Next > Finish

H381

◆ Bạn đặt tên cho Dataset, giả sử chúng ta đặt tên là LAPTRINHQLHS.

Data Form Wizard

Choose the dataset you want to use
The dataset contains the tables and columns to work with in your form.

The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

☐ Use the following dataset:

Cancel < Back Next > Finish

H382

◆ Click nút Next, màn hình sau hiện ra

Data Form Wizard

Choose a data connection

The data connection specifies how to find and log into a specific server and database.

Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.

Which connection should the wizard use?

ACCESS E:\QLSINHVIEN.mdb.Admin

New Connection...

Cancel

< Back

Next >

Finish

H383

Vì bài trước chúng ta đã kết nối đến CSDL có tên QLSINHVIEN.mdb, nên các bạn thấy như vậy. Bài này kết nối đến CSDL SQL SERVER do đó bạn click nút New Connection. Trong hộp thoại hiện ra, click chọn tab có tên Provider, màn hình sau hiện ra

Data Link Properties

Provider | Connection | Advanced | All

Select the data you want to connect to:

OLE DB Provider(s)

Connectivity Service Provider
MediaCatalogDB OLE DB Provider
MediaCatalogMergedDB OLE DB Provider
MediaCatalogWebDB OLE DB Provider
Microsoft ISAM 1.1 OLE DB Provider
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for DTS Packages
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Outlook Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider

Next >>

OK

Cancel

Help

H384

◆ Bạn chọn Microsoft OLE DB Provider for SQL Server như hình trên.

◆ Click nút OK.

Ghi chú:

Microsoft OLE DB Provider for SQL Server là bộ máy cho phép truy cập đến CSDL của Microsoft SQL Server. Nếu bạn dùng CSDL Oracle thì chọn Microsoft OLE DB Provider for Oracle. Nếu bạn dùng CSDL Access thì chọn Microsoft jet 4.0 OLE DB Provider.

◆ Click nút Next để chuyển sang tab Connection như màn hình sau:

Data Link Properties

Provider | **Connection** | Advanced | All |

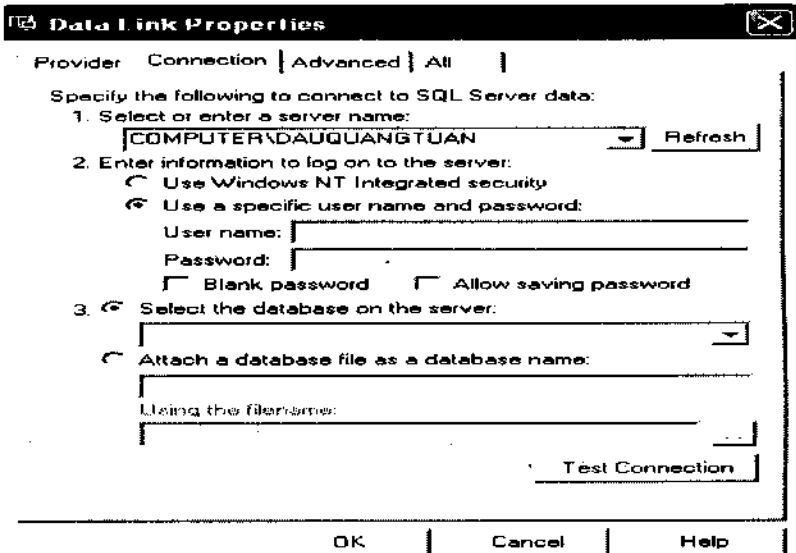
Specify the following to connect to SQL Server data:

1. Select or enter a server name:
[Text Box] Refresh
2. Enter information to log on to the server:
☐ Use Windows NT Integrated security
☒ Use a specific user name and password:
User name: [Text Box]
Password: [Text Box]
☐ Blank password ☐ Allow saving password
3. ☒ Select the database on the server:
[Dropdown Menu]
☐ Attach a database file as a database name:
[Text Box]
Using the filename: [Text Box]
Test Connection

OK Cancel Help

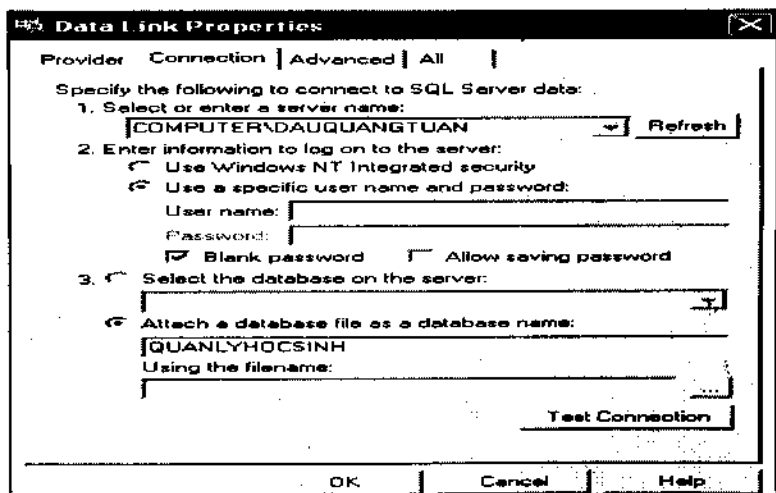
H385

◆ Nhập COMPUTER\DAUQUANGTUAN vào khung Select or enter a server name



H386

◆ Click chọn mục Blank password và mục Attach a database file as a database name, nhập QUANLYHOCSINH trong khung Attach a database file as a database name như hình sau:



H387

◆ Click nút OK, màn hình sau hiện ra

Data Form Wizard

Choose tables or views

The tables or views you choose will determine which columns will be available to display on your form.



The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

- Tables
 - DIEMTB
 - DIEMTHILAI
 - HOCSINH
 - KHENTHUONG
 - KYLUAT
- Views

Selected item(s):

- Tables
- Views

Buttons: Cancel, < Back, Next >, Finish

H391

Biểu mẫu này chứa hai bảng KHENTHUONG và HOCSINH nên chúng ta lần lượt chọn hai bảng này, và click nút ">" sau mỗi lần chọn. Bạn sẽ có hình như sau:

Data Form Wizard

Choose tables or views

The tables or views you choose will determine which columns will be available to display on your form.



The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

- Tables
 - DIEMTB
 - DIEMTHILAI
 - KYLUAT
- Views

Selected item(s):

- Tables
 - HOCSINH
 - KHENTHUONG
- Views

Buttons: Cancel, < Back, Next >, Finish

H392

◆ Click nút Next, màn hình sau hiện ra

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.

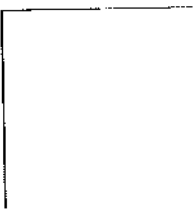
Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

Parent table: Child table: 

Keys:



Relations: 

H393

Giả sử bạn gõ "KETNOIHAIBANGHOCSINHVAKHENTHUONG" vào khung Name.

◆ Click đầu mũi tên quay xuống của khung Parent table, chọn bảng HOCSINH.

◆ Bên bảng Child table chọn bảng KHENTHUONG

◆ Click đầu mũi tên trong khung Keys (khung bên dưới khung Parent table), chọn khóa là MSHS, khung bên phải bên cạnh, chọn KTMSHS. Bạn sẽ thấy như hình sau:

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.

Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

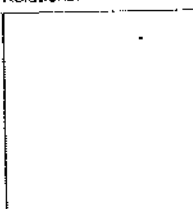
Name:

Parent table: Child table: 

Keys:

MSHS	KTMSHS



Relations: 

H394

◆ Click nút ">" để đưa sang khung Relations, bạn thấy như hình sau:

Data Form Wizard

Create a relationship between tables
The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.

Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

Parent table: Child table: > X

Keys:

Relations: KETNOIHAIBANGHOCS!

Cancel < Back Next > Finish

H395

◆ Click nút Next, màn hình sau hiện ra

Data Form Wizard

Choose tables and columns to display on the form
Your form can display any of the tables and columns available in the dataset.

If you display more than one table on the form, the tables have a master-detail relationship

What tables and columns do you want to display on the form?

Master or single table: KHENSTINH

Detail table: KHENTHUONG

Columns: ☐ MSHS ☐ Holot ☐ Ten ☐ Lop ☐ Nam ☐ Ngaysinh

Columns: ☐ MSKT ☐ Ndkhen ☐ Ngkhen ☐ Phanthuong ☐ KTMSHS

Cancel < Back Next > Finish

H396

Data Form Wizard tự chọn hết các vùng của hai bảng, nếu không thích vùng nào trong hai bảng, bạn có thể click bỏ dấu chọn ở vùng đó. Ở đây chúng ta chọn hết.

◆ Click nút Next, màn hình sau hiện ra

Data Form Wizard

Choose the display style

You can display a single record at a time or display all records at once.



How do you want to display your data?

- ☒ All records in a grid
☐ Single record in individual controls

What additional controls do you want on the form?

☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

- ☐ Add - Creates a new record.
☐ Delete - Deletes the current record.
☐ Cancel - Cancels changes to the current record.
☐ Navigation controls - Moves to first, previous, next, or last record.

The wizard now has the information it needs. Click Finish to exit and generate your new form.

Cancel

< Back

Next >

Finish

H397

◆ Click chọn mục thứ hai theo như yêu cầu của đầu bài, đó là mục Single record in individual control.

Data Form Wizard

Choose the display style

You can display a single record at a time or display all records at once.



How do you want to display your data?

- ☐ All records in a grid
☒ Single record in individual controls

What additional controls do you want on the form?

☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

- ☒ Add - Creates a new record.
☒ Delete - Deletes the current record.
☒ Cancel - Cancels changes to the current record.
☒ Navigation controls - Moves to first, previous, next, or last record.

The wizard now has the information it needs. Click Finish to exit and generate your new form.

Cancel

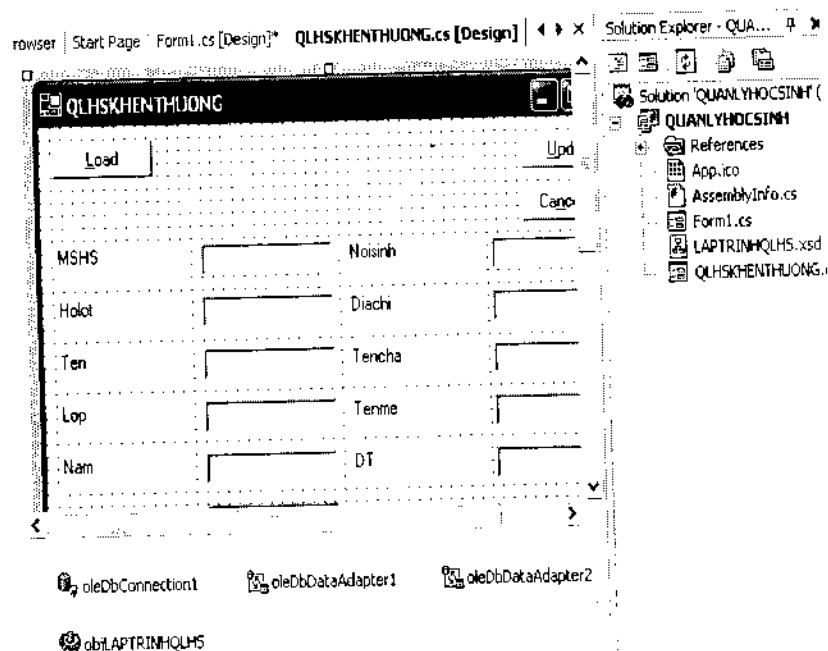
< Back

Next >

Finish

H398

◆ Click nút Finish để kết thúc, bạn thấy Data Form Wizard tự thiết kế cho chúng ta biểu mẫu như hình sau:



H399

◆ Click nút phải chuột tại vị trí trống trong biểu mẫu, chọn Properties trong menu hiện ra, xác lập thuộc tính Font cho Form là VNI-Times, size 12.

◆ Click chọn Form1 trong cửa sổ Solution Explorer.

◆ Click nút View Code bên trên, vào sửa code lại thay vì:

```
static void Main()
```

```
{
    Application.Run(new Form1());
}
```

Thành:

```
static void Main()
```

```
{
    Application.Run(new QLHSHKENTHUONG());
}
```

Nhấn phím F5 chạy thử chương trình, bạn thấy hình sau hiện ra

QLHSKHENTHUONG

Load Update
Cancel

MSHS: _____ Noisinh: _____
 Holot: _____ Diachi: _____
 Ten: _____ Tencha: _____
 Lop: _____ Tenme: _____
 Nam: _____ DT: _____
 Ngaysinh: _____

<< < No Records > >>
 Add Delete Cancel

MSKT	Ndkhen	Ngkhen	Phanthuo	KTMSHS
*				

H400

◆ Click nút Load trên biểu mẫu, bạn sẽ thấy

QLHSKHENTHUONG

Load Update
Cancel

MSHS: 00001 Noisinh: Hà Nội
 Holot: Trần Mạnh Diachi: KTXĐH
 Ten: An Tencha: Trần Hoàng L
 Lop: 10A1 Tenme: Nguyễn Phúc
 Nam: X DT: 8984154
 Ngaysinh: 12/12/1989 12

<< < 1 of 8 > >>
 Add Delete Cancel

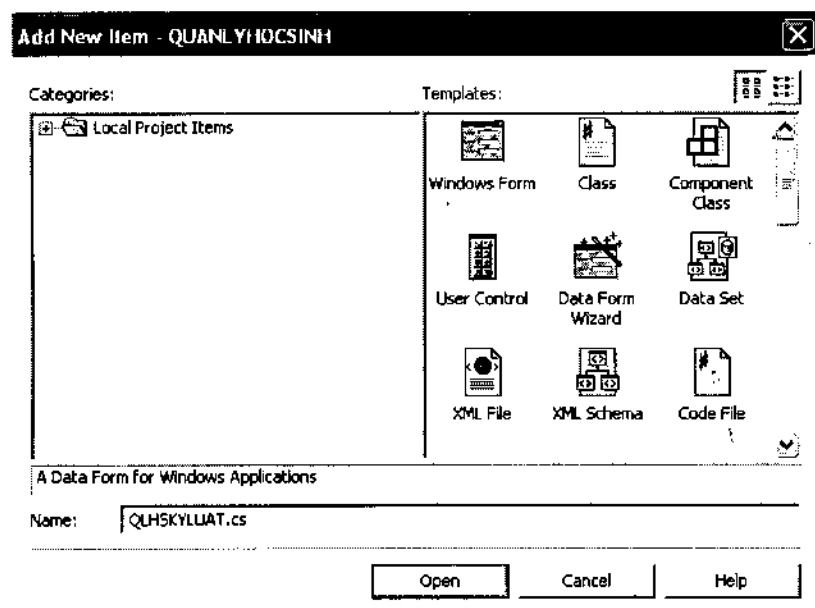
MSKT	Ndkhen	Ngkhen	Phanthuo	KTMSHS
▶ KT001	Học sinh	4/20/2005	Một máy	00001
*				

H401

Bạn thấy thông tin trong bảng HOCSINH là em có MSHS là 00001, vì có mối quan hệ một nhiều dựa trên khóa MSHS nên trong bảng KHENTHUONG hiện đúng ngay nội dung khen thưởng của học sinh đó. Bạn nhìn cột KTMSHS trong điều khiển lưới bên dưới, bạn thấy có giá trị là 00001.

Nếu click các nút có mũi tên để đến mẫu tin khác, nếu học sinh đó có khen thưởng thì nội dung khen thưởng của học sinh đó sẽ hiện ra, nếu không có thì điều khiển lưới bên dưới sẽ trống trơn.

Tiếp theo chúng ta sẽ thêm một biểu mẫu mới bằng cách vào menu Project, chọn Add New Item, các bạn thực hiện các bước tương tự như trên. Trong phần này vì chúng ta đã kết nối với CSDL QUANLYHOCSINH.dbo trong SQL SERVER rồi, nên khi hộp thoại tạo kết nối hiện ra, các bạn chọn tên CSDL đó trong danh sách nếu như chưa chọn bằng cách click vào đầu mũi tên quay xuống, sau đó click nút Next để tiếp tục qua các khai báo tiếp. Chú ý tên của tập tin là QLHCKYLUAT.cs, Tên của Dataset là LAPTRINHQLHSHAI... Bạn có thể xem những hộp thoại cần thiết khi khai báo như các hình sau:



H402

Data Form Wizard

Choose the dataset you want to use

The dataset contains the tables and columns to work with in your form.



The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

LAPTRINHQLHSHAI

☐ Use the following dataset:

QUANLYHOC SINH LAPTRINHQLHS

Cancel

< Back

Next >

Finish

H403

Data Form Wizard

Choose a data connection

The data connection specifies how to find and log into a specific server and database.



Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.

Which connection should the wizard use?

ACCESS.E:\QLSINHVIEN.mdb.Admin

ACCESS.E:\QLSINHVIEN.mdb.Admin

ACCESS.E:\SEAGAMES23.mdb.Admin

COMPUTER\E:\QL\QUANGTUAN\QUANLYHOC\SQLH.dbo

New Connection...

Cancel

< Back

Next >

Finish

H405

Data Form Wizard

Choose a data connection

The data connection specifies how to find and log into a specific server and database.



Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.

Which connection should the wizard use?

COMPUTER\DAUQUANGTUAN\QUANLYHOCSINH.dbo

New Connection...

Cancel

< Back

Next >

Finish

H404

Data Form Wizard

Choose tables or views

The tables or views you choose will determine which columns will be available to display on your form.



The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

Tables
DIEMTB
DIEMTHILAI
KHENTHUONG
Views



Selected item(s):

Tables
HOCSINH
KYLAT
Views

Cancel

< Back

Next >

Finish

H406

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.



Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

Relations:

KETNOIHAIBANGHOCSINHVAKYLUAT

Parent table:

Child table:

HOCSINH

KYLUAT

Keys:

MSHS

KyluatMSHS

Cancel

< Back

Next >

Finish

H407

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.



Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

Relations:

KETNOIHAIBANGHOCSI

Parent table:

Child table:

Keys:

Cancel

< Back

Next >

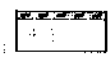
Finish

H408

Data Form Wizard

Choose tables and columns to display on the form

Your form can display any of the tables and columns available in the dataset.



If you display more than one table on the form, the tables have a master-detail relationship

What tables and columns do you want to display on the form?

Master or single table:

HOCSINH

Detail table:

KYLUAT

Columns:

- ☒ MSHS
- ☒ Holot
- ☒ Ten
- ☒ Lop
- ☒ Nam
- ☒ Ngaysinh

Columns:

- ☒ MSKL
- ☒ Vipham
- ☒ Ngvipham
- ☒ Xuly
- ☒ KyluatMSHS

Cancel

< Back

Next >

Finish

H410

Data Form Wizard

Choose the display style

You can display a single record at a time or display all records at once.



How do you want to display your data?

- ☐ All records in a grid
- ☒ Single record in individual controls

What additional controls do you want on the form?

- ☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

- ☒ Add - Creates a new record.
- ☒ Delete - Deletes the current record.
- ☒ Cancel - Cancels changes to the current record.
- ☒ Navigation controls - Moves to first, previous, next, or last record.

The wizard now has the information it needs. Click Finish to exit and generate your new form.

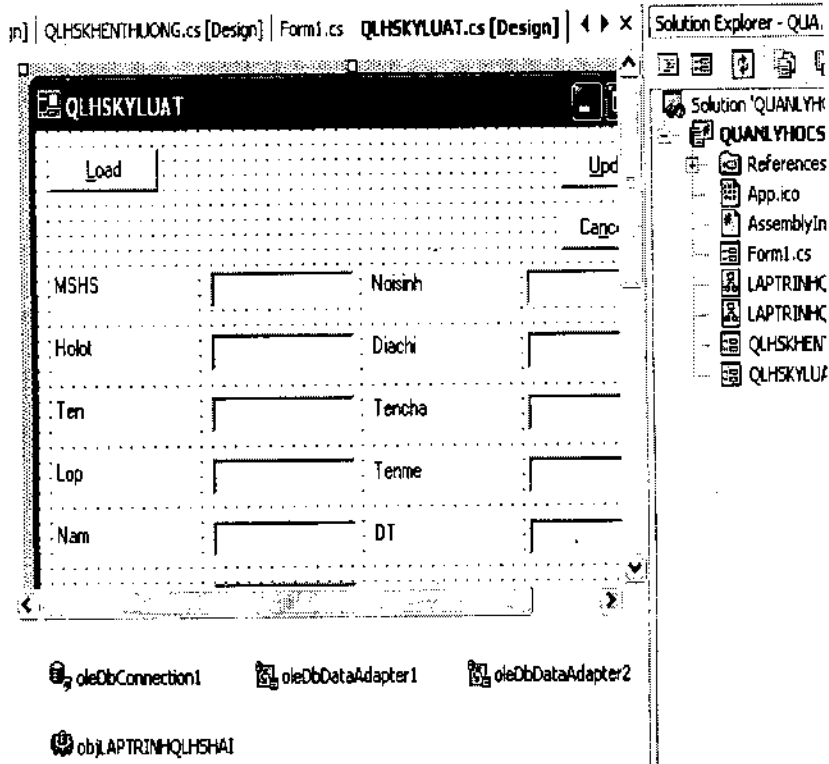
Cancel

< Back

Next >

Finish

H411



H412

◆ Click nút phải chuột tại vị trí trống trong Form, chọn Properties trong menu hiện ra, xác lập thuộc tính Font là VNI-Times có size là 12.

◆ Click chọn Form1.cs

◆ Click nút View Code thay vì

static void Main()

```
{
    Application.Run(new QLHSHENTHUONG());
}
```

Sửa lại thành:

static void Main()

```
{
    Application.Run(new QLHSKYLUAT());
}
```

Nhấn phím F5 chạy thử chương trình, bạn thấy hình sau hiện ra

QLHSKYLUAT

Load Update Cancel

MSHS		Noisinh	
Holot		Diachi	
Ten		Tencha	
Lop		Tenme	
Nam		DT	
Ngaysinh			

<< < No Records > >>

Add Delete Cancel

MSKL	Vipham	Ngvipha	Xuly	KyluatM
*				

H413

◆ Click nút Load, chúng ta thấy kết quả như hình sau:

QLHSKYLUAT

Load Update Cancel

MSHS	00001	Noisinh	Hà Nội
Holot	Trần Mạnh	Diachi	KTXĐH
Ten	An	Tencha	Trần Hoàng L
Lop	10A1	Tenme	Nguyễn Phúc
Nam	X	DT	8984154
Ngaysinh	12/12/1989 12		

<< < 1 of 8 > >>

Add Delete Cancel

MSKL	Vipham	Ngvipha	Xuly	KyluatM
*				

H414

Bạn thấy nội dung của bảng HOSINH hiện ra trong các TextBox bên trên, học sinh có mã số 00001 này không bị vi phạm kỷ luật nên trong điều khiển lưới bên dưới không có.

Giả sử di chuyển đến một học sinh khác, bạn sẽ thấy các mẫu tin tương ứng trong bảng KYLUAT nếu có sẽ hiện ra trong điều khiển lưới bên dưới.

QLHSKYLUAT

Load

Update

Cancel

MSHS	00002	Noisinh	TP.HCM
Holot	Hoàng Minh	Diachi	P26 QBT
Ten	Hùng	Tencha	Hoàng Minh I
Lop	10A1	Tenme	Nguyễn Thị L
Nam	X	DT	8998978
Ngaysinh	4/23/1989 12:00		

<<

<

2 of 8

>

>>

Add

Delete

Cancel

	MSKL	Vipham	Ngvipha	Xuly	KyluatM
▶	KL001	Đánh nha	2/13/2005	Mời PHH	00002
*					

H415

Tiếp theo chúng ta sẽ thêm một biểu mẫu mới bằng cách vào menu Project, chọn Add New Item, các bạn thực hiện các bước tương tự như trên. Trong phần này vì chúng ta đã kết nối với CSDL QUANLYHOCSINH.dbo trong SQL SERVER rồi, nên khi hộp thoại tạo kết nối hiện ra, các bạn chọn tên CSDL đó trong danh sách nếu như chưa chọn bằng cách click vào đầu mũi tên quay xuống, sau đó click nút Next để tiếp tục qua các khai báo tiếp. Chú ý tên của tập tin là QLHSKETQUA.cs, Tên của Dataset là LAPTRINHQLHSBA... Bạn có thể xem những hộp thoại cần thiết khi khai báo như các hình sau:



Categories:

Local Project Items

Templates:



Windows Form



Class

Component
Class

User Control

Data Form
Wizard

Data Set



XML File



XML Schema



Code File



A Data Form for Windows Applications

Name:

QLH5KETQUA.cs

Open

Cancel

Help

H416

Data Form Wizard

Choose the dataset you want to use

The dataset contains the tables and columns to work with in your form.



The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

LAPTRINHQLHSBA

☐ Use the following dataset:

QUANLYHOCSINH.LAPTRINHQLHS

Cancel

< Back

Next >

Finish

H417

Data Form Wizard

Choose a data connection

The data connection specifies how to find and log into a specific server and database.



Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.

Which connection should the wizard use?

COMPUTER\DAUQUANGTUAN\QUANLYTHOCSINH.dbc

New Connection...

Cancel

< Back

Next >

Finish

H418

Data Form Wizard

Choose tables or views

The tables or views you choose will determine which columns will be available to display on your form.



The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

Selected item(s):

Tables

DIEMTHILAI
KHENTHUONG
KYLUAU



Tables

HOCSINH
DIEMTB
Views

Cancel

< Back

Next >

Finish

H419

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.



Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

KETNOIHAIBANGHOCSINHVADIEMTB

Relations:

Parent table:

HOCSINH

Child table:

DIEMTB



Keys:

MSHS

DiemMSHS

Cancel

< Back

Next >

Finish

H420

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.



Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

Relations:

KETNOIHAIBANGHOCSI

Parent table:

Child table:



Keys:

Cancel

< Back

Next >

Finish

H421

Data Form Wizard

Choose tables and columns to display on the form

Your form can display any of the tables and columns available in the dataset.



If you display more than one table on the form, the tables have a master-detail relationship

What tables and columns do you want to display on the form?

Master or single table:

HOCSINH

Detail table:

DIEMTB

Columns:

- ☒ MSHS
- ☒ Holot
- ☒ Ten
- ☒ Lop
- ☒ Nam
- ☒ Ngaysinh

Columns:

- ☒ MSKQ
- ☒ Van1
- ☒ Van2
- ☒ Vancn
- ☒ Toan1
- ☒ Toan2

H422

Data Form Wizard

Choose the display style

You can display a single record at a time or display all records at once.



How do you want to display your data?

- ☐ All records in a grid
- ☒ Single record in individual controls

What additional controls do you want on the form?

☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

- ☒ Add - Creates a new record.
- ☒ Delete - Deletes the current record.
- ☒ Cancel - Cancels changes to the current record.
- ☒ Navigation controls - Moves to first, previous, next, or last record.

The wizard now has the information it needs. Click Finish to exit and generate your new form.

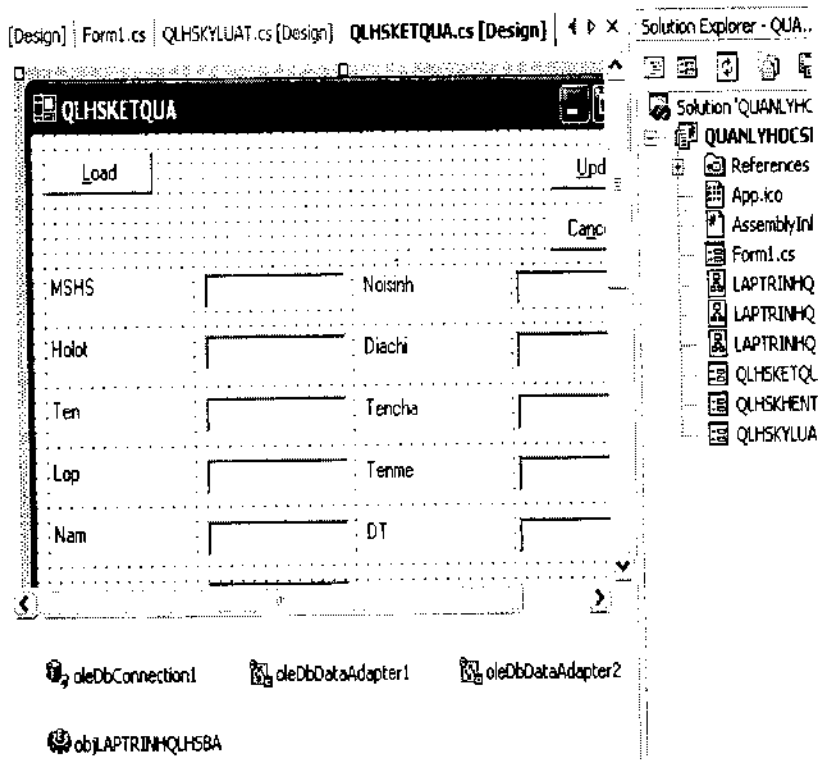
Cancel

< Back

Next >

Finish

H423



H424

◆ Click nút phải chuột tại vị trí trống trong Form, chọn Properties trong menu hiện ra, xác lập thuộc tính Font là VNI-Times có size là 12.

◆ Click chọn Form1.cs

◆ Click nút View Code thay vì

static void Main()

```
{
    Application.Run(new QLHSKYLUAT());
}
```

Sửa lại thành:

static void Main()

```
{
    Application.Run(new QLHSKETQUA());
}
```

Nhấn phím F5 chạy thử chương trình, bạn thấy hình sau hiện ra

QLHSKFIQUA

Load Update Cancel

MSHS Noisinh
 Holot Diachu
 Ten Tencha
 Lop Tenme
 Nam DT
 Ngaysinh

<< < No Records > >>
 Add Delete Cancel

MSKQ	Van1	Van2	Vancn	Toan1	T
*					

H425

Click nút Load, bạn có kết quả như hình sau:

QLHSKFIQUA

Load Update Cancel

MSHS 00001 Noisinh Hà Nội
 Holot Trần Mạnh Diachu KTXĐH
 Ten An Tencha Trần Hoàng L
 Lop 10A1 Tenme Nguyễn Phúc
 Nam X DT 8984154
 Ngaysinh 12/12/1989 12

<< < 1 of 8 > >>
 Add Delete Cancel

MSKQ	Van1	Van2	Vancn	Toan1	T
KQ001	9	9	9	7	7

H426

Ở đây do khi thiết kế bảng DIEMTB trong SQL Server, chúng tôi tạo khóa ngoại nằm sau cùng, vì có quá nhiều vùng, nên các bạn không thấy để kiểm chứng, rút kinh nghiệm, khóa ngoại nên cho lên trên, nằm ngay dưới khóa chính của bảng này.

Trong chương trình bạn click chuột trong mẫu tin của bảng DIEMTB, dùng phím hướng → để di chuyển đến vùng khóa ngoại tương ứng. Bạn có thể thấy như hình sau:

The screenshot shows a Windows application window titled "QLHSKETQUA". It contains a form with several input fields and buttons. The form is organized into two columns of fields. The first column has fields for "MSHS", "Holot", "Ten", "Lop", "Nam", and "Ngaysinh". The second column has fields for "Noisinh", "Diachi", "Tencha", "Tenme", and "DT". There are also buttons for "Load", "Update", "Cancel", "Add", "Delete", and "Cancel". Below the form is a data grid with columns: "Danhhiieu", "Danhhiieu", "Danhhiieu", "Ketquacu", and "DiemMS". The first row of data shows "Xuất sắc", "Xuất sắc", "Xuất sắc", "Xuất sắc", and "10001".

MSHS	00001	Noisinh	Hà Nội
Holot	Trần Mạnh	Diachi	KTXĐH
Ten	An	Tencha	Trần Hoàng L
Lop	10A1	Tenme	Nguyễn Phúc
Nam	X	DT	8984154
Ngaysinh	12/12/1989 12		

Navigation buttons: << < 1 of 8 > >>

Action buttons: Add Delete Cancel

	Danhhiieu	Danhhiieu	Danhhiieu	Ketquacu	DiemMS
	Xuất sắc	Xuất sắc	Xuất sắc	Xuất sắc	10001

H427

Tiếp theo chúng ta sẽ thêm một biểu mẫu cuối cùng bằng cách vào menu Project, chọn Add New Item, các bạn thực hiện các bước tương tự như trên. Trong phần này vì chúng ta đã kết nối với CSDL QUANLYHOCSINH.dbo trong SQL SERVER rồi, nên khi hộp thoại tạo kết nối hiện ra, các bạn chọn tên CSDL đó trong danh sách nếu như chưa chọn bằng cách click vào đầu mũi tên quay xuống, sau đó click nút Next để tiếp tục qua các khai báo tiếp. Chú ý tên của tập tin là QLHSDIEMTHILAI.cs, Tên của Dataset là LAPTRINHQLHSBON... Bạn có thể xem những hộp thoại cần thiết khi khai báo như các hình sau:

Add New Item - QUANLYHOCSINH

Categories:

Templates:

Local Project Items

Windows Form
Class
Component Class
User Control
Data Form Wizard
Data Set
XML File
XML Schema
Code File

A Data Form for Windows Applications

Name: QLHSDIEMTHILAI.cs

Open

Cancel

Help

H428

Data Form Wizard

Choose the dataset you want to use

The dataset contains the tables and columns to work with in your form.

The wizard helps you create a new dataset. Alternatively, you can use a dataset already defined in your project or one provided by a Web service.

Which dataset do you want to use?

☒ Create a new dataset named:

LAPTRINHQLHSBON

☐ Use the following dataset:

QUANLYHOCSINH.LAPTRINHQLHSBON

Cancel

< Back

Next >

Finish

H429

Data Form Wizard

Choose a data connection

The data connection specifies how to find and log into a specific server and database.



Choose from the list of data connections currently in the Server Explorer, or add a new connection if the one you want is not listed.

Which connection should the wizard use?

COMPUTER\DAUQUANSTUAN\QUANLYHOCSTINH.dbc

New Connection...

Cancel

< Back

Next >

Finish

H430

Data Form Wizard

Choose tables or views

The tables or views you choose will determine which columns will be available to display on your form.



The wizard creates a data adapter to populate the dataset from available tables or views. If you pick more than one item, you can establish a relationship between them in the next step.

What item (or items) do you want to access?

Available item(s):

Tables
DIEMTB
KHENTHUONG
KYLUAU
Views

>

<

Selected item(s):

Tables
HOCSINH
DIEMTHILAI
Views

Cancel

< Back

Next >

Finish

H431

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.



Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

KETNOIHAIBANGHOCSINHADIEMTHILAI

Relations:

Parent table:

HOCSINH

Child table:

DIEMTHILAI



Keys:

MSHS

ThilaiMSHS



Cancel

< Back

Next >

Finish

H432

Data Form Wizard

Create a relationship between tables

The wizard will use the relationships to generate code that keeps the tables synchronized as you work with them.



Relationships are based on common keys between tables. Name your new relation, choose the parent and child tables and key fields, and then add it to the relations list using the arrow button.

Name:

Relations:

KETNOIHAIBANGHOCSI

Parent table:

Child table:



Keys:



Cancel

< Back

Next >

Finish

H433

Data Form Wizard

Choose tables and columns to display on the form

Your form can display any of the tables and columns available in the dataset.



If you display more than one table on the form, the tables have a master-detail relationship

What tables and columns do you want to display on the form?

Master or single table:

HOC SINH

Detail table:

DIEM THILAI

Columns:

- ☒ MSHS
- ☒ Holot
- ☒ Ten
- ☒ Lop
- ☒ Nam
- ☒ Ngaysinh

Columns:

- ☒ MSTL
- ☒ VanT
- ☒ VanS
- ☒ ToanT
- ☒ ToanS
- ☒ Ngoaingut

H434

Data Form Wizard

Choose the display style

You can display a single record at a time or display all records at once.



How do you want to display your data?

- ☐ All records in a grid
- ☒ Single record in individual controls

What additional controls do you want on the form?

☒ Cancel All - Cancels changes to all records in the dataset.

If you select individual controls, you can add controls for editing and navigation:

- ☒ Add - Creates a new record.
- ☒ Delete - Deletes the current record.
- ☒ Cancel - Cancels changes to the current record.
- ☒ Navigation controls - Moves to first, previous, next, or last record.

The wizard now has the information it needs. Click Finish to exit and generate your new form.

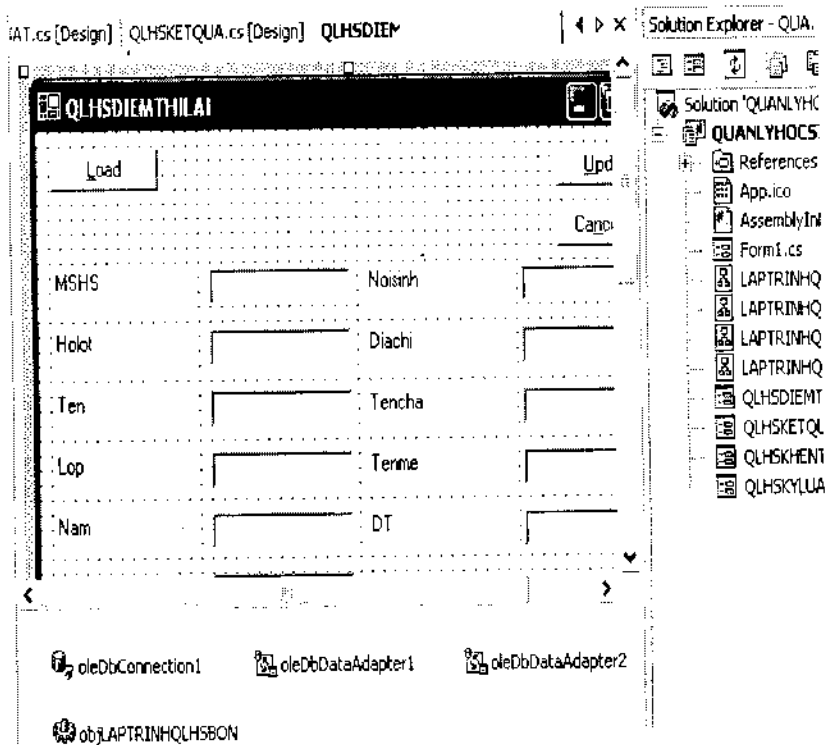
Cancel

< Back

Next >

Finish

H435



H436

◆ Click nút phải chuột tại vị trí trống trong Form, chọn Properties trong menu hiện ra, xác lập thuộc tính Font là VNI-Times có size là 12.

◆ Click chọn Form1.cs

◆ Click nút View Code thay vì
static void Main()

```
{
    Application.Run(new QLHSETQUA());
}
```

Sửa lại thành:

```
static void Main()
{
    Application.Run(new QLHSDIEMTHILAI());
}
```

Nhấn phím F5 chạy thử chương trình, bạn thấy hình sau hiện ra

QLHSKETQUA

Load Update Cancel

MSHS Noisinh
 Holot Diachi
 Ten Tencha
 Lop Tenme
 Nam DT
 Ngaysinh

<< < No Records > >>
 Add Delete Cancel

MSKQ	Van1	Van2	Vancn	Toanl	T.
*					

H437

◆ Click nút Load, ta có kết quả

QLHSKETQUA

Load Update Cancel

MSHS 00001 Noisinh Hà Nội
 Holot Trần Mạnh Diachi KTXĐH
 Ten An Tencha Trần Hoàng L
 Lop 10A1 Tenme Nguyễn Phúc
 Nam X DT 8984154
 Ngaysinh 12/12/1989 12

<< < 1 of 8 > >>
 Add Delete Cancel

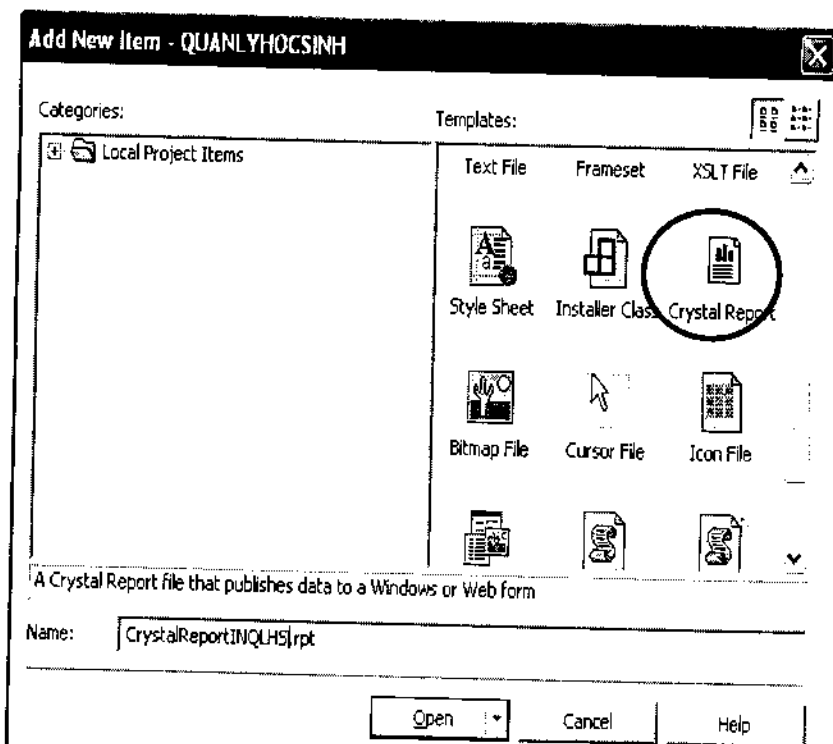
MSKQ	Van1	Van2	Vancn	Toanl	T.
KQ001	9	9	9	7	7
*					

H438

Bài 3

In nội dung các bảng bằng Crystal Report

- ◆ Mở chương trình QUANLYHOCSINH ở trên
- ◆ Vào menu **Project**, chọn **Add New Item**
- ◆ Trong cửa sổ hiện ra, di chuyển con trỏ dọc, chọn **Crystal report** trong khung bên phải, đặt tên cho tập tin là **CrystalReportINQLHS.rpt** trong khung Name như hình sau:



H439

- ◆ Click nút **Open**, hộp thoại sau hiện ra



Experience the Power of Crystal Reports

Register now to access these additional resources:

- Award winning Technical Support
- Valuable product updates
- Helpful Samples, Tutorials and Technical Briefs
- Information on Crystal Analysis and Crystal Enterprise

Please Note: Registration is required in order to access some of the benefits listed above. Registration is not a requirement of use.

[< Back](#)[Next >](#)[Register Later](#)

H440

◆ Click nút **Next**, hình sau hiện ra



Registration Options

Choose how you would like to register or enter your Registration Number in the space below.



☒ I want to register by

Web

☐ I want to enter the registration number and keycode for this product now.

Please enter the 10 digit registration number and the key code in the spaces provided below

Registration Number:

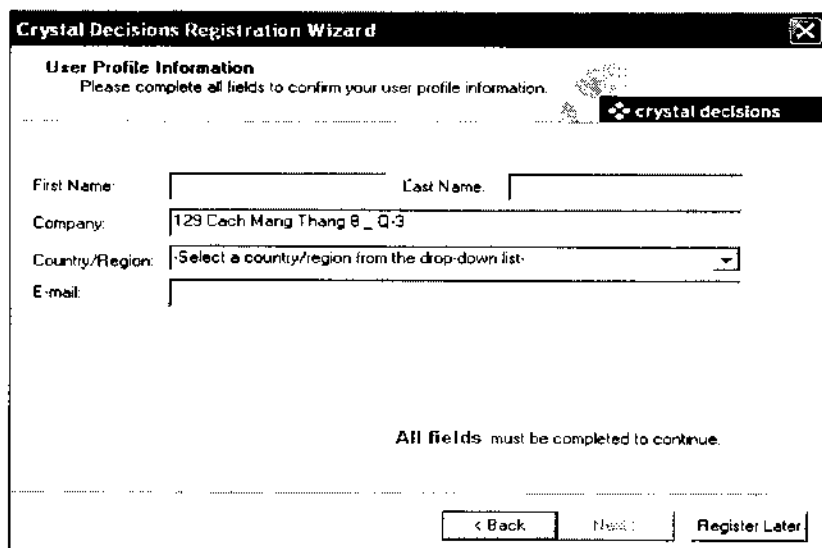
Key Code:

[Click here to read Crystal Decisions' privacy policy](#)

[< Back](#)[Next >](#)[Register Later](#)

H441

◆ Click nút **Next**, hình sau hiện ra



Crystal Decisions Registration Wizard

User Profile Information
Please complete all fields to confirm your user profile information.

First Name: _____ Last Name: _____

Company: 129 Cach Mang Thang 8 _ Q.3

Country/Region: Select a country/region from the drop-down list

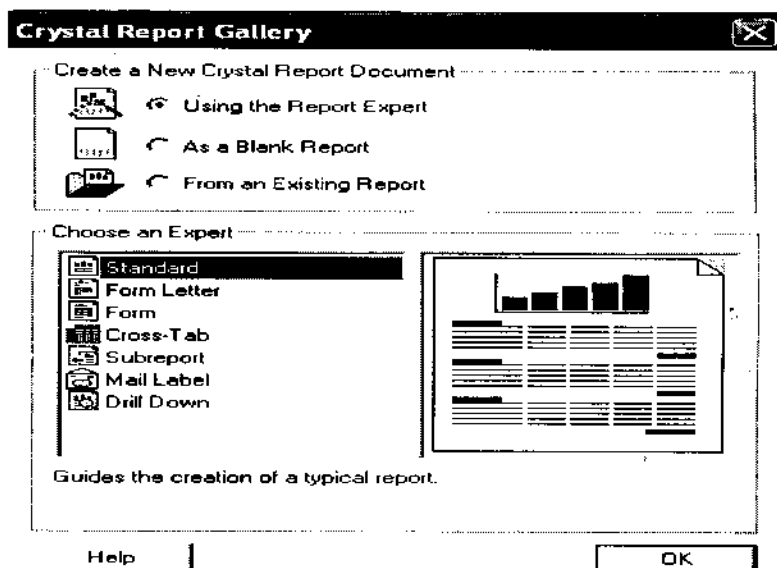
E-mail: _____

All fields must be completed to continue.

< Back Next Register Later

H442

◆ Click dấu X để đóng hộp thoại, hộp thoại sau hiện ra



Crystal Report Gallery

Create a New Crystal Report Document

-  Using the Report Expert
-  As a Blank Report
-  From an Existing Report

Choose an Expert

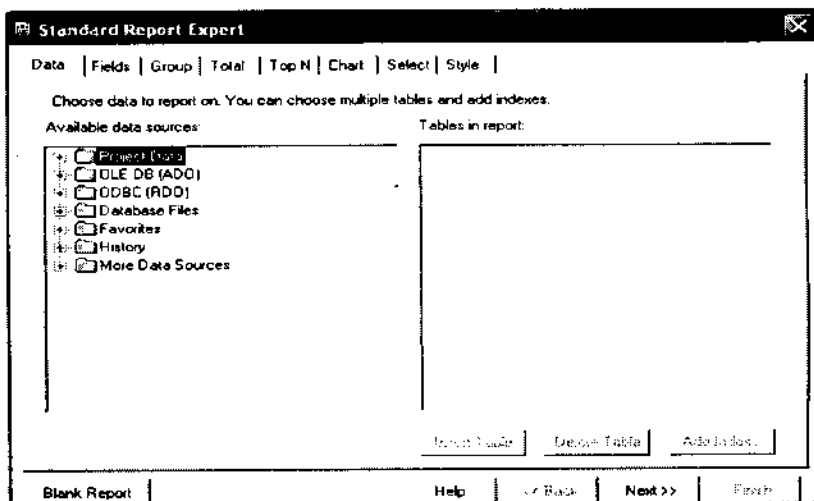
- Standard
- Form Letter
- Form
- Cross-Tab
- Subreport
- Mail Label
- Drill Down

Guides the creation of a typical report.

Help OK

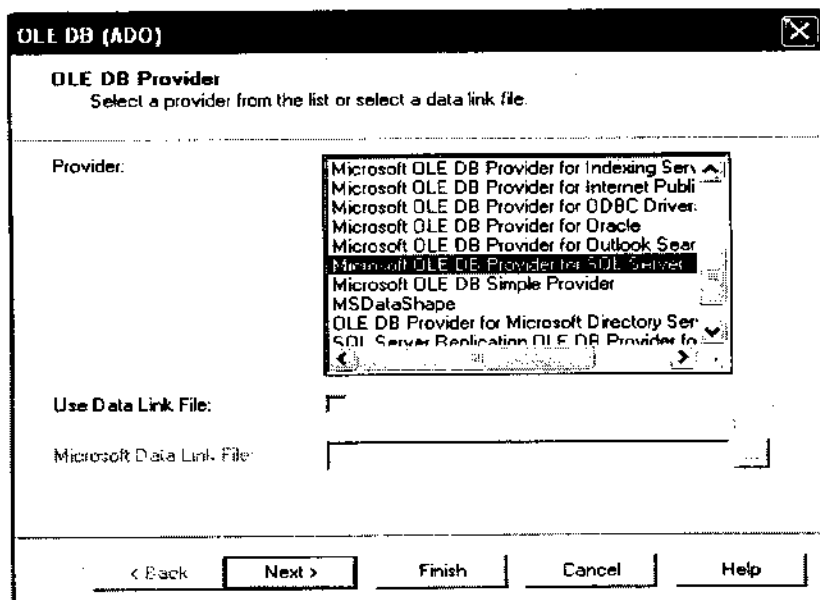
H443

◆ Chọn như trong hình, click nút OK, hộp thoại sau hiện ra



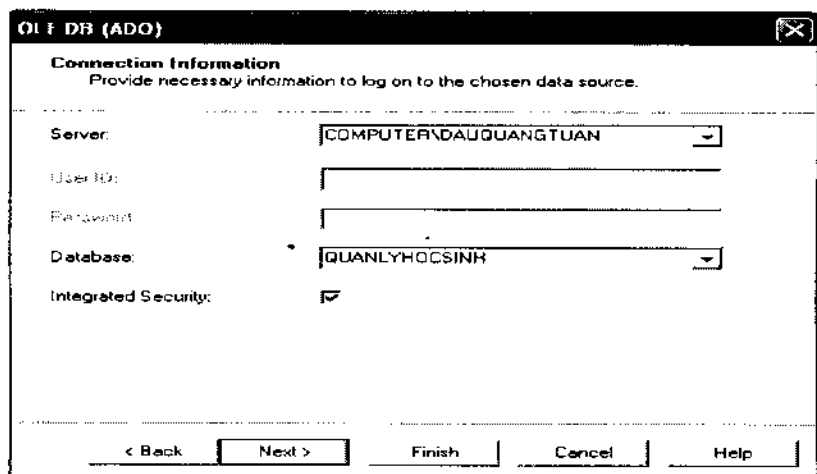
H444

◆ Click vào dấu “+” trước OLE DB (ADO), bạn thấy hình sau hiện ra



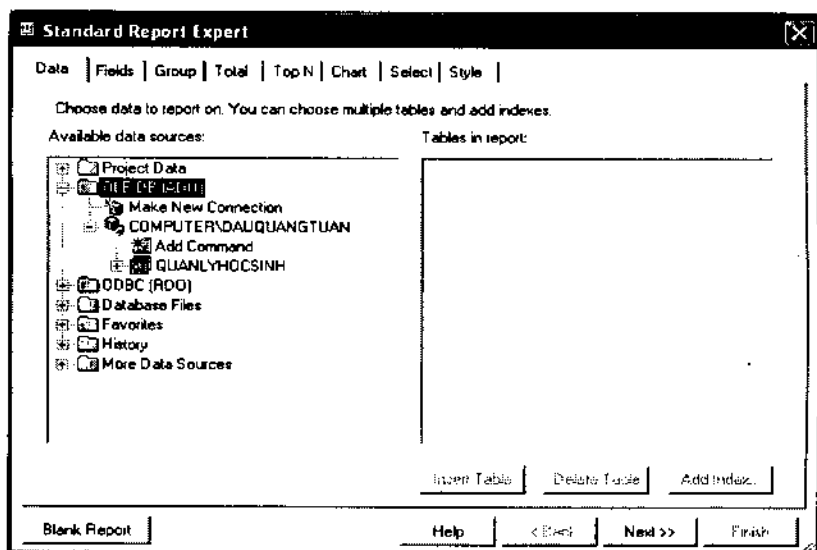
H445

◆ Chọn Microsoft OLE DB Provider for SQL Server, click nút Next



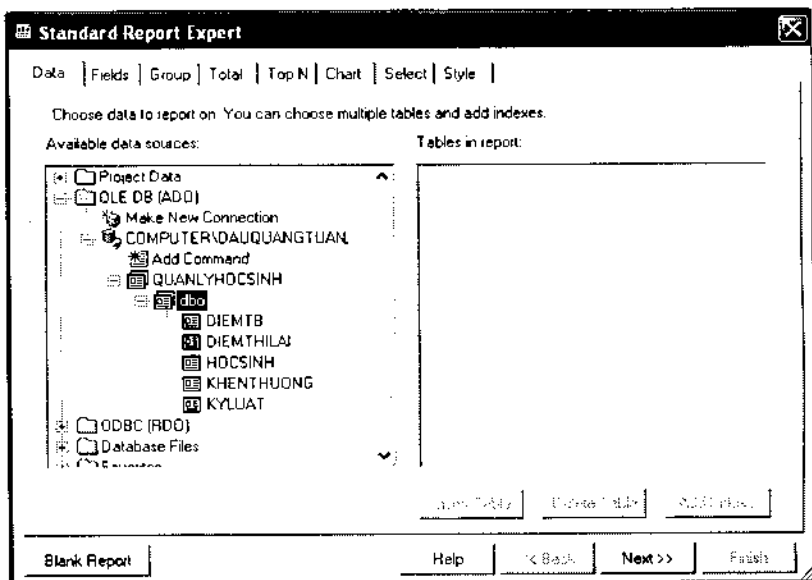
H446

◆ Gõ tên **Server** là **COMPUTER\DAUQUANGTUAN**, **Database** là **QUANLYHOCSINH**, và click dấu chọn mục **Integrated Security**, click nút **Next**



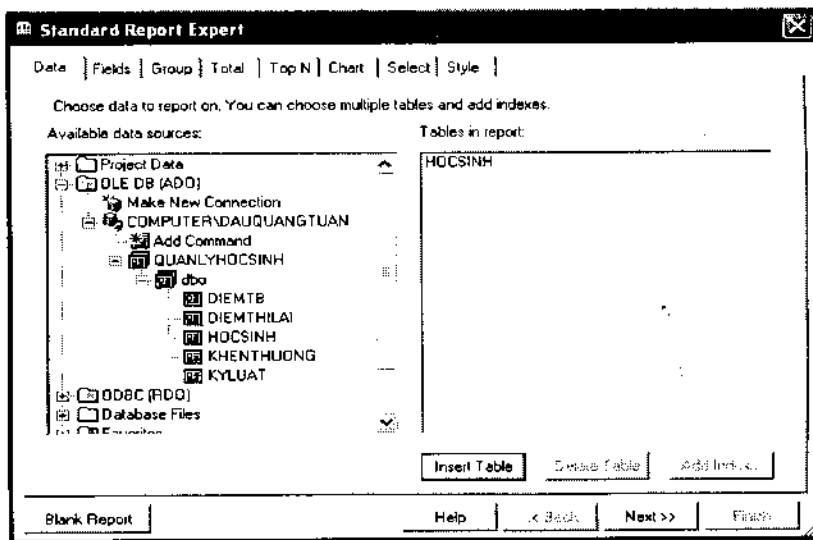
H447

◆ Click dấu "+" trước **QUANLYHOCSINH**, trước **dbo**, các bảng của **CSDL** này hiện ra.



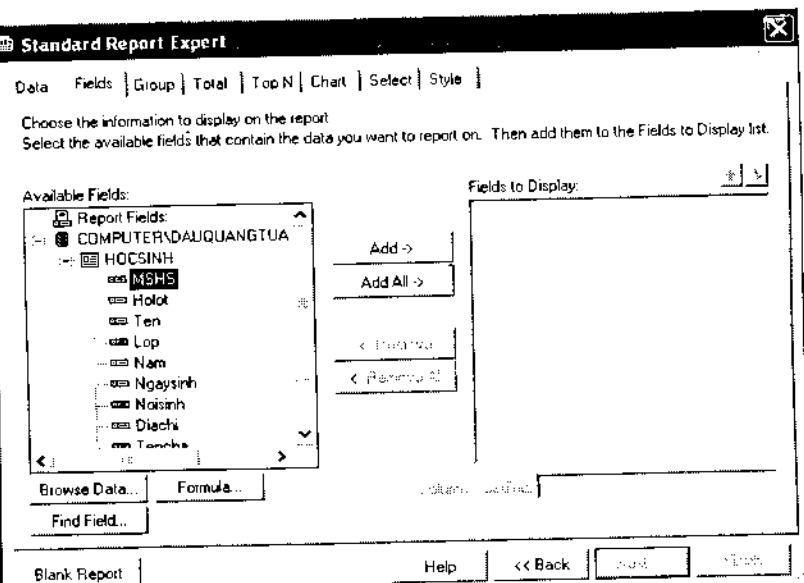
H448

◆ Chọn bảng HOCSINH, lúc này nút có tên Insert Table sẽ đậm lên.
Click nút có tên **Insert Table**, bạn thấy hình như sau:



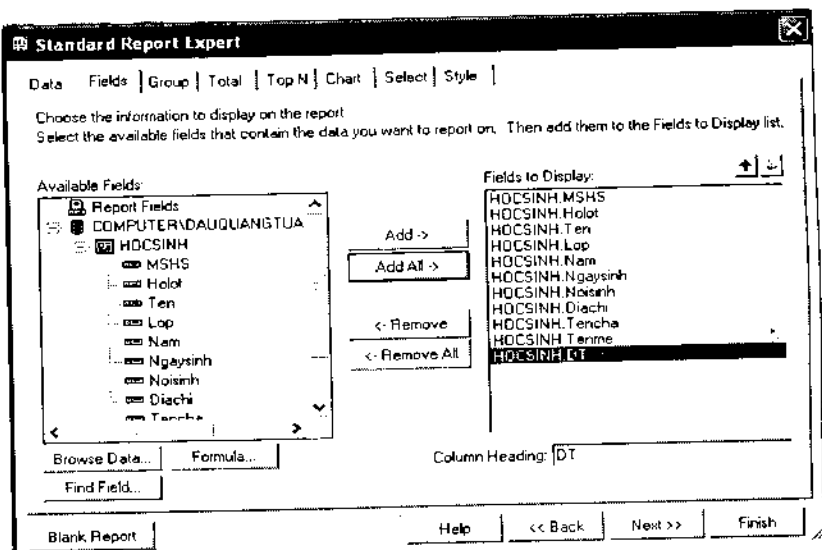
H449

◆ Click nút **Next** hộp thoại sau hiện ra



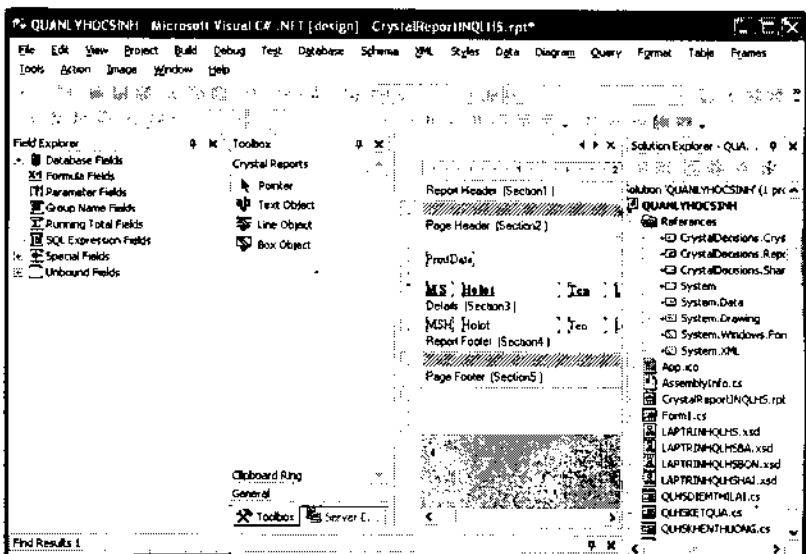
H450

Click nút Add All → để đưa hết các vùng của bảng HOCSINH qua khung bên phải.



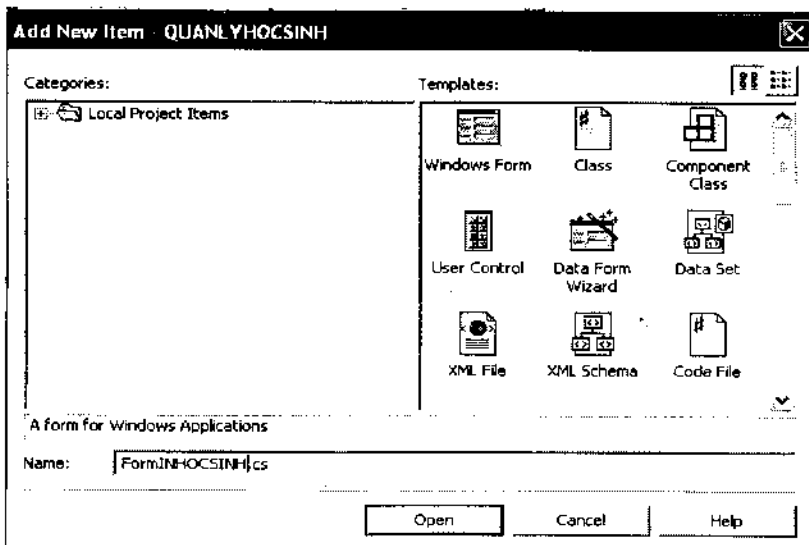
H451

◆ Click nút Finish



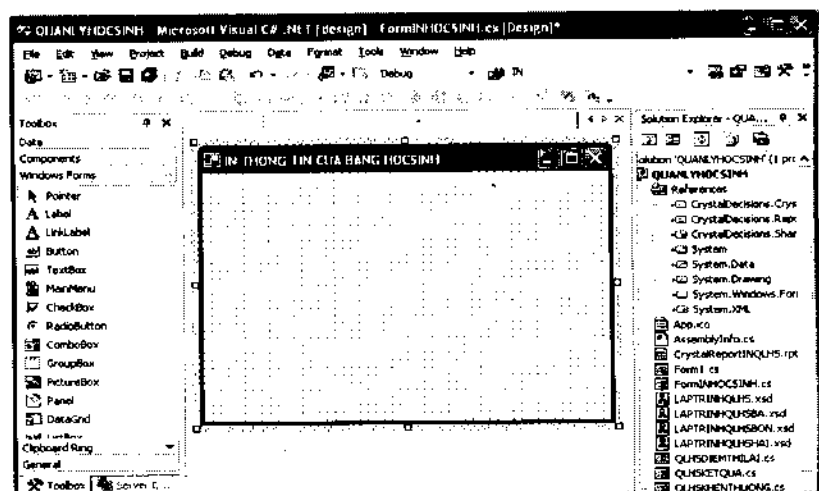
H452

- ◆ Vào menu **File**, chọn **Save CrystalReportINQLHS.rpt**
- ◆ Vào menu **Project**, chọn **Add Windows Form**, đặt tên cho Form là **FormINHOC SINH.vb** như hình sau:



H453

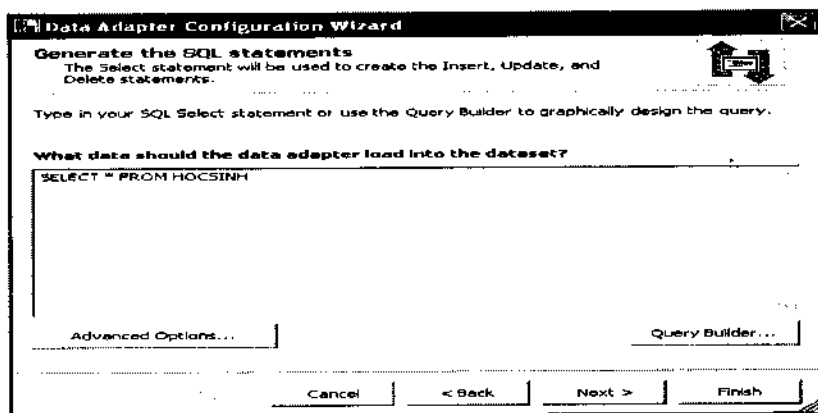
◆ Click nút **Open**, Form thiết kế hiện ra, Click nút phải chuột trên Form, chọn **Properties**, xác lập thuộc tính **Font** là **VNI-Times**, **12**, **bold** và thuộc tính **Text** là **"IN THÔNG TIN CUA BANG HOCSINH"**.



H454

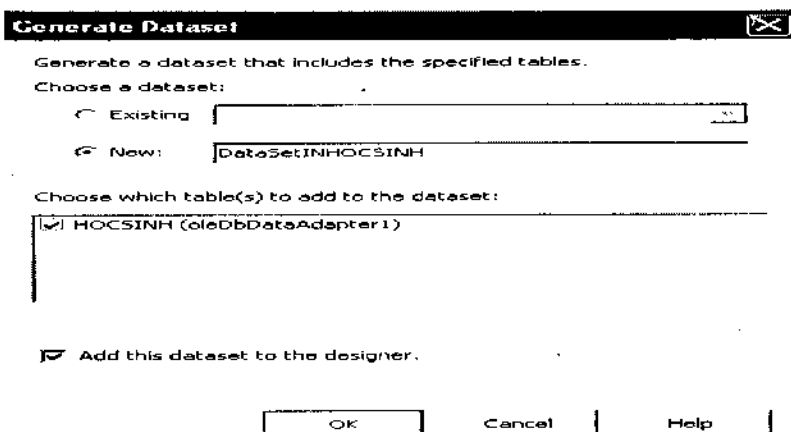
◆ Click tab có tên **Data** trong hộp công cụ, đưa điều khiển **OleDbDataAdapter** vào Form, click nút **Next**, chọn **CSDL** có tên **QUANLYHOCSINH.dbo**, click nút **Next** cho đến khi hiện ra hộp thoại nhập câu lệnh SQL, bạn nhập câu lệnh sau:

SELECT * FROM HOCSINH



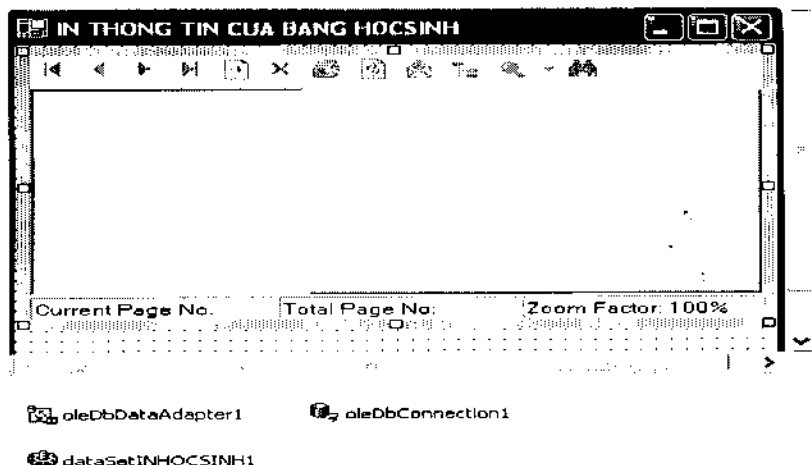
H455

- ◆ Click nút **Finish**
- ◆ Click chọn **Form**
- ◆ Vào menu **Data**, chọn **Generate Dataset**
- ◆ Click nút **New**, nhập **DataSetINHOC SINH**



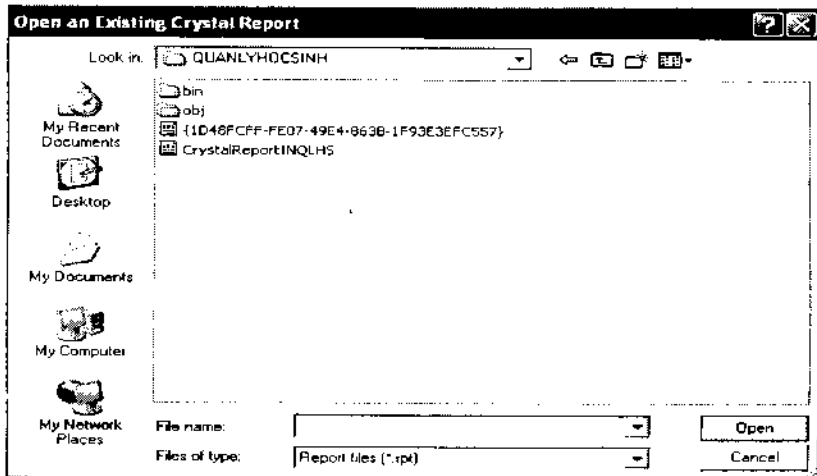
H456

- ◆ Click nút **OK**
- ◆ Click tab có tên **Windows Forms**, chọn điều khiển có tên **CrystalReportViewer** và đưa vào Form để có hình như sau:



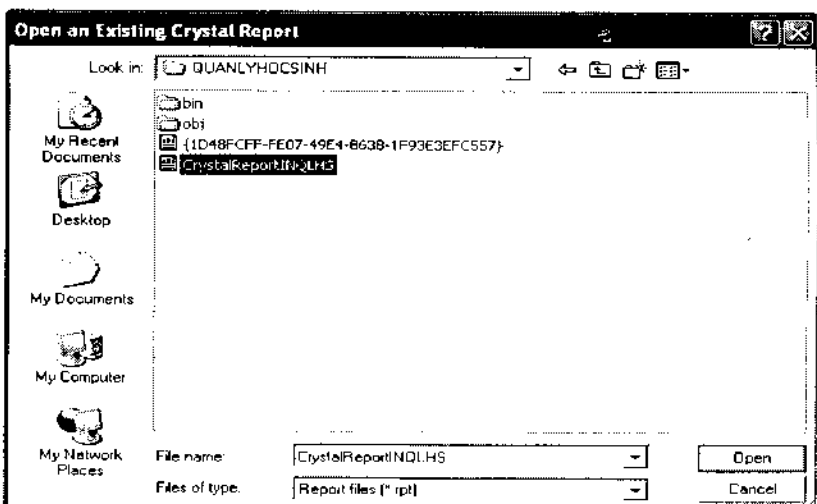
H457

◆ Click *nút phải chuột* trên điều khiển **CrystalReportViewer**, chọn **Properties**, xác lập thuộc tính **Font** là VNI-Times, 12, Bold, xác lập thuộc tính **ReportSource** là đường dẫn chứa tập tin **CrystalReportINQLHS.rpt** (click đầu mũi tên của khung bên phải, click nút **Browse** để tìm đến tập tin **.rpt** ở trên, tập tin này sẽ nằm trong thư mục **QUANLYHOCSINH**)



H458

◆ Click chọn tên **CrystalReportINQLHS**



H459

Click nút Open, bạn thấy thông tin về đường dẫn đến tập tin này hiện ra trong khung bên phải của thuộc tính ReportSource.

Click chọn Form1.cs trong cửa sổ Solution Explorer

Click chọn View Code

Sửa lại code thay vì

```
static void Main()
```

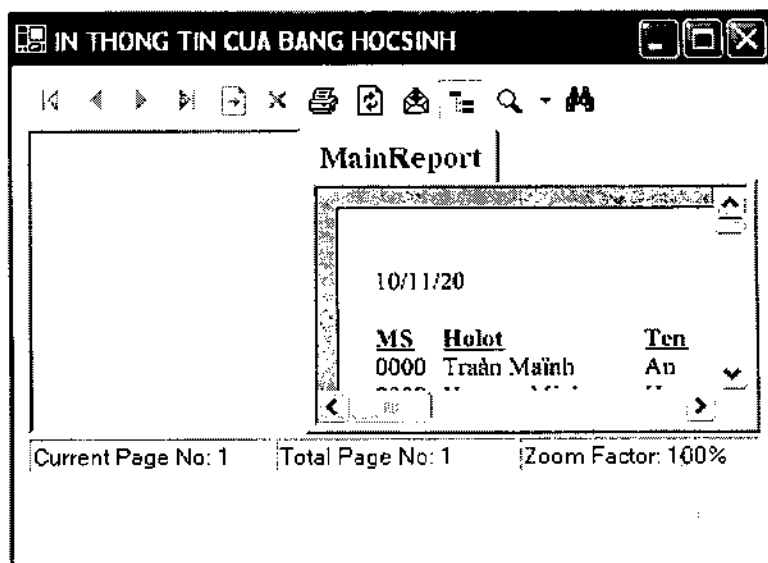
```
{  
    Application.Run(new QLHSDIEMTHILAI());  
}
```

Sửa lại thành:

```
static void Main()
```

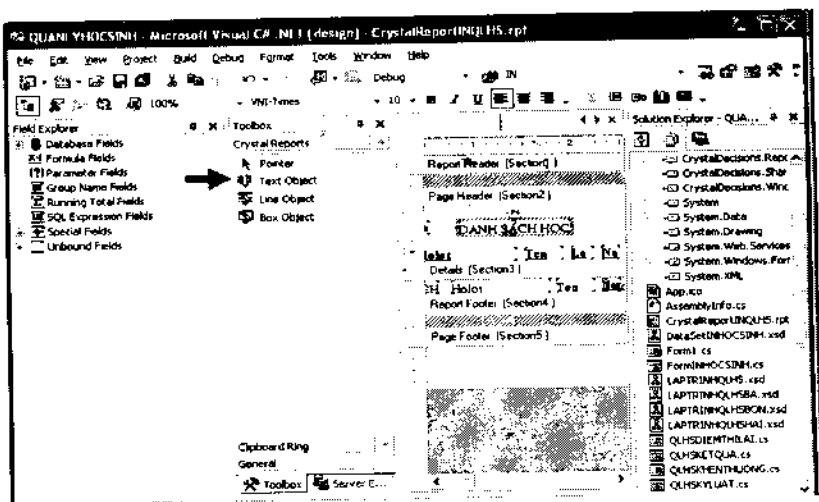
```
{  
    Application.Run(new FormINHOCSINH());  
}
```

Nhấn phím F5 chạy thử chương trình, bạn thấy hình sau hiện ra



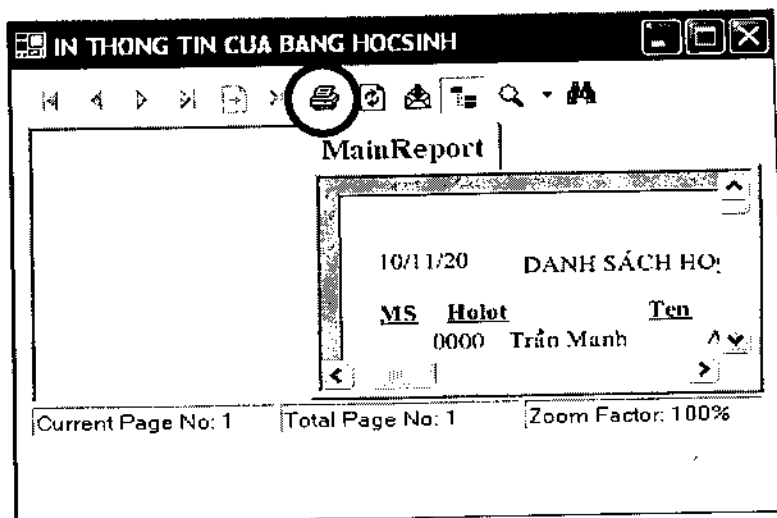
H460

◆ Bạn click nút phải chuột trên tập tin **CrystalReportINQLHS.rpt** trong cửa sổ Solution Explorer, chọn Open, nó hiện trên màn hình như hình sau:



H461

Chọn điều khiển **Text Object** trong hộp công cụ, vẽ và nhập "**DANH SÁCH HỌC SINH**" như hình trên, click chọn khung Text Object, chọn Font VNI-Times, size 10. Nhấn phím **Ctrl** đồng thời click chọn tên các vùng MSHS, Holot, Ten ... trong dòng của mục **Details (Section3)**, chọn Font VNI-Times, 10, bold. Nhấn phím **F5** chạy lại chương trình, bạn sẽ có hình như sau:



H462

Để in ra, bạn click vào hình chiếc máy in trên thanh công cụ.

Ghi chú:

Chúng tôi chỉ giới thiệu cách in một bảng, các bạn có thể in thông tin của hai hay nhiều bảng, nó sẽ thể hiện khi bạn chọn các bảng, chọn các vùng của bảng... trong phần thiết kế tập tin .rpt, các bạn thử tìm tòi và thiết kế thử, không có gì là khó cả. Và chữ trong các vùng bây giờ là tiếng Việt theo font VNI-Times, size 10.

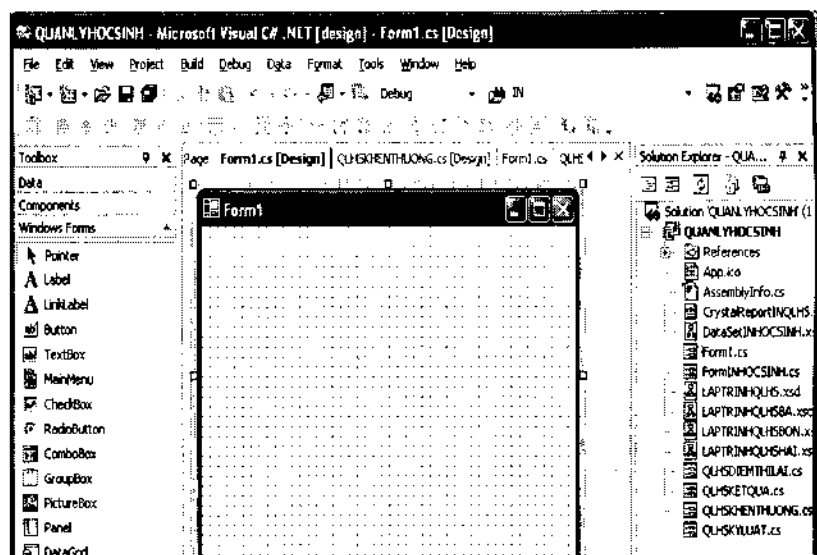
Bài 4

Đóng gói và phân phối ứng dụng trong Visual C#

Khi đã hoàn tất chương trình viết trên Visual C# .NET, để phân phối ứng dụng đến tay người sử dụng hoặc tải lên Internet để mọi người cùng dùng, hoặc đóng gói thành một sản phẩm hoàn chỉnh. Visual C# .NET cung cấp cho ta nhiều cách để đóng gói, trong bài này, chúng tôi hướng dẫn bạn cách đóng gói dựa vào Setup Wizard.

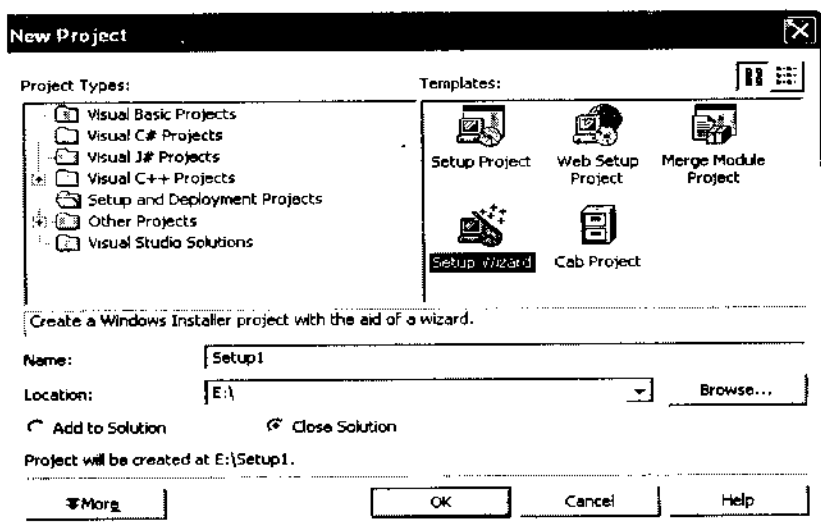
Muốn đóng gói chương trình nào, bạn mở chương trình đó trong Visual Studio .NET, trong bài này chúng ta thử đóng gói chương trình có tên QUANLYHOCSINH vừa hoàn thành ở trên

☉ Mở Project có tên QUANLYHOCSINH



H463

☉ Vào menu **File**, chọn **New**, chọn **Project** hộp thoại hiện ra như hình sau:



H464

Bước này ta sẽ tạo dự án thứ hai cho Solution, dự án này sẽ tự động tạo bộ đóng gói cho chương trình **QUANLYHOCSINH**.

☉ Chọn **Setup and Deployment Project** trong khung **Project Types** bên trái của hộp thoại.

☉ Chọn **Setup Wizard** trong khung **Templates** bên phải của hộp thoại.
Ghi chú:

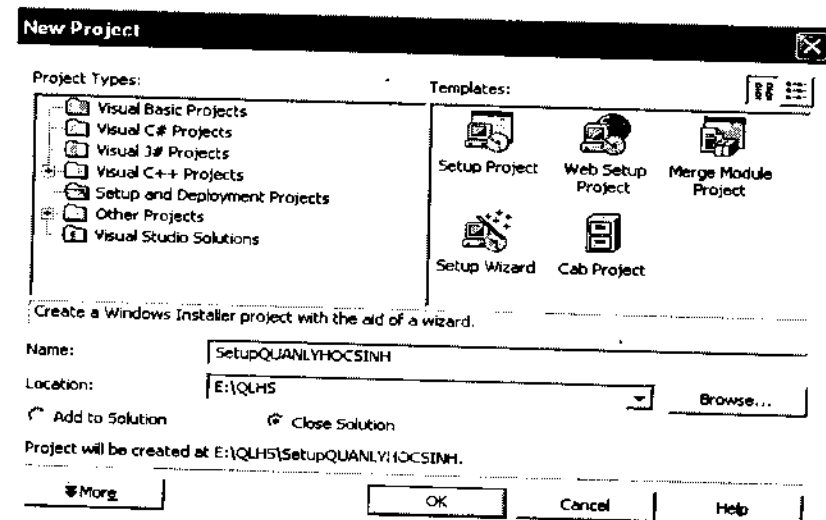
Khi chọn xong, nhìn trong khung **Name**, bạn sẽ thấy **Visual C# .NET** tự động điền tên **Setup1**.

Trong khung **Templates** có 5 mẫu cho bạn chọn để đóng gói. Nếu chọn **Setup Project** sẽ tạo bộ đóng gói cài đặt bởi **Windows Installer**. Nếu chọn **Web Setup Project** sẽ tạo trình **Setup** cho phép **Windows Installer** cài đặt chương trình từ một **Web Server**. Nếu chọn **Merge Module Project** được thiết kế cho mục đích đóng gói chung các dự án **Visual C#** khác nhau. Nếu dùng **Cab Project** cho phép đóng gói chương trình cài đặt thành một hoặc nhiều file nén **.cab**, dùng mẫu này nếu bạn muốn người dùng có thể download bộ cài đặt của bạn từ Internet. Mẫu cuối cùng **Setup Wizard** có lẽ là thông dụng nhất, mẫu này sẽ đóng gói chương trình của bạn thông

qua một số câu hỏi và yêu cầu bạn thực hiện một số bước cấu hình đơn giản.

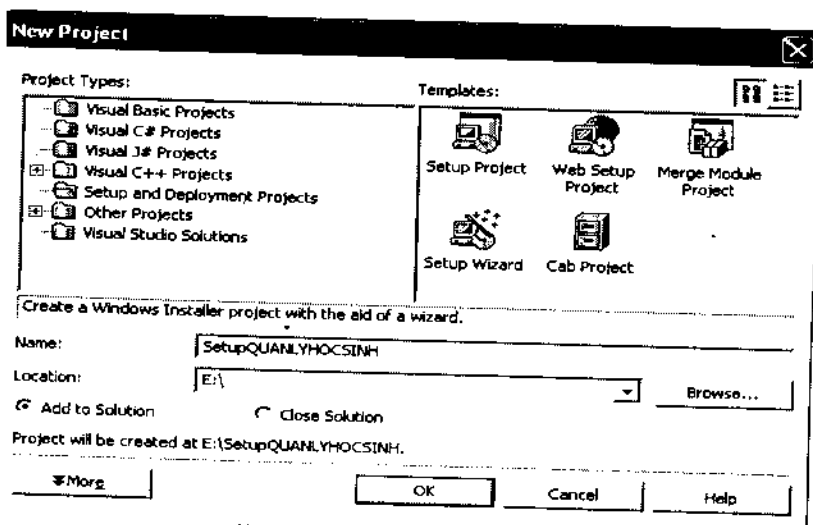
◎ Bạn gõ **SetupQUANLYHOCSINH** vào khung **Name**

◎ Gõ **E:\QLHS** trong khung **Location**



H465 Nhập thông tin vào ô Name và ô Location

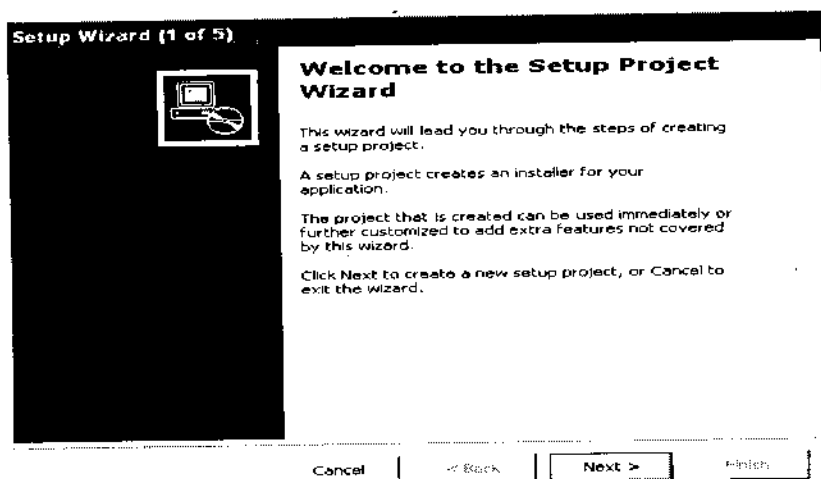
◎ Click nút chọn có tên **Add to Solution**, bạn sẽ có hình như sau:



H466 Khi chọn mục Add to Solution

Khi chọn xong, bạn thấy ở ô **Location** sẽ thay đổi chỉ còn E:\ bạn cần lưu ý nút chọn này, nó có ý nghĩa quan trọng, nếu bạn không chọn nút này Visual Studio sẽ đóng dự án hiện hành.

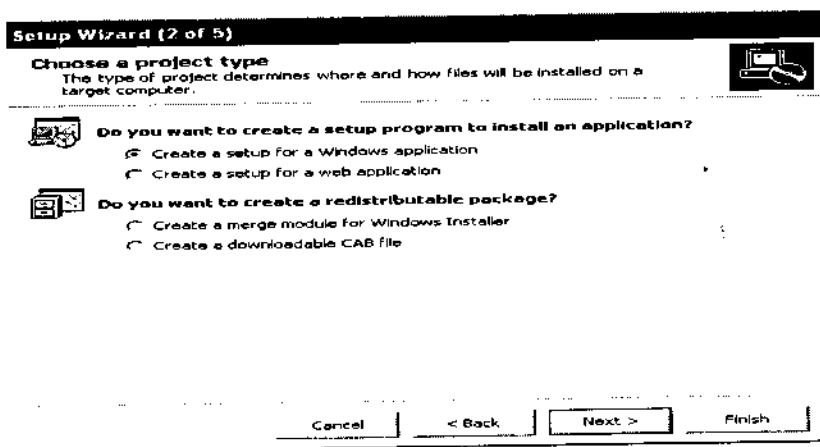
⊙ Click nút **OK** trên hộp thoại. Trình **Setup Wizard** sẽ hiện lần lượt năm hộp thoại để chúng ta xác lập. Đầu tiên chúng ta sẽ có hộp thoại như sau:



H467 Hộp thoại thứ nhất

Đây là hộp thoại giới thiệu

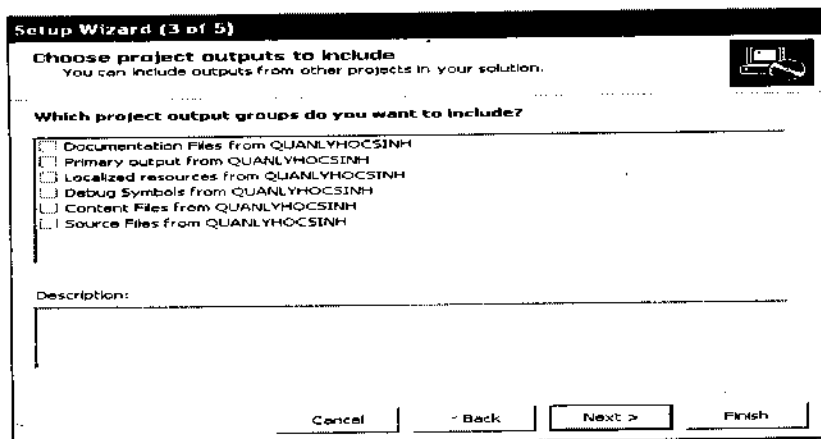
⊙ Click nút **Next** để tiếp qua hộp thoại thứ hai



H468 Hộp thoại thứ hai

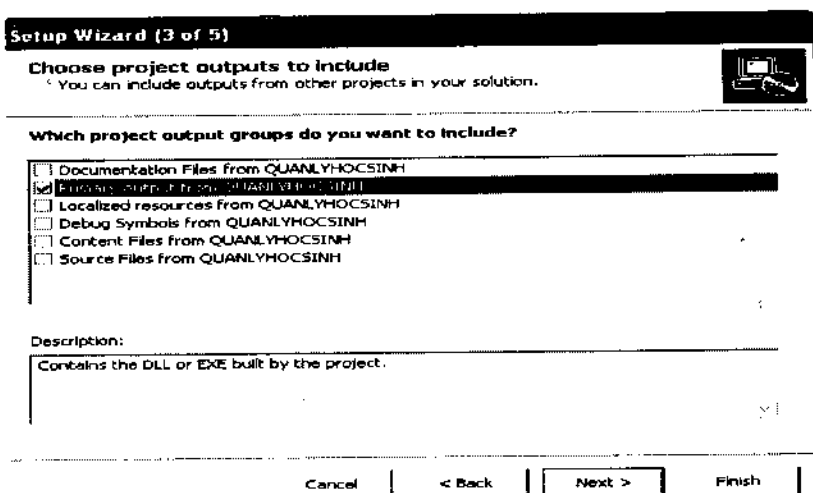
Hộp thoại này cho phép bạn kiểm soát cách đóng gói và phân phối ứng dụng, các tùy chọn này giống với các mẫu trong khung Template mà bạn đã thấy trong hộp thoại New Project trước đây.

- Click chọn mục **Create a setup for a Windows application**
- Click nút **Next** để qua hộp thoại thứ ba



H469

● Bạn chọn mục **Primary output from QUANLYHOCSINH** thường sẽ được chọn để đưa các file .exe hoặc .dll của mình vào trình đóng gói.



H470

⊙ Click nút Next để qua hộp thoại thứ tư

Setup Wizard (4 of 5)

Choose files to include

You can add files such as ReadMe files or HTML pages to the setup.



Which additional files do you want to include?

	Add...
	Remove

Cancel

< Back

Next >

Finish

H471 Hộp thoại thứ tư

Trong hộp thoại này, các bạn có thể đưa thêm các file dữ liệu, các file để hướng dẫn bổ sung cho chương trình lúc thực thi, các tài liệu quảng cáo... Trong bài này chúng ta không cần phải đưa vào tập tin nào.

Ghi chú:

.NET Framework là thành phần chính giúp cho chương trình có thể chạy, nhưng chúng ta không được nhúng kèm theo .NET Framework trong bộ Setup. Các bạn phải cài đặt riêng độc lập với chương trình

⊙ Click nút **Next** để qua hộp thoại cuối cùng, hộp thoại thứ năm

Setup Wizard (5 of 5)

Create Project

The wizard will now create a project based on your choices.



Summary:

Project type: Create a setup for a Windows application

Project groups to include:

Primary output from QUANLYHOC SINH

Additional files: (none)

Project Directory: E:\SetupQUANLYHOC SINH\SetupQUANLYHOC SINH.vdproj

Cancel

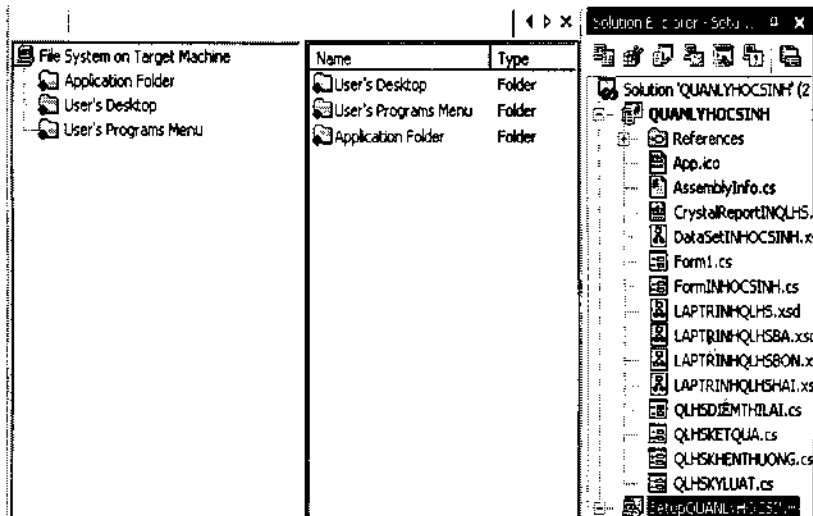
< Back

Next >

Finish

H472 Hộp thoại thứ năm

© Click nút **Finish** để kết thúc quá trình **Setup Wizard**.



H473



Thiết lập các cấu hình

© Vào menu **Build**, chọn **Configuration Manager**, hộp thoại sau hiện

ra

Configuration Manager

Active Solution Configuration:

Debug

Project Contexts (check the project configurations to build or deploy):

Project	Configuration	Platform	Build
QUANLYHOCSINH	Debug	.NET	☒
SetupQUANLYHOCSINH	Debug		☐

Close

Help

H474

Hộp thoại hiển thị chế độ biên dịch hiện hành cho các dự án đang có trong Solution. Chúng ta đang có hai dự án là QUANLYHOCSINH và SetupQUANLYHOCSINH. Cả hai dự án đang ở chế độ biên dịch Debug, đây là chế độ biên dịch kèm theo các thông tin gỡ lỗi. Tuy nhiên chúng ta nên chọn cách biên dịch tối ưu hơn đó là biên dịch **Release**.

© Click chọn **đầu mũi tên quay xuống** trong khung **Active Solution Configuration**.

Configuration Manager

Active Solution Configuration:

Debug

Debug

Release

<New...>

<Edit...>

QUANLYHOCSINH

SetupQUANLYHOCSINH

to build or deploy):

Platform	Build
.NET	☒
	☐

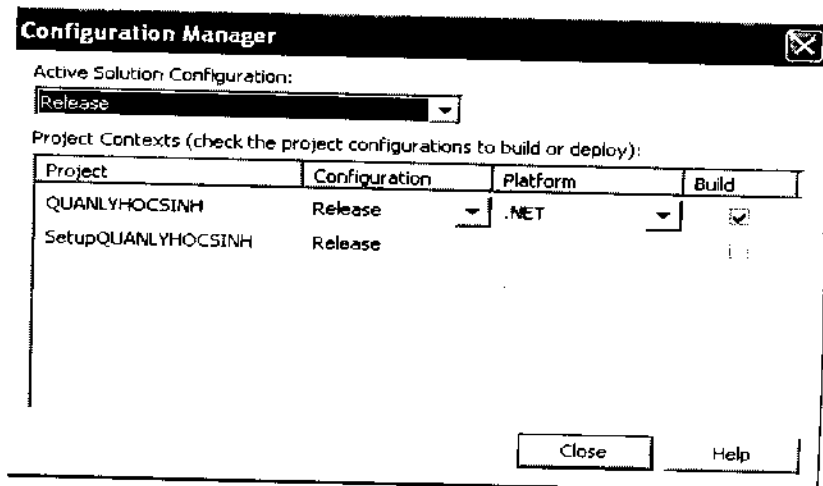
Close

Help

H475

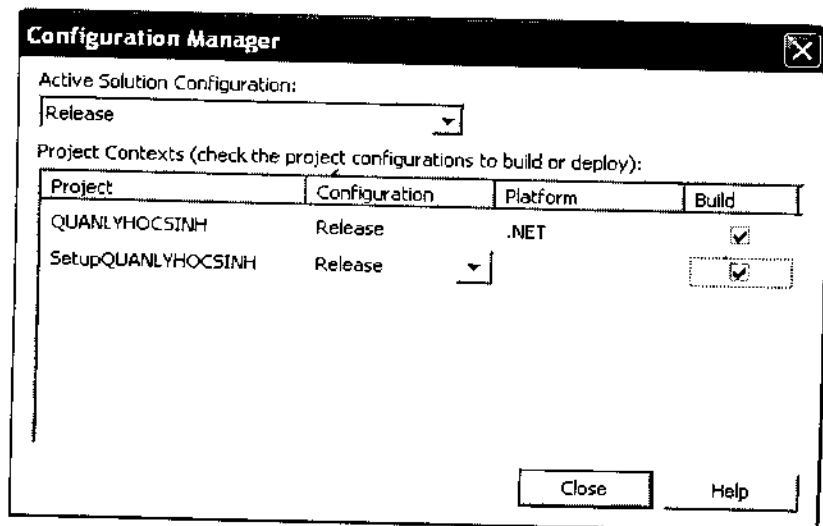
© Click chọn **Release** trong khung hiện ra.

Khi chọn mục này, các bạn thấy trong cột **Configuration** của hai dự án **QUANLYHOCSINH** và **SetupQUANLYHOCSINH** cũng đổi từ **Debug** sang **Release**. Nếu không thấy đổi, bạn click vào đầu mũi tên quay xuống của những mục này để đổi sang **Release**.



H476 Đã đổi sang Release

© Click chọn vào cột **Build** cho cả hai dự án



H477 Đã chọn mục Build cho cả hai dự án

© Click nút **Close** để đóng hộp thoại lại

Tiếp theo các bạn phải tiến hành biên dịch và đóng gói dự án, bạn tiến hành như sau

➤ **Biên dịch và đóng gói dự án – Kiểm tra việc cài đặt**

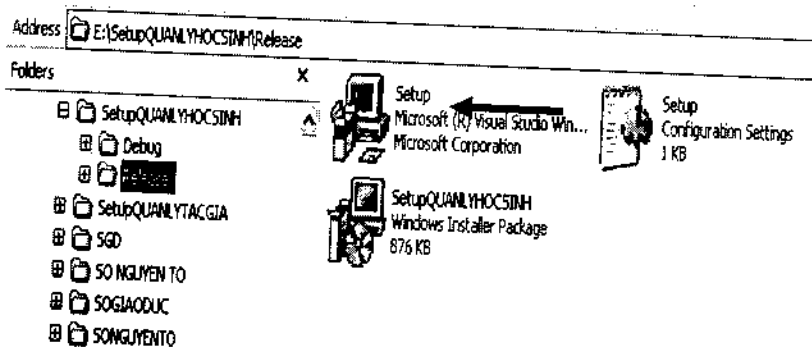
⊙ Vào menu Build, chọn Build Solution. Visual Studio sẽ biên dịch cả hai dự án QUANLYSINHVIEN và SetupQUANLYGIAOVIEN. Quá trình biên dịch đóng gói diễn ra nếu chú ý phía dưới Form, bạn sẽ thấy thanh tiến trình xuất hiện như hình dưới (nó xuất hiện phía dưới Form)



H478 Thanh tiến trình biên dịch

Nếu biên dịch thành công, bạn thấy câu "Build Succeeded" xuất hiện phía tận cùng bên trái nằm cùng với thanh tiến trình.

- ⊙ Click nút phải chuột trên menu Start, chọn Explorer
- ⊙ Vào thư mục E:\SetupQUANLYHOCSINH\Release



H479 Chọn E:\SetupQUANLYHOCSINH\Release

⊙ Nhấp đúp vào tập tin **Setup** được chọn trong hình hoặc click nút phải chuột trên tập tin này và chọn **Open** trong menu hiện ra. Bạn sẽ thấy hộp thoại như hình dưới

Ghi chú:

Nếu muốn có bản cài đặt trên đĩa CD, bạn chép toàn bộ nội dung thư mục Release này sang đĩa CD.

Welcome to the SetupQUANLYHOCSINH Setup Wizard



The installer will guide you through the steps required to install SetupQUANLYHOCSINH on your computer.

WARNING: This computer program is protected by copyright law and international treaties. Unauthorized duplication or distribution of this program, or any portion of it, may result in severe civil or criminal penalties, and will be prosecuted to the maximum extent possible under the law.

Cancel

< Back

Next >

H480 Hộp thoại giới thiệu

☉ Click nút **Next** để tiếp tục, hộp thoại chọn thư mục để yêu cầu xác định thư mục cài đặt cho chương trình cùng một vài tùy chọn cần thiết như hình sau:

Select Installation Folder



The installer will install SetupQUANLYHOCSINH to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder:

[C:\Program Files\129 Cách Mạng Tháng 8 _ Q-3\SetupQUANLYHOCSINH

Browse...

Disk Cest...

Install SetupQUANLYHOCSINH for yourself, or for anyone who uses this computer:

☐ Everyone

☒ Just me

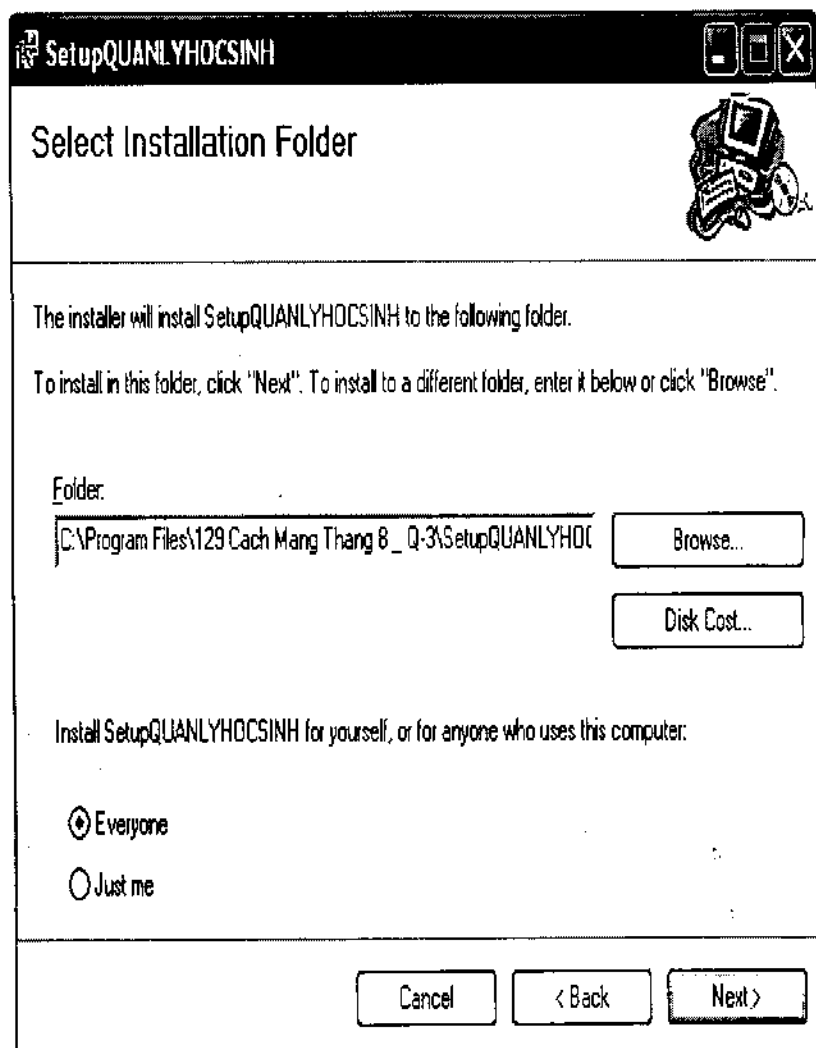
Cancel

< Back

Next >

Trong bài này chúng tôi chọn thư mục C:\Program Files\129 CachMangThang8_Q3\SetupQUANLYHOCSINH (nếu thích bạn có thể nhập trực tiếp thư mục khác trong khung này, hoặc nhấn nút Browse để chọn)

☉ Click chọn mục **Everyone**



SetupQUANLYHOCSINH

Select Installation Folder

The installer will install SetupQUANLYHOCSINH to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder:

C:\Program Files\129 Cach Mang Thang 8 _ Q-3\SetupQUANLYHOCSINH

Browse...

Disk Cost...

Install SetupQUANLYHOCSINH for yourself, or for anyone who uses this computer:

☒ Everyone

☐ Just me

Cancel < Back Next >

H482 Chọn mục Everyone

☉ Click nút **Next** để bắt đầu cài đặt.

Confirm Installation



The installer is ready to install SetupQUANLYHOCSINH on your computer.

Click "Next" to start the installation.

Cancel

< Back

Next >

H483

☉ Click nút **Next**

Installation Complete



SetupQUANLYHOCSINH has been successfully installed.

Click "Close" to exit.

Cancel

< Back

Close

H484

Please use Windows Update to check for any critical updates to the .NET Framework.

Quá trình cài đặt sẽ chép các tập tin vào thư mục dành cho chương trình. Sau khi hoàn tất, trình Setup sẽ đăng ký ứng dụng QUANLYHOCSINH với Windows Registry.

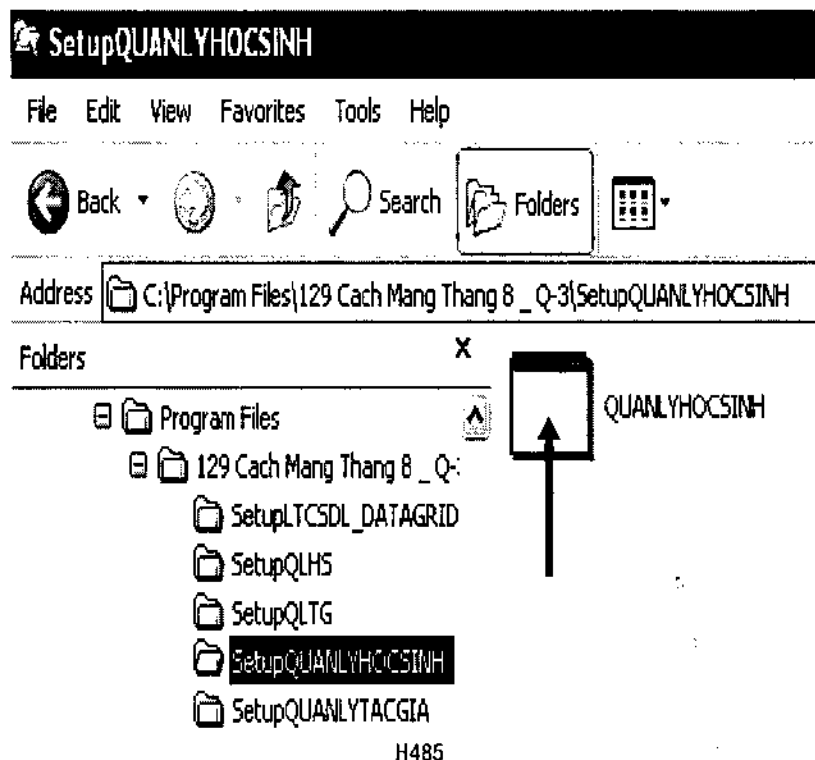
Ghi chú:

Sau này muốn xóa, các bạn vào Control Panel, chọn Add/Remove Program để gỡ bỏ

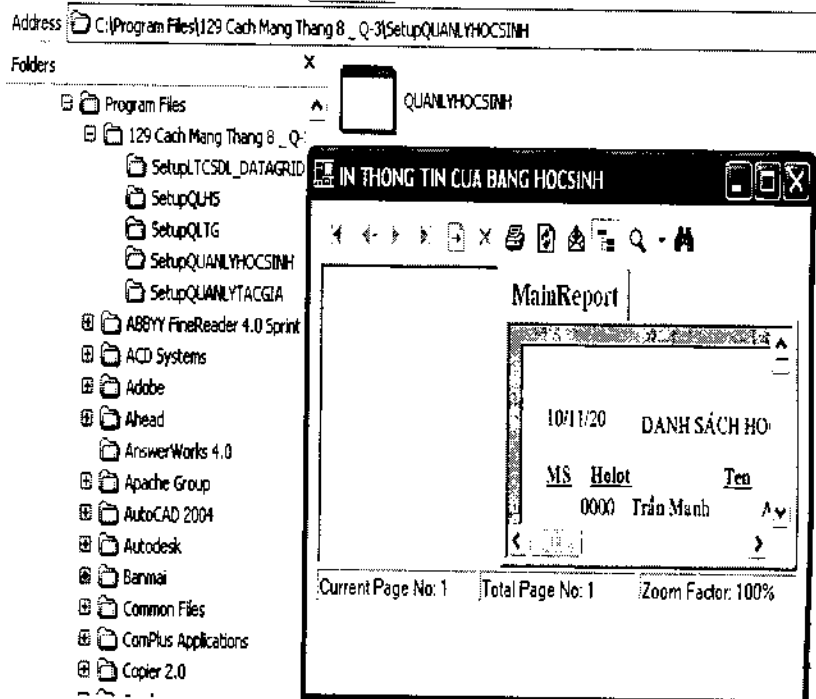
☉ Click nút **Close** khi quá trình cài đặt đã hoàn tất.

➤ **Chạy thử chương trình vừa cài đặt**

Click nút phải chuột trên menu Start, chọn Explorer. Chọn thư mục C:\Program Files\129CachMangThang8_Q3\SetupQUANLYHOCSINH. Bạn thấy tập tin QUANLYHOCSINH như hình sau:



☉ Click nút phải chuột trên tập tin này, chọn **Open** trong menu hiện ra, bạn sẽ thấy chương trình QUANLYHOCSINH hiện ra không cần mở Visual Studio .NET



H486

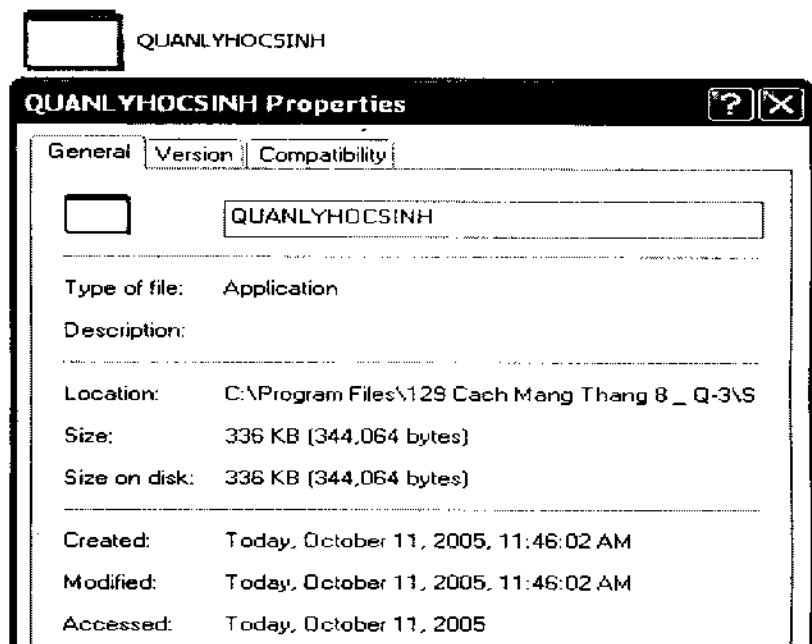
☐ Lúc này các bạn có thể thao tác giống như khi chạy chương trình.

Ghi chú:

Vì chương trình QUANLYHOCSINH đang chạy mặc nhiên với tập tin FormINHOCSINH nên bạn thấy biểu mẫu chương trình hiện ra như trên.

Trên mỗi máy chạy chương trình .NET cần phải có thư viện hay bộ khung .NET Framework cài đặt sẵn. Microsoft cung cấp cho bạn bộ cài đặt .NET runtime (file Dotnetfx.exe) trên đĩa CD Windows Component Update. Bạn cũng có thể lấy trên mạng từ Site Microsoft (kích thước khoảng 20Mb), sau khi cài đặt, thư viện này chiếm khoảng 30Mb trên đĩa, do đó chúng ta không còn dùng được đĩa mềm để sao chép cài đặt. Tuy nhiên sắp tới Microsoft cam kết sẽ kèm theo bộ .NET runtime trong tất cả các phiên bản của Windows sau này. Nếu máy tính bạn định cài chương trình đã có sẵn .NET Framework runtime, bạn chỉ cần copy chương trình sang máy mới và chạy. Tuy nhiên để có một bộ cài đặt hoàn chỉnh, trình Setup Wizard thường nhúng luôn thư viện .NET runtime kèm theo chương trình để người dùng khỏi cần download hay chép từ nơi khác. Vì vậy không có gì ngạc nhiên nếu chương trình của bạn biên dịch chỉ có khoảng

336Kb nhưng khi đóng gói cài đặt có thể lên đến 25Mb, đó chẳng qua là thư viện .NET runtime kèm theo.



H487

Các bạn có thể tìm đọc bộ sách .NET của chúng tôi

- ◆ Tự học lập trình Visual Basic .NET
- ◆ Tự học lập trình cơ sở dữ liệu Access 2003 và Visual Basic .NET
- ◆ Tự học lập trình cơ sở dữ liệu SQL Server 2000 và Visual Basic .NET
- ◆ Tự học lập trình cơ sở dữ liệu Visual C++ .NET

Mục lục

Phần 1: ◆ Cách vào MS Deverlopment Environment Tìm hiểu cửa sổ thiết kế ◆ MS Deverlopment Environment	1
Phần 2: ◆ Dùng Visual C# Project và Console Application Bài 1: Cho xuất hiện một câu chào	5
Bài 2: Cấu trúc IF	13
Bài 3: Cấu trúc IF có ELSE	22
Bài 4: Cấu trúc IF ELSE lồng nhau	25
Bài 5: Cấu trúc chọn SWITCH CASE	30
Bài 6: Cấu trúc lặp FOR	46
Bài 7: Cấu trúc lặp WHILE/DO WHILE	51
Bài 8: Mảng trong C#	56
Bài 9: Lớp và đối tượng trong C#	68
Bài 10: Kiểu dữ liệu của người lập trình	94
Bài 11: Các phương thức trùng tên (chống phương thức)	99
Bài 12: Truyền tham số	103
Bài 13: Kế thừa lớp đối tượng	115
Bài 14: Tương ứng bội (Polymorphism)	121
Phần 3: ◆ Lập trình Visual C# dùng cơ sở dữ liệu Access 2003. Vài nét cơ bản về Access.	129
Bài 1: Theo dõi kết quả SEAGAMES23	153
Bài 2: Theo dõi kết quả SEAGAMES23 dùng Data Form Wizard	180
Bài 3: Quản lý sinh viên thiết kế bằng Data Form Wizard	200

Phần 4: ◆ Lập trình Visual C# dùng cơ sở dữ liệu SQL SERVER 2000 Cách cài đặt SQL SERVER 2000	227
Tạo cơ sở dữ liệu, bảng và mối liên hệ giữa các bảng trong SQL SERVER	238
Bài 1: Chương trình quản lý học sinh dùng SQL SERVER 2000 và Visual C#	285
Bài 2: Thiết kế chương trình quản lý học sinh bằng Data Form Wizard	326
Bài 3: In nội dung các bảng bằng Crystal Report	363
Bài 4: Đóng gói và phân phối ứng dụng bằng Visual C#	377

**TỰ HỌC LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
VÀ LẬP TRÌNH CƠ SỞ DỮ LIỆU C#
ĐẬU QUANG TUẤN**

Chịu trách nhiệm xuất bản :
HOÀNG CHÍ DŨNG

Biên tập : **NGUYỄN BẢO THÙY**
Sửa bản in : **NGUYỄN ĐẠI THẮNG**
Trình bày : **ĐẬU MINH NHẬT**
 ĐẬU THUY QUỲNH TRÂM
Bìa : **THIÊN Ý**

Thực hiện liên doanh : **ĐẬU QUANG TUẤN**

In 1.000 bản, khổ 14,5 x 20,5cm. Tại **Xí nghiệp IN SỐ 5**, Quận 5, Tp.HCM. Số xuất bản: 43 – 1/230/XB-QLXB cấp ngày 03.03.2005.
In xong và nộp lưu chiểu tháng 12-2005.

SÁCH PHÁT HÀNH NĂM 2006



Xin chân thành cảm ơn quý độc giả đã đón đọc những tập sách của chúng tôi.

Tất cả những tập sách trên được phát hành
ĐT : 8984154 - DD: 0908.653 560
E-mail: augustinodauquangquan@yahoo.com

Giá : 50.000đ