

Hướng Dẫn Thực Hành

Lập Trình Windows Căn Bản

Khối: Cao Đẳng nghề và Trung Cấp

Năm 2011



Hướng dẫn:

- Bài tập thực hành được chia làm nhiều Module
- Mỗi Module được thiết kế cho thời lượng là 3 tiết thực hành tại lớp với sự hướng dẫn của giảng viên.
- Tùy theo số tiết phân bổ, mỗi tuần học có thể thực hiện nhiều Module.
- Sinh viên phải làm tất cả các bài tập trong các Module ở tuần tương ứng. Những sinh viên chưa hoàn tất phần bài tập tại lớp có trách nhiệm tự làm tiếp tục ở nhà.
- Các bài có dấu (*) là các bài tập nâng cao dành cho sinh viên khá giỏi.

Module 1

Nội dung kiến thức thực hành:

- + Làm quen với ngôn ngữ C#
- + Tạo ứng dụng dạng Console
- + Một số thuộc tính, phương thức qua Console.

Bài 1:

Mục đích:

Sử dụng các properties và methods của các đối tượng trong ứng dụng Console.

Yêu cầu:

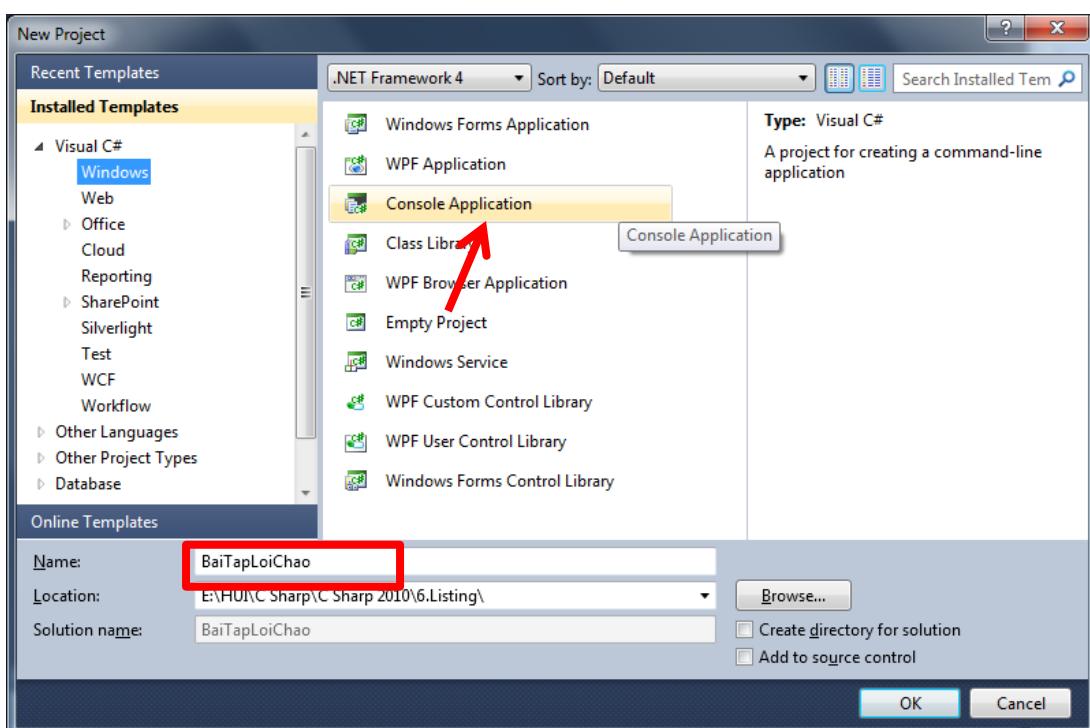
Viết ứng dụng dạng Console xuất lời chào.

Yêu cầu xuất ra lời chào sau:

**Welcome to
C# 2010 Programming!
Programming by your name**

Hướng dẫn:

- + Mở Microsoft Visual Studio 2010



Project tạo ra một file mặc định là: Program.cs có chứa một hàm Main()

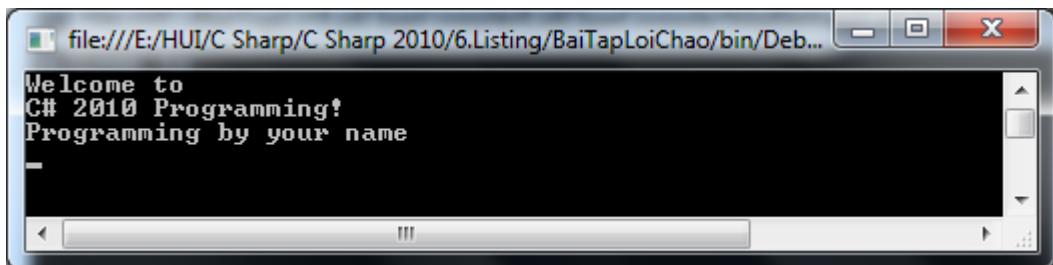
```
namespace BaiTapLoiChao
{
    class Program
    {
        static void Main(string[] args)
        {
            //Viết code trong đây
        }
    }
}
```

+ Bạn sửa lại hàm Main() thành như sau.

```
namespace BaiTapLoiChao
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Welcome to ");
            Console.WriteLine("C# 2010 Programming!");
            Console.WriteLine("Programming by your name");
            Console.ReadLine();

        }
    }
}
```

+ Chạy chương trình bằng cách nhấn phím **F5** được kết quả mong muốn sau:



Bài 2:

Mục đích:

Sử dụng các Properties và Methods của các đối tượng trong ứng dụng Console
Kết hợp định dạng xuất dữ liệu.

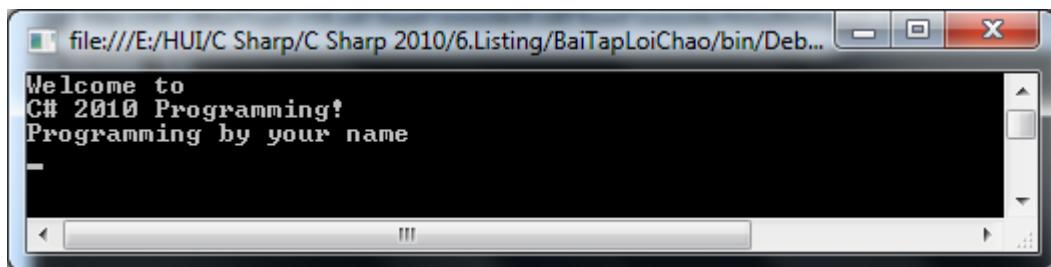
Yêu cầu:

+ Rào các statement trong hàm Main của bài 1 chuyển sang dạng comment.

+ Viết lại hàm Main như sau:

```
static void Main(string[] args)
{
    Console.WriteLine("{0}\n{1}", "Welcome to ", "C# 2010 Programming!");
    Console.WriteLine("Programming by your name");
    Console.ReadLine();
}
```

+ Chạy chương trình bằng cách nhấn phím F5 được kết quả mong muốn không đổi như bài tập 1:



Bài 3:

Mục đích:

Sử dụng một số hàm toán học và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.

Yêu cầu:

+ Tạo Project mới với tên là **ThaoTacSo**:

Khi chạy chương trình xuất ra yêu cầu nhập vào hai số từ bàn phím xuất ra kết quả tổng của hai số vừa nhập.

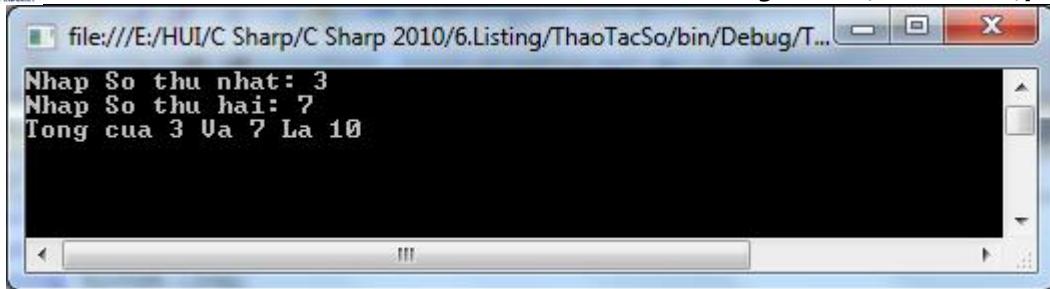
Xuất ra màn hình yêu cầu có dạng:

Nhap So Thu Nhat A: ??

Nhap So Thu Hai B: ??

Tong cua ?? Va ?? La: ??

Ví dụ xem hình :



Hướng dẫn:

Hàm Main() viết code như sau:

```
int nSo1, nSo2, nTong=0;
Console.Clear();
Console.Write("{0}", "Nhập Số thứ nhất: ");
// read first number from user
nSo1 = Convert.ToInt32(Console.ReadLine());
Console.Write("Nhập Số thứ hai: ");
// read second number from user
nSo2 = Convert.ToInt32(Console.ReadLine());
nTong = nSo1 + nSo2;
Console.WriteLine("Tổng của " + nSo1 + " Và " + nSo2 + " Là " + nTong);
Console.ReadLine();
```

- + Mở rộng làm các phép tính cơ bản (cộng, trừ, nhân, chia)

Bài 4:

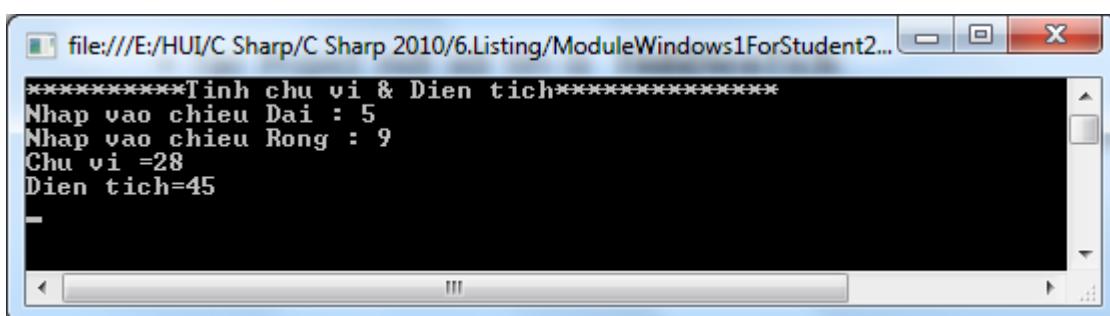
Mục đích:

Sử dụng một số hàm toán học và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.

Yêu cầu:

- + Tạo Project mới với tên là **TinhDienTich**:
- + Khi chạy chương trình xuất ra yêu cầu nhập vào hai cạnh của một tứ giác vuông xuất ra kết quả chu vi, diện tích....

Xem hình:



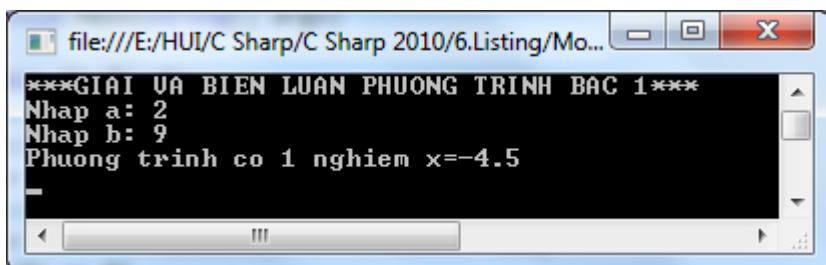
Bài 5:

Mục đích:

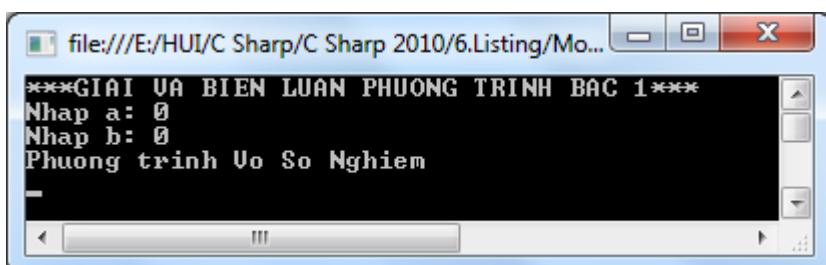
Sử dụng một số hàm toán học và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.

Yêu cầu:

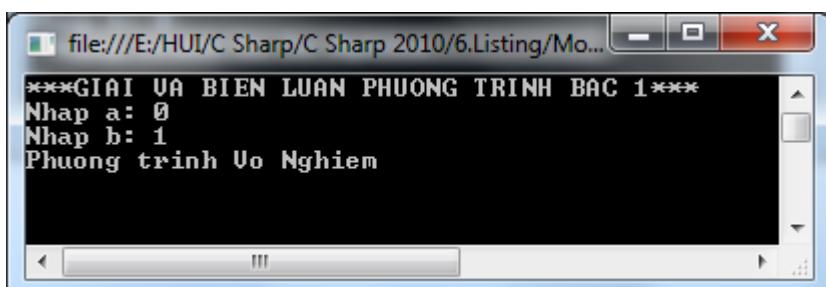
- + Tạo Project mới với tên là **PhuongTrinhBac1**:
- + Giải và biện luận phương trình bậc 1 : $ax+b=0$
- + Sinh viên chú ý rằng phương trình có 3 trường hợp: vô nghiệm, vô số nghiệm và có một nghiệm
- + xem hình để hiểu rõ yêu cầu



```
***GIAI VA BIEN LUAN PHUONG TRINH BAC 1***
Nhap a: 2
Nhap b: 9
Phuong trinh co 1 nghiem x=-4.5
```



```
***GIAI VA BIEN LUAN PHUONG TRINH BAC 1***
Nhap a: 0
Nhap b: 0
Phuong trinh Vo So Nghiem
```



```
***GIAI VA BIEN LUAN PHUONG TRINH BAC 1***
Nhap a: 0
Nhap b: 1
Phuong trinh Vo Nghiem
```

Bài 6:

Mục đích:

Sử dụng một số hàm toán học và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.(làm quen class **Math**)

Math.Pow(x,y) lấy x^y

Math.Sqrt(x) lấy căn bậc 2 của x

Math.Round(x,2) làm tròn x tới 2 đơn vị

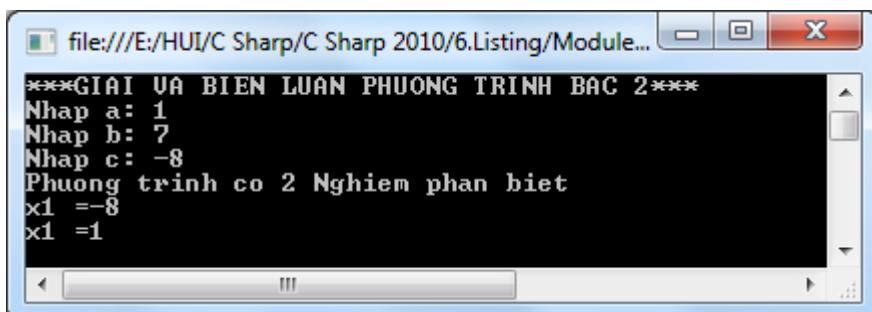
Yêu cầu:

+ Tạo Project mới với tên là **PhuongTrinhBac2**:

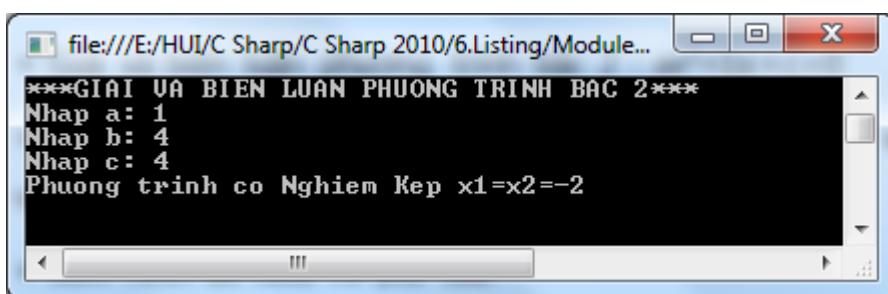
+ Giải và biện luận phương trình bậc 2 : $ax^2+bx+c=0$

+ Sinh viên chú ý rằng phương trình có 5 trường hợp: vô nghiệm, vô số nghiệm và có một nghiệm, có 2 nghiệm phân biệt, có 1 nghiệm kép

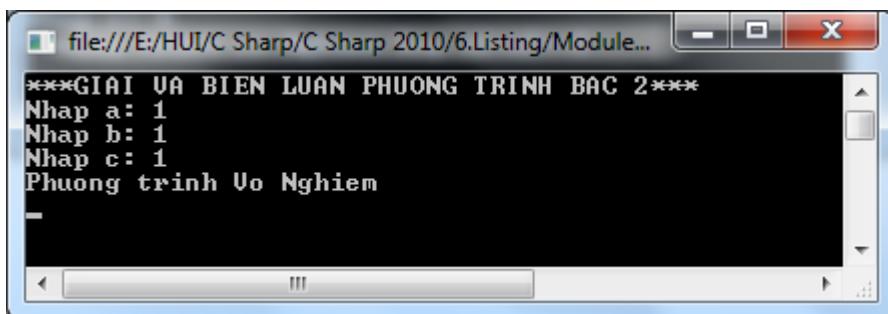
+ xem hình để hiểu rõ yêu cầu



```
***GIAI VA BIEN LUAN PHUONG TRINH BAC 2***
Nhap a: 1
Nhap b: 7
Nhap c: -8
Phuong trinh co 2 Nghiem phan biet
x1 = -8
x2 = 1
```



```
***GIAI VA BIEN LUAN PHUONG TRINH BAC 2***
Nhap a: 1
Nhap b: 4
Nhap c: 4
Phuong trinh co Nghiem Kep x1=x2=-2
```



```
***GIAI VA BIEN LUAN PHUONG TRINH BAC 2***
Nhap a: 1
Nhap b: 1
Nhap c: 1
Phuong trinh Vo Nghiem
```

Bài 7:

Mục đích:

Sử dụng một số hàm lượng giác và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.(tiếp tục làm quen với class **Math**)

Math.PI lấy số PI

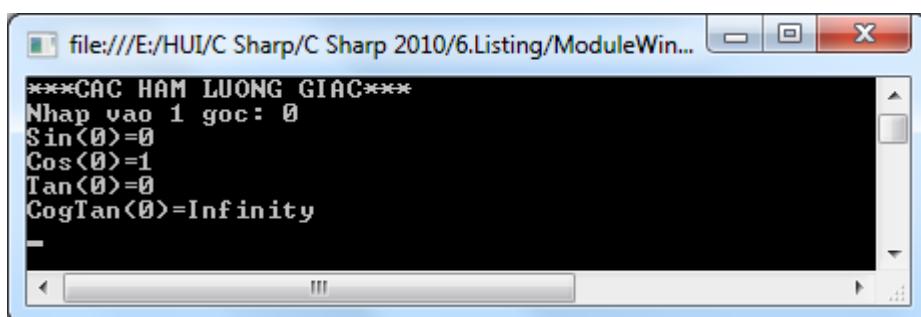
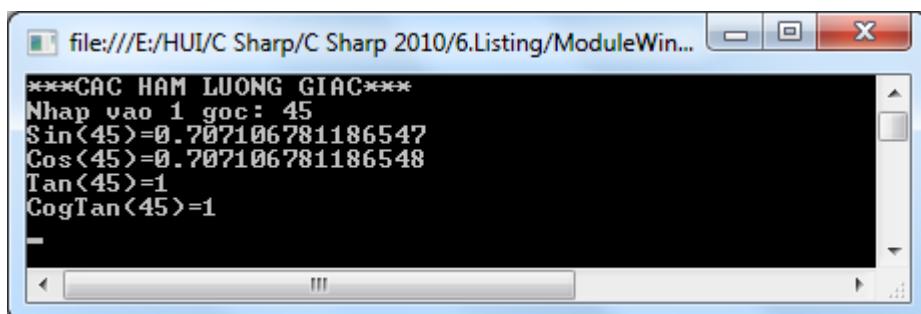
Math.Sin(a) tính sin, với a là radian

Math.Cos(a) tính Cos, với a là radian

Math.Tan(a) tính Tan, với a là radian

Yêu cầu:

- + Tạo Project mới với tên là **CacHamLuongGiac**:
- + Nhập 1 góc x. Tính và xuất sinx, cosx, tgx, cotgx.
- + Sinh viên chú ý rằng hàm lượng giác truyền vào là 1 số Radian, do đó các em phải đổi góc x ra radian, công thức đổi như sau $a=x*\pi/180$
- + Xem hình để hiểu rõ yêu cầu



Gợi ý tính hàm sin:

```
double fx;
Console.WriteLine("Nhập vào 1 góc: ");
fx = Convert.ToDouble(Console.ReadLine());
double rad = fx * Math.PI / 180;
Console.WriteLine("Sin({0})={1}", fx, Math.Sin(rad));
```

Bài 8:

Mục đích:

Sử dụng một số hàm lượng giác và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.(tiếp tục làm quen với class **Math**)

Yêu cầu:

+ Tạo Project mới với tên là **ChuViDienTichTamGiac**:

+ Nhập cạnh $a > 0$, $b > 0$, $c > 0$. Nếu a , b , c tạo thành tam giác thì hãy tính và xuất chu vi, diện tích hình tam giác. Ngược lại thông báo "Không tạo thành tam giác"

HD: $cv = a + b + c$, $p = cv / 2$ và $dt = \sqrt{p * (p - a) * (p - b) * (p - c)}$

Bài 9(*):

Mục đích:

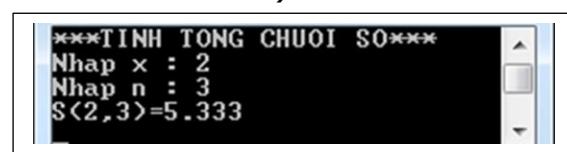
Sử dụng hàm toán học và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.

Yêu cầu:

+ Tạo Project mới với tên là **TinhTongChuoiSo**:

+ Nhập x , n để tính tổng chuỗi số sau (làm tròn 3 chữ số):

— — —



Bài 10(*):

Mục đích:

Sử dụng hàm toán học và nhập dữ liệu từ bàn phím Kết hợp định dạng xuất dữ liệu.

Yêu cầu:

+ Tạo Project mới với tên là **SoGaSoCho**:

+ Tìm số gà và số chó, biết:

Vừa gà vừa chó
Bó lại cho tròn
Ba mươi sáu con
Một trăm chân chẵn

Gợi ý:

Với X là số gà, Y là số Chó

Module 2

Nội dung kiến thức thực hành:

- + Làm quen với ngôn ngữ C#
- + Tạo ứng dụng trên windows form.

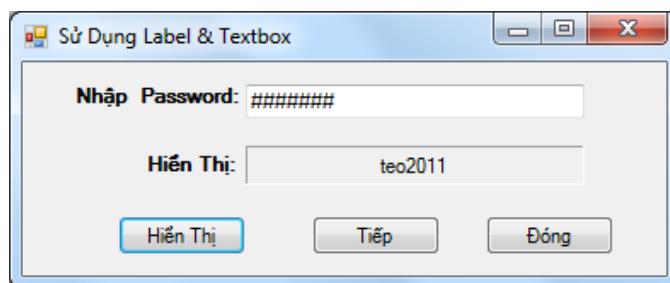
Mục đích:

Sử dụng các properties, methods và event của các đối tượng trong Windows Form.

Bài 1:

Yêu Cầu

Thiết kế form như hình bên dưới



Viết lệnh để chương trình hoạt động như sau:

- + Người sử dụng sẽ nhập Password vào textbox , sau đó nhấn nút “Hiển Thị” thì nội dung của password sẽ được hiển thị ra trên label bên dưới

Hướng dẫn:

Trong sự kiện Click của button "Hiển Thị", gõ vào câu lệnh sau

`IblHienThi.Text = txtPassWord.Text;`

- + Người sử dụng nhấn nút “Tiếp” để xoá nội dung textbox, label và đặt con trỏ vào textbox password để người sử dụng nhập nội dung mới thuận lợi.

Hướng dẫn:

Trong sự kiện Click của button "Tiếp", gõ vào đoạn lệnh sau

`IblHienThi.Text = "";`

`txtPassWord.Clear();`

`txtPassWord.Focus();`

- + Nhấn nút “Đóng” để kết thúc chương trình.

Hướng dẫn:

Trong sự kiện Click của button “Đóng”, gõ vào câu lệnh sau

```
this.Close();  
//hoặc Application.Exit();
```

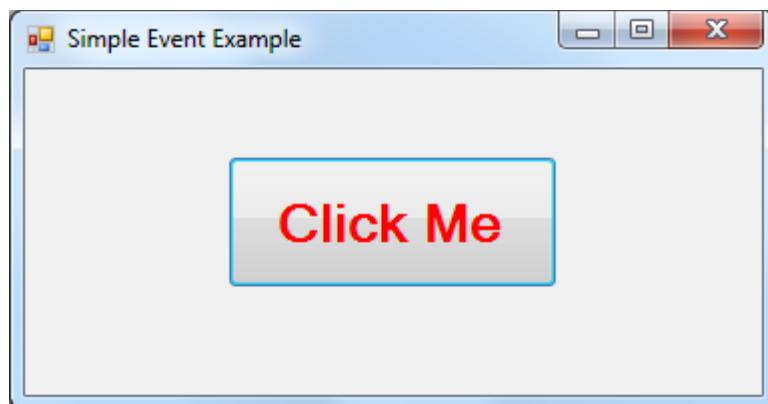
Bài 2:

Mục Đích:

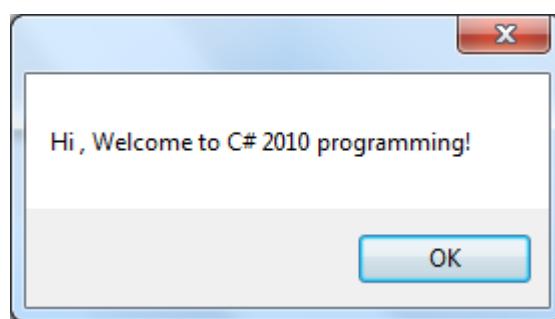
Sử dụng các sự kiện (event) cơ bản trên Form

Yêu Cầu

Thiết kế form chỉ chứa 1 button (xem hình)



Khi form hiển thị, sẽ xuất ngay messageBox hiển thị lời chào như sau



Và khi người sử dụng Click và button hay vào form cũng sẽ xuất hiện các message box hiển thị các thông báo tương ứng như :

Button was clicked.

Form was clicked.

Hướng Dẫn:

// sự kiện khi Button được click

```
private void bntClickMe_Click( object sender, EventArgs e )  
{  
    MessageBox.Show( "Button was clicked." );  
}
```

// sự kiện khi load form

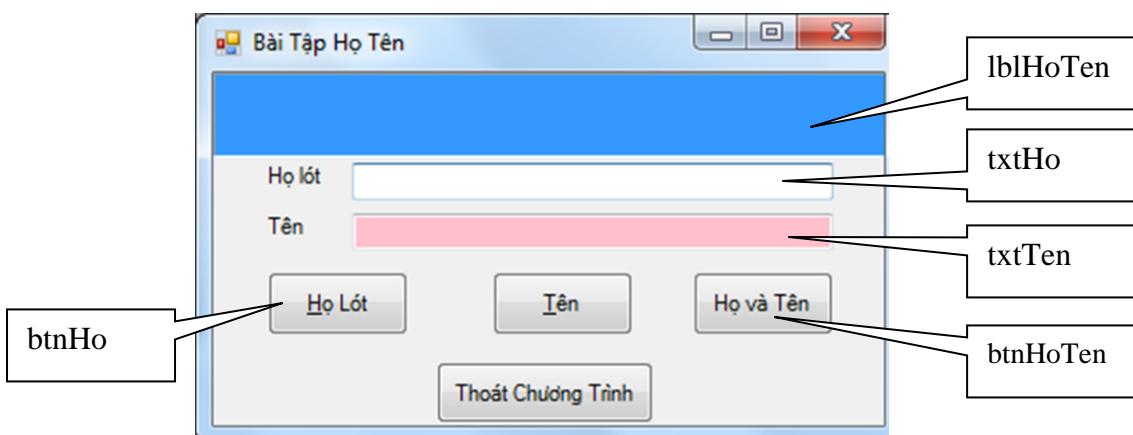
```
private void frmSimpleEvent_Load(object sender, EventArgs e)  
{  
    MessageBox.Show("Hi , Welcome to C# 2010 programming!");  
}
```

// sự kiện khi form được Click

```
private void frmSimpleEvent _Click(object sender, EventArgs e)  
{  
    MessageBox.Show("Form was clicked.");  
  
}
```

Bài 3:

Thiết kế giao diện như hình sau:



Viết lệnh để chương trình hoạt động như sau:

- Nhấn vào Button btnHo thì gán nội dung của txtHo cho lblHoTen
- Nhấn vào Button btnTen thì gán nội dung của txtTen cho lblHoTen
- Nhấn vào Button btnHoTen thì gán nội dung của txtHo cộng với txtTen cho lblHoTen

Hướng dẫn: *lblHoTen.Text = txtHo.Text + " " + txtTen.Text*

- Nhấn Double click vào lblHoTen thì nội dung của lblHoTen bị xoá
- Nhấn vào btnKetThuc thi đóng chương trình

Bài 4:

Mục Đích:

Sử dụng các control property và Layout trên Form

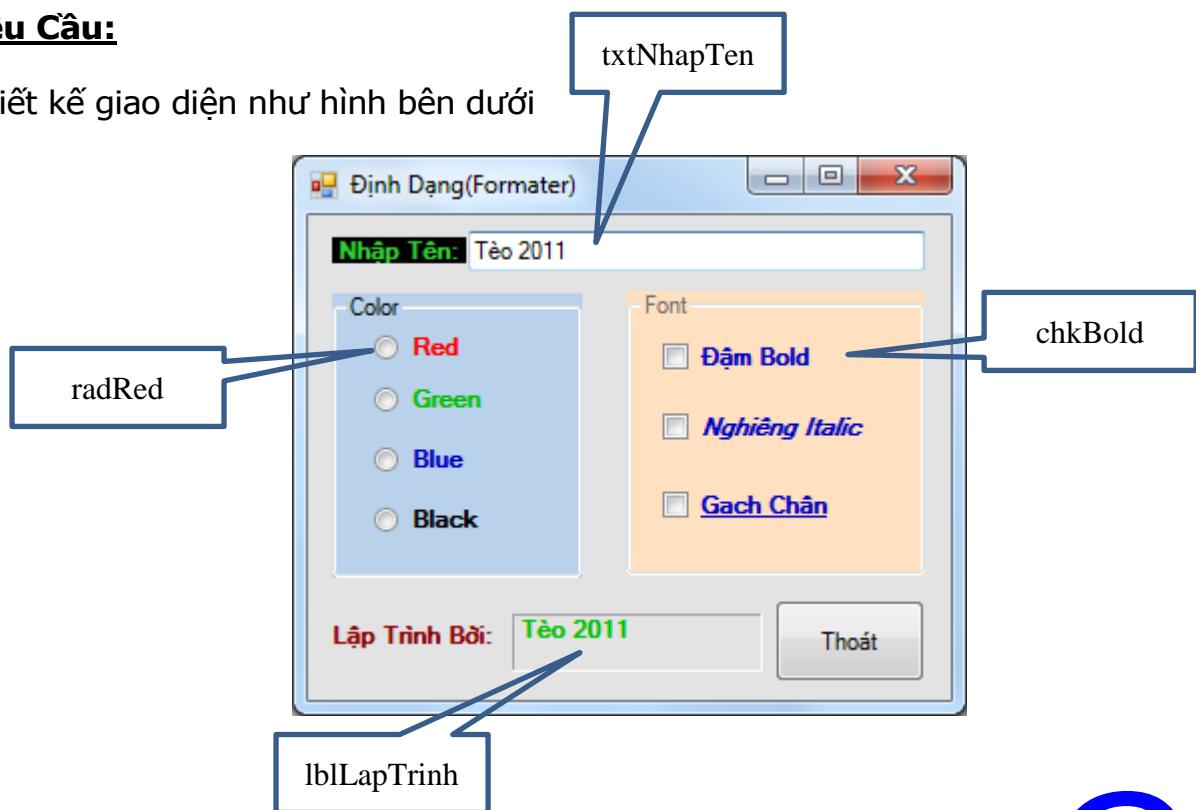
Yêu cầu:

Với giao diện của bài 2, yêu cầu sinh viên sử dụng thuộc tính Dock và Anchor sao cho khi người sử dụng resize kích thước form, thì các đối tượng trên form vẫn được sắp xếp hợp lý, dễ nhìn.

Bài 5:

Yêu Cầu:

Thiết kế giao diện như hình bên dưới



Yêu cầu

1. Khi Form load.
 - a. Radiobutton Red được chọn mặc định
 - b. Con trỏ xuất hiện ngay tại ô Nhập tên.
2. Khi gõ tên vào ô Nhập tên (gõ ký tự nào thì **table lập trình** chạy song song cùng giá trị trong ô nhập tên)

Hướng dẫn:

viết trong sự kiện txtNhapTen_TextChanged(..)

```
private void txtNhapTen_TextChanged(object sender, EventArgs e)
{
    lblLapTrinh.Text = txtNhapTen.Text
}
```

3. Nhấn nút thoát hoặc Esc thì thoát chương trình.
4. Nhấn Radiobutton Red, Green.. thì đổi màu chữ trong ô Lập trình bởi và ô nhập tên có màu tương ứng với Radiobutton phù hợp.

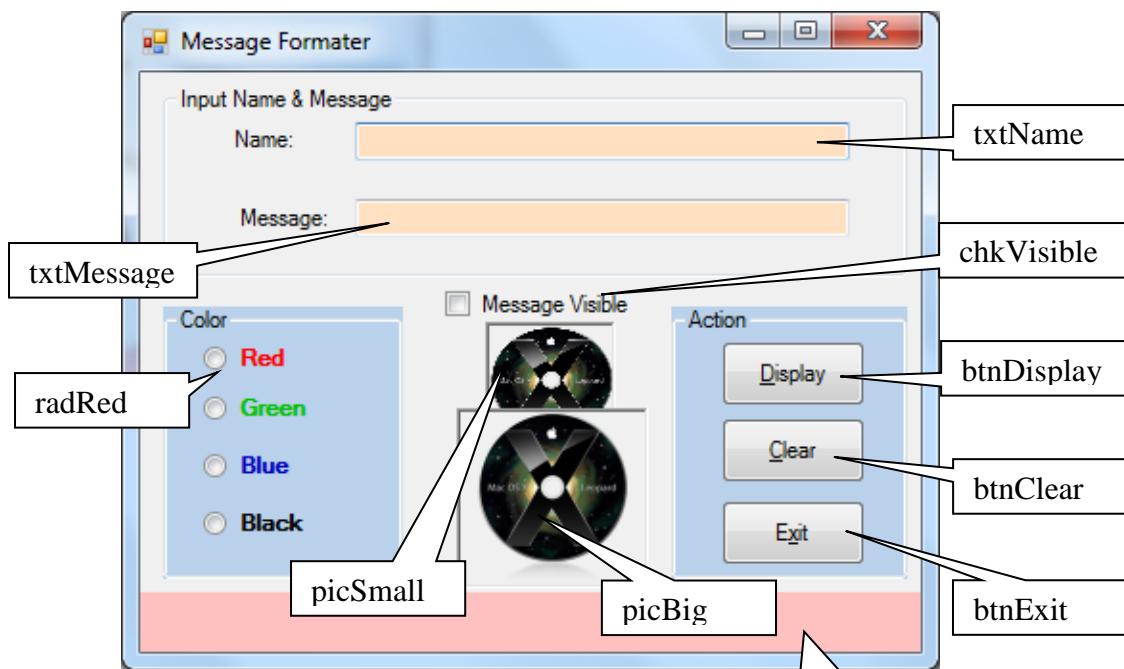
```
private void raddo_CheckedChanged(object sender, EventArgs e)
{
    lblLapTrinh.ForeColor = Color.Red;
    txtNhapTen.ForeColor = Color.Red;
}
```

5. Nhấn các CheckBox chữ đậm, nghiêng... thì đổi font chữ trong ô Lập trình bởi và ô nhập tên tương ứng.

```
private void chkdam_CheckedChanged(object sender, EventArgs e)
{
    lblLapTrinh.Font = new Font(lblLapTrinh.Font.Name, lblLapTrinh.Font.Size,
                               lblLapTrinh.Font.Style ^ FontStyle.Bold);
}
```

Bài 6:

Thiết kế giao diện như hình bên dưới



Viết lệnh để chương trình hoạt động như sau:

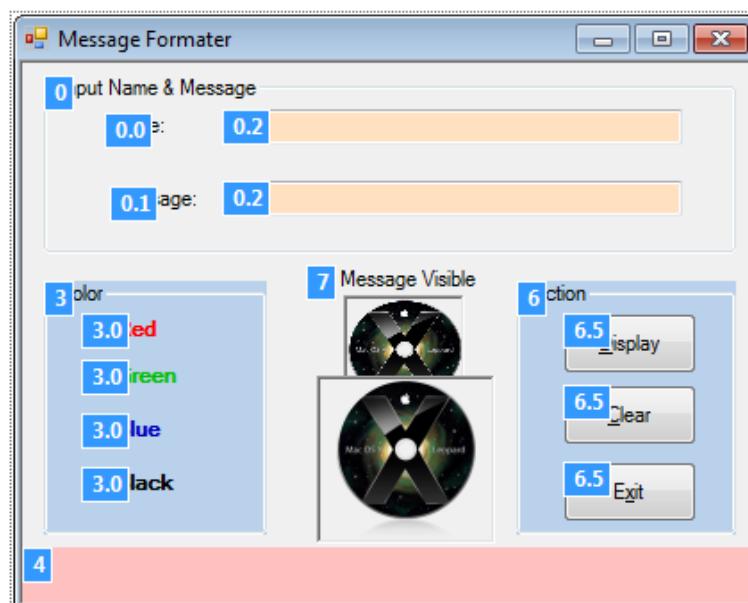
– Khi form vừa load lên thì:

- Con trỏ được đặt tại vị trí ô Name
- Hình CDRom lớn hiện ngược lại hình nhỏ ẩn.

Hướng dẫn: hiệu chỉnh property Visible của picSmall thành false (trong lúc thiết kế)

- Đưa chuột vào hình CDRom (lớn, nhỏ) hiện lên tooltip "Click Me"

– Hiệu chỉnhTabIndex sao cho hợp lý (Tham khảo hình vào Menu **View\Tab Order**)



- Nhấn vào Button Display hoặc phím Enter thì gán nội dung của ô Name và Message cho Label Message (lblMessage)

Hướng dẫn: *lblMessage.Text = txtName.Text + " : " +
txtMessage.Text;*

- Nhấn vào Button Clear thì xóa nội dung trong ô Name và Message

Hướng dẫn:

txtMessage.Clear();

txtName.Clear();

- Nhấn vào RadioButton (Red, Green ...) thì đổi màu chữ trong ô Message thành màu tương ứng.

Hướng dẫn: *lblMessage.ForeColor = Color.Red;*

- Ẩn hiện Message (lblMessage) phụ thuộc vào giá trị của Check box (chkVisible)

Hướng dẫn: *lblMessage.Visible = chkVisible.Checked;:*

- Nhấn vào Button Exit hoặc phím Esc thi đóng chương trình

This.Close();

- Click chuột vào hình CDRom nào thì hình đó ẩn đi và hiện hình CDRom khác lên

Hướng dẫn:

picBig.Visible = false;

picSmall.Visible = true;

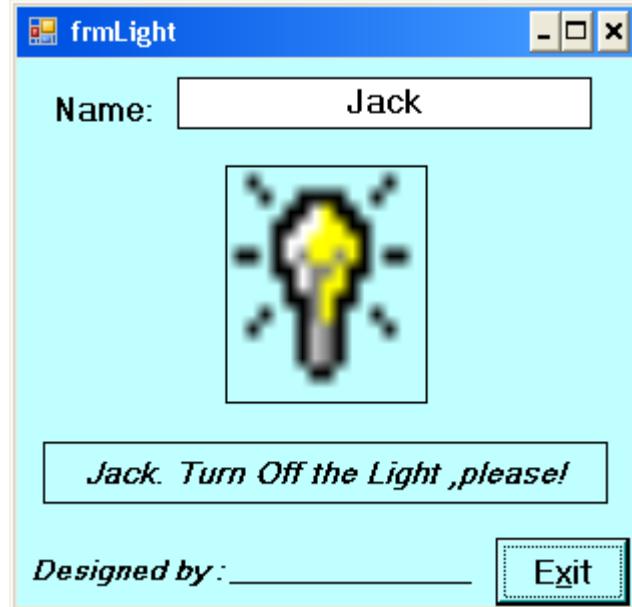
- Khi chương trình chạy hoàn chỉnh hãy hiệu chỉnh hai hình CDRom chồng lên nhau sao cho đồng tâm.

Bài 7:

Mục đích:

Kết hợp sử dụng component ToolTip trên các control

Yêu cầu



Yêu cầu:

1. Khi chương trình vừa thực thi.
 - a. Form được hiển thị giữa màn hình
 - b. Các đối tượng được “neo” sao cho khi người dùng thay đổi kích thước form thì các đối tượng được resize kích thước sao cho “dễ nhìn”
 - c. picTurnOn được hiển thị, picTurnOff được ẩn ngay bên dưới picTurnOn. Khi người dùng đưa chuột vào picTurnOn sẽ xuất hiện tip “Click me to Turn OFF the Light!”, và ngược lại trên picTurnOFF.
- Hướng dẫn: sử dụng component ToolTip**
- d. Hiện tên mặc định vào ô Name
 - e. Label ngay bên dưới pictureBox hiển thị thông báo “Jack. Turn Off the Light ,please!”

Hướng dẫn:

```
private void FrmLightSwitcher_Load(object sender, EventArgs e)
{
    txtName.Text = "Jack";
    lblHienThi.Text = txtName.Text + " Turn Off the Light ,please!";
}
```

2. Khi người dùng Click lên picture nào thì picture đó sẽ phải ẩn đi, picture còn lại xuất hiện. Đồng thời thay đổi nội dung trên label phù hợp.

```
private void picTurnOn_Click(object sender, EventArgs e)
{
    picTurnOn.Visible = false;
    picTurnOFF.Visible = true;
    lblHienThi.Text = txtName.Text + " Turn Off the Light ,please!";
}
```

3. Nút Exit cho phép đóng ứng dụng

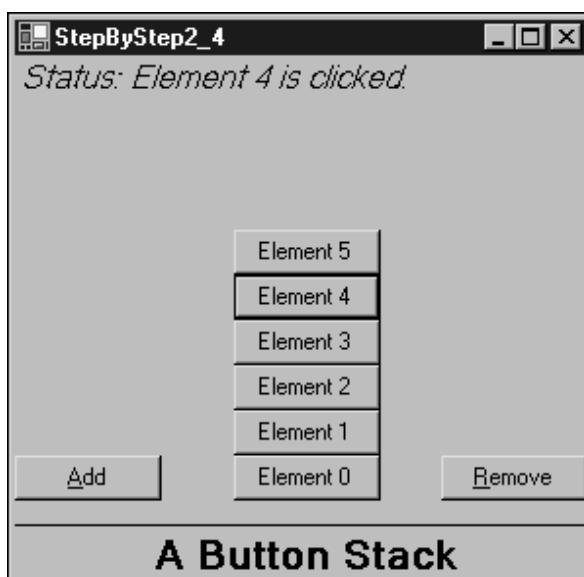
Bài 8: Adding Controls Programmatically

Mục đích:

Yêu cầu

Thiết kế form gồm 2 button Add và Remove. Khi người sử dụng Click và button Add, chương trình sẽ phát sinh động 1 button trên form (mỗi button là một Element thứ n). Khi button Remove được Click, thì button trên cùng sẽ được remove. (Có thể hình dung đây là một Stack). (xem hình)

Khi người dùng Click vào một Element nào thì sẽ hiển thị trạng thái của Element đó lên một label ở phía trên form. (xem hình)



Gợi ý:

Xem hướng dẫn thực hiện trong giáo trình

Bài 9:

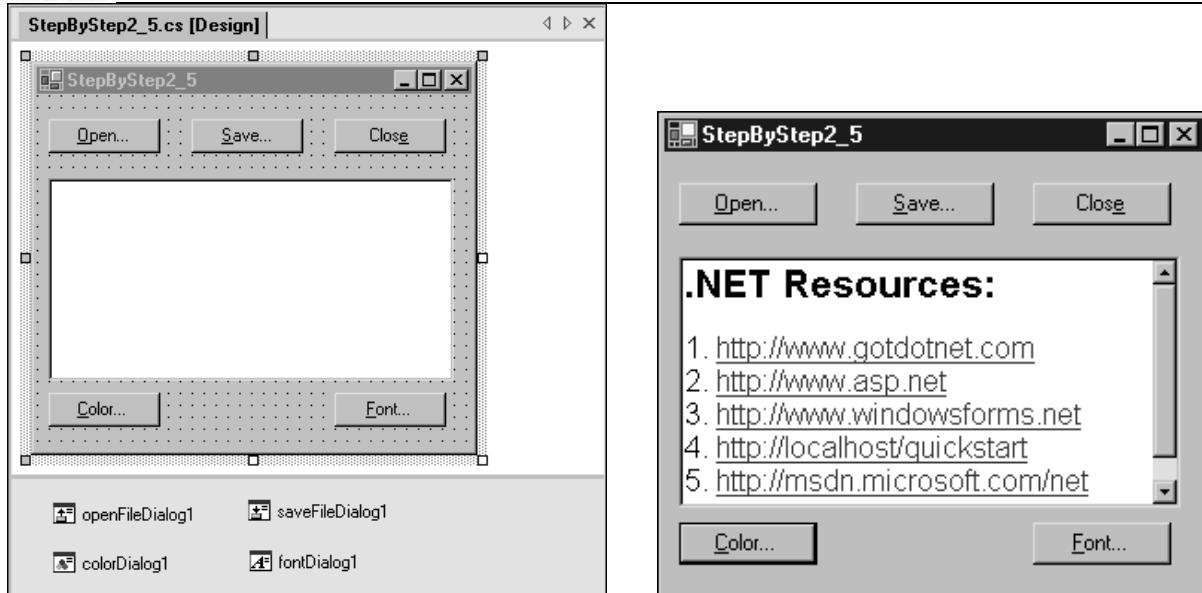
Mục đích:

Sử dụng các dialog được cung cấp bởi C#.

Sử dụng RichTextBox

Yêu cầu

Ex1: Using Common Dialog Boxes



Gợi ý:

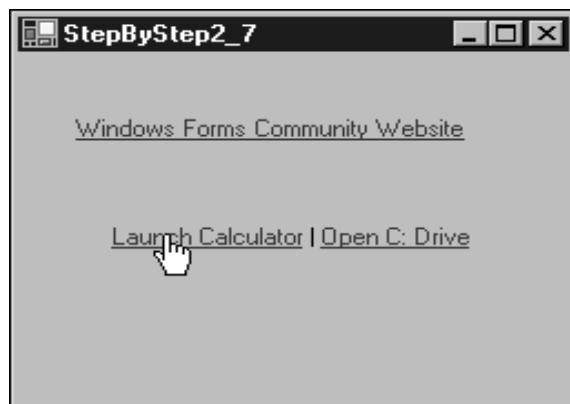
Xem hướng dẫn thực hiện trong giáo trình

Bài 10:

Mục đích:

Sử dụng đối tượng LinkLabel liên kết đến một WebSite, mở ứng dụng

Yêu cầu: Using LinkLabel Controls (page 456)



Gợi ý:

Xem hướng dẫn thực hiện trong giáo trình

Bài 11:

Mục đích:

Sử dụng GroupBox và Panel

Yêu cầu :

Using GroupBox and Panel Controls

**Gợi ý:**

Xem hướng dẫn thực hiện trong giáo trình

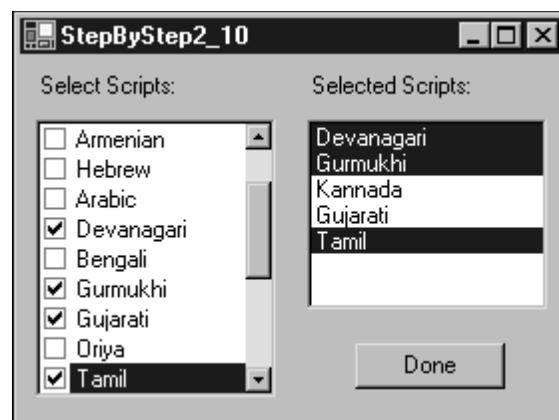
Bài 12:**Mục đích:**

Sử dụng ListBox, CheckedListBox, Combobox

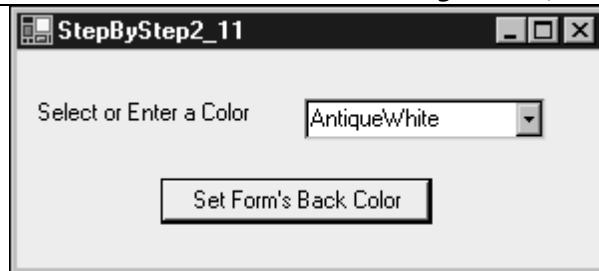
Yêu cầu

Step by Step

Ex1: Using ListBox and CheckedListBox Controls (page 477)



Ex2: Using ComboBox Controls

**Gợi ý:**

Xem hướng dẫn thực hiện trong giáo trình

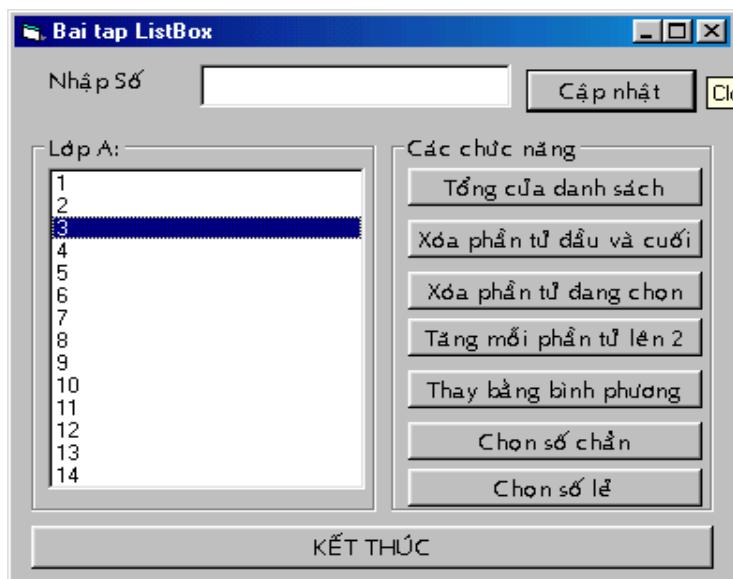
Bài 13:

Mục đích:

Thao tác dữ liệu số trên Listbox

Yêu cầu

Thiết kế form và thực hiện các chức năng sau:



Khi người sử dụng nhập một số vào textbox rồi Enter hoặc nhấp vào nút cập nhật thì số này được thêm vào listbox, đồng thời nội dung trong textbox bị xóa và focus được chuyển về textbox.

Người dùng nhấn vào nút nào thì thực hiện chức năng tương ứng của nút đó.

Gợi ý:

Lưu ý thuộc tính AcceptButton của form

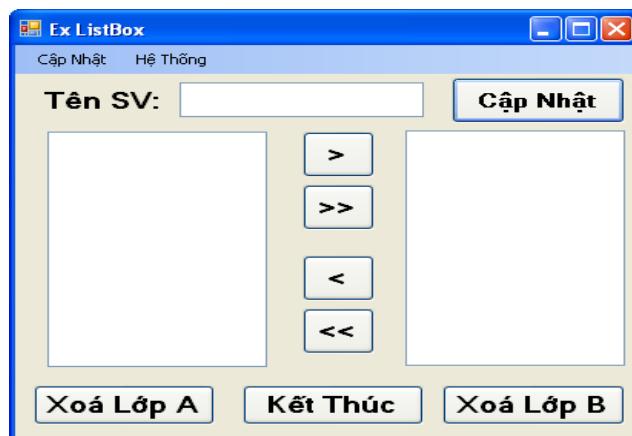
Phần tử đầu tiên trong danh sách có Index = 0

Bài 14:

Mục đích:

Yêu cầu

Thiết kế form và thực hiện các chức năng sau



- Quy định form hiển thị giữa màn hình. Không cho người sử dụng thay đổi kích thước form.
- Quy định việc di chuyển Tab hợp lý.
- Các ListBox được phép chọn nhiều mục (kết hợp giữa phím Shift, Control và chuột)
- Khi người sử dụng Click nút “Cập nhật” hoặc nhấn phím Enter thì mặc định nhập tên sinh viên từ textbox vào danh sách lớp A (không chấp nhận dữ liệu rỗng). Theo đó người sử dụng có thể chọn lớp để cập nhật từ combobox lớp.
- Các nút “>” và “<” khi được Click sẽ **di chuyển** tất cả các mục **đang chọn** sang ListBox bên kia tương ứng.
- Các nút “>>” và “<<” khi được Click sẽ di chuyển toàn các mục sang Listbox bên kia tương ứng.
- Nút lệnh “Xóa lớp A”, “Xóa lớp B” cho phép xóa các mục đang chọn trong list hiện hành.

Bài 15:

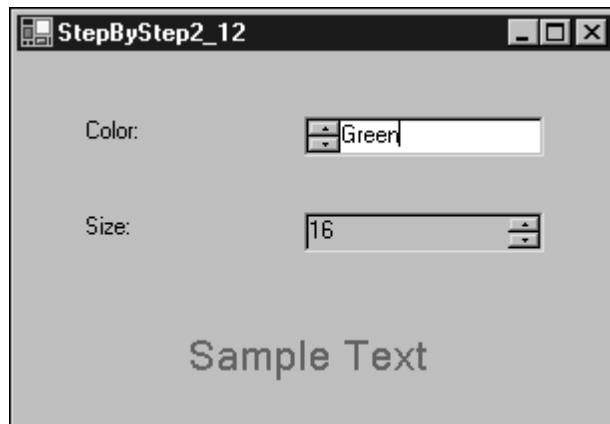
Mục đích:

Sử dụng các control của Visual Studio

Yêu cầu

Step by step

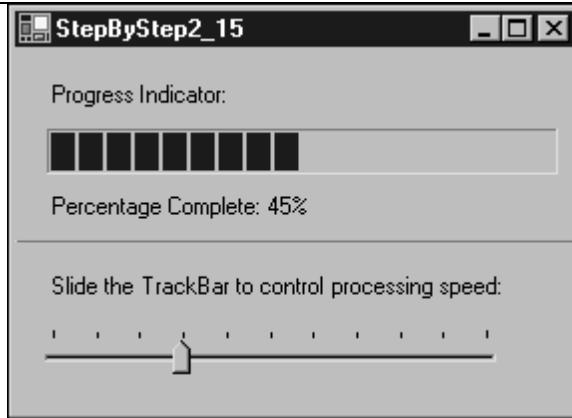
Ex1: Using DomainUpDown and NumericUpDown Controls



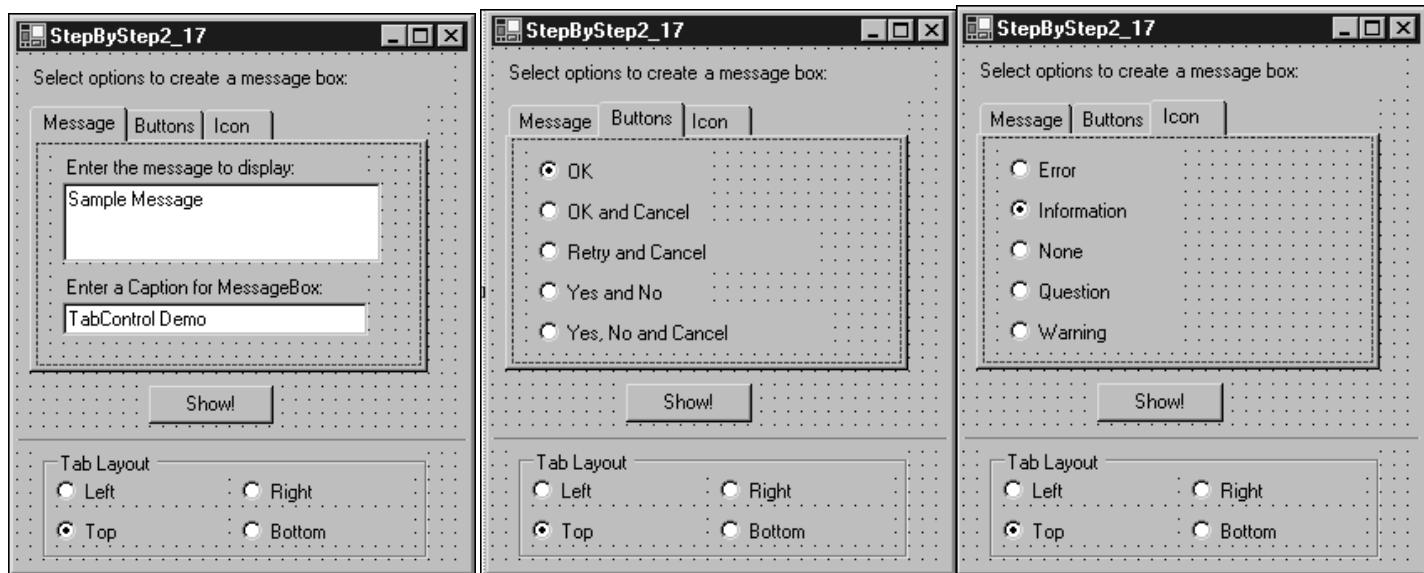
Ex2: The MonthCalendar and DateTimePicker Controls



Ex3: The Timer, TrackBar, and ProgressBar Controls



Ex4: Using TabControl Controls



Gợi ý:

Xem hướng dẫn thực hiện trong giáo trình (Page 491)

Bài 16:

Mục đích:

Xây dựng hệ thống menu : MainMenu và ContextMenu(menu cảm ứng cảnh) (Page 498)

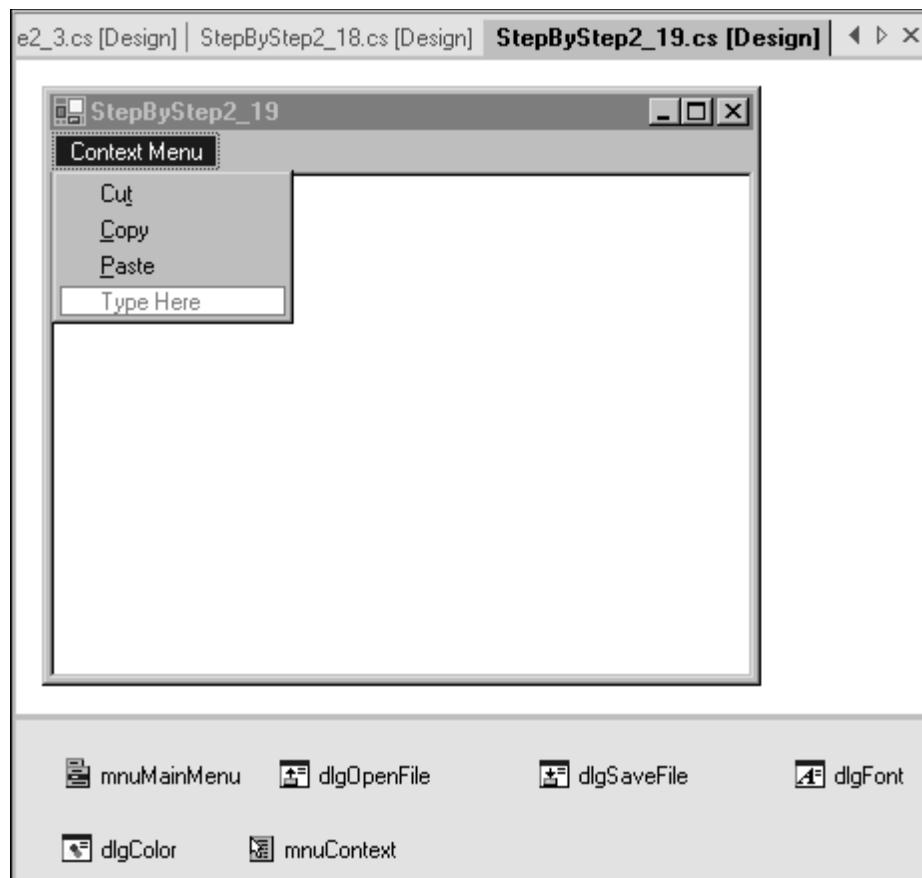
Yêu cầu

Step by Step

Ex: Creating a Main Menu for a Form



Ex : Creating a Context Menu for a Form



Gợi ý:

Xem hướng dẫn thực hiện trong giáo trình(page 498)

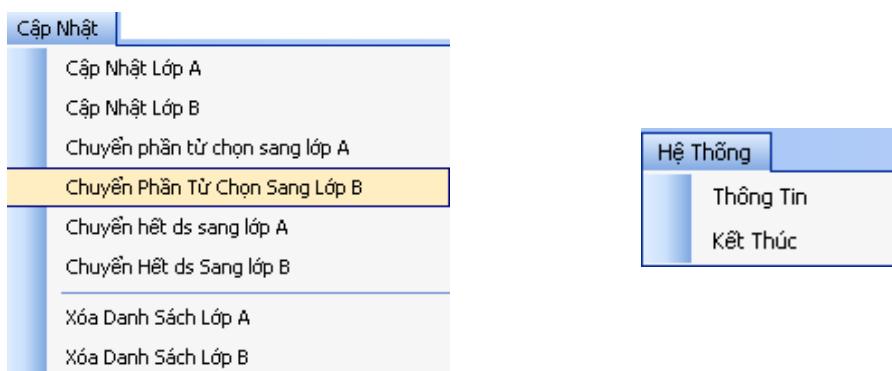
Bài 17:

Mục đích:

Sử dụng menu

Yêu cầu

Thiết kế bổ sung hệ thống menu cho Bài tập số 14.



- Các menu item sẽ thực hiện chức năng tương tự như các button của bài 14
- Menu “Thông tin” khi được Click sẽ xuất messageBox, cho biết thông tin của sinh viên thực hiện chương trình.

**Gợi ý:**

Từ sự kiện của menu item gọi lại các sự kiện của Button đã xử lý

Bài 18:**Mục đích:**

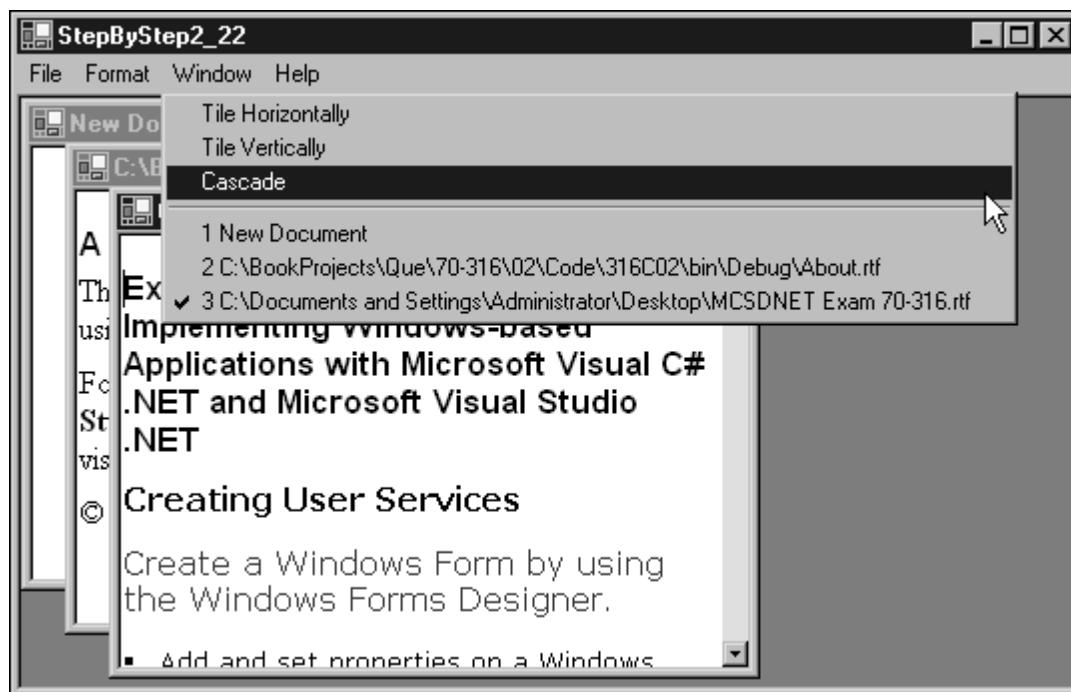
Xây dựng ứng dụng đa form,

Sử dụng form MDI

Yêu cầu

Step by step

Creating an MDI Application (page 512)

**Gợi ý:**

Xem hướng dẫn thực hiện trong giáo trình

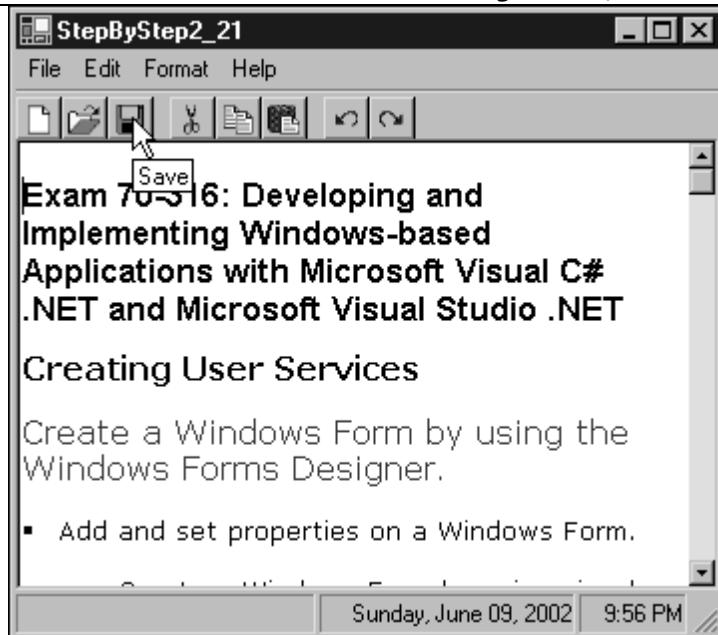
Bài 19:**Mục đích:**

Xây dựng ứng dụng kết hợp đối tượng statusBar

Yêu cầu

Step by step

Creating a Status Bar for a Form

**Gợi ý:**

Xem hướng dẫn thực hiện trong giáo trình

Bài 20:**Mục đích:**

Xây dựng ứng dụng liên kết nhiệm form

Yêu cầu

Step by step

Creating an MDI Application (Page 512)

Gợi ý:

Xem hướng dẫn thực hiện trong giáo trình

Bài 21:**Mục đích:**

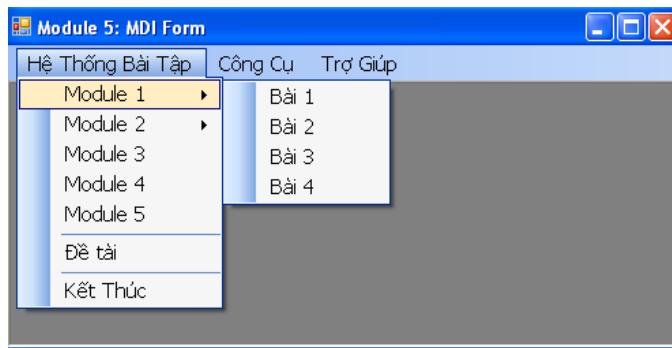
Tạo một ứng dụng liên kết các ứng dụng đã tạo ra từ các project trước.

Tạo form khởi động (form Flash)

Tạo form About

Yêu cầu

Tổ chức form chính như mẫu

**Gợi ý:**

Thiết kế Form:

```
StartPosition thiết lập = CenterToScreen;  
WindowState thiết lập = MaximumSize;
```

Liên kết các project đã có sẵn

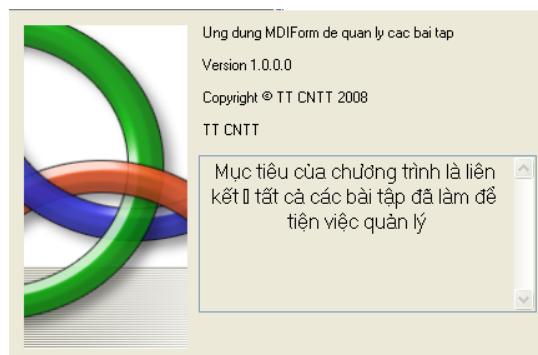
Vào menu Project chọn <Add Existing Items>

Chọn File cần Add vào chương trình (.cs)*

Viết code để liên kết form

```
frmFormOld f = new frmFormOld();  
  
f.MdiParent = this;  
  
f.Show();
```

- Thiết kế Form Splash (**frmFlashForm**)



Hướng dẫn:

Thiết kế Form:

FormBorderStyle thiết lập là *None*

StartPosition thiết lập là *CenterToScreen*

TopMost thiết lập = *true*

Thả vào form 1 đối tượng Timer để điều khiển việc đóng form tự động.

Thiết lập thuộc tính : Enabled=True và Interval = 10000

Trong sự kiện Timer_tick() gõ vào đoạn code

```
this.DialogResult = DialogResult.OK;  
timer1.Enabled = false;
```

- Thay đổi code trong class Program.cs

Viết lại hàm main như sau

```
static void Main()  
{  
    Application.EnableVisualStyles();  
    Application.SetCompatibleTextRenderingDefault(false);  
  
    frmFlashForm f = new frmFlashForm();  
    f.ShowDialog();  
    if (f.DialogResult == DialogResult.OK)  
    {  
        Application.Run(new frmMain());  
    }  
}
```

- Thiết kế Form About (tương tự form Splash)

Bài 22:**Mục đích:**

Tạo thanh công cụ (toolBar) trên ứng dụng

Yêu cầu

Step by step

creating a Toolbar for a Form (page 503)

Gợi ý:

Xem hướng dẫn thực hiện trong giáo trình

Module 3

Nội dung kiến thức thực hành:

- + Các kiểu dữ liệu trên C#
- + Cấu trúc điều khiển

Mục đích:

Sử dụng thành thạo các kiểu dữ liệu, các cấu trúc điều khiển cơ bản trên C#.

Bài 1:

Thiết kế giao diện như hình vẽ



Yêu cầu:

- Khi form thực thi
 - Tất cả text boxes và labels đều rỗng.
 - Đặt focus vào text box Tên Hàng và xác lập việc di chuyển Tab hợp lý ([menu View\Tab Order](#))
 - Form hiển thị giữa màn hình. ([StartPosition =CenterScreen](#))
 - Thiết lập nút Thanh Tính nhận sự kiện phím Enter ([AcceptButton](#)), nút Tiếp nhận sự kiện phím ESC ([CancelButton](#))
 - Thực hiện canh lề phải cho các đối tượng chứa dữ liệu là số ([Text Align=Right](#))
- **Xử lý nút lệnh "Thanh Tính".**

- o Các text box **Số lượng, Đơn Giá** buộc phải có giá trị và chỉ cho phép nhập số.

Thêm vào đó:

- **Số lượng >=0.** (int)
- **Đơn giá >=0** (decimal)

Cung cấp các Message Box khác nhau để thông báo rõ ràng cho việc dữ liệu bị nhập sai.

- o Tính toán cho mặt hàng hiện tại xuất kết quả ra label Thành Tiền.

$$\text{Thành tiền} = \text{số lượng} * \text{đơn giá};$$

- **Nút Tiếp:**

- o Xóa nội dung các textbox và label
- o Đặt focus vào textbox đầu tiên

- **Nút Kết Thúc:**

- o Phát sinh Messagebox hỏi lại việc đóng form có chính xác không, nếu người sử dụng xác nhận bằng nút Yes thì cho thoát chương trình

Hướng Dẫn :

Hướng dẫn:

1. Nút Thanh Tính

```
//khai báo biến cục bộ
int intSoLuong;
decimal decDonGia, decThanhTien;

//kiểm tra giá trị rỗng
if (txtSoLuong.Text == "")
{
    MessageBox.Show("Chưa nhập số lượng hàng");
    txtSoLuong.Focus();
}
else if (txtDonGia.Text == "")
{
    MessageBox.Show("Chưa nhập đơn giá hàng");
    txtDonGia.Focus();
}
else
{
    //nếu dữ liệu khác rỗng, gán vào biến
    intSoLuong = Convert.ToInt32(txtSoLuong.Text);
    decDonGia = Convert.ToDecimal(txtDonGia.Text);
}
```

```
//Kiểm tra dữ liệu hợp lệ
if (intSoLuong <= 0)
{
    MessageBox.Show("Số lượng hàng > 0 ");
    txtSoLuong.SelectAll();
    txtSoLuong.Focus();
}
else if (decDonGia <= 0)
{
    MessageBox.Show("Đơn Giá > 0 ");

    txtDonGia.SelectAll();
    txtDonGia.Focus();
}
else
{
    //Tính thành tiền
    decThanhTien = intSoLuong * decDonGia;
    //Xuất kết quả ra label
    lblThanhTien.Text = decThanhTien.ToString();
}
```

2. Chỉ cho nhập số vào TextBox (sự kiện KeyPress trên textbox)

```
private void txtSoLuong_KeyPress(object sender,
KeyEventArgs e)
{
    if ((e.KeyChar >= 'A' && e.KeyChar <= 'Z') ||
        (e.KeyChar >= 'a' && e.KeyChar <= 'z'))
        e.Handled = true;
}
```

3. Nút Kết Thúc

```
DialogResult dl;
dl = MessageBox.Show("Có chắc là bạn muốn đóng ứng dụng ?",
                     "warning",
                     MessageBoxButtons.YesNo,
                     MessageBoxIcon.Question);
if (dl == DialogResult.Yes)
    Close();
```

Bài 2:

Thiết kế và viết chương trình tính diện tích của hình chữ nhật. Phải kiểm tra dữ liệu nhập vào từ bàn phím một cách chặt chẽ.

Bài 3:

Thiết kế và viết chương trình tính chu vi của hình tròn có bán kính R nhập vào từ bàn phím (lưu ý kiểm tra dữ liệu chặt chẽ)(Chuvi=2*pi*R). Dùng hàm Math.PI

Bài 4

Thiết kế và xử lý chương trình giải phương trình bậc 1, 2.

Bài 5:

Mục đích:

Sử dụng hằng số, biến toàn cục (module variable)

Yêu cầu

Thiết kế giao diện như hình bên dưới.



Yêu cầu:

- 1) Sinh viên tự thiết kế các đối tượng trên form cho phù hợp
- 2) Thiết đặt TabIndex cho hợp lý.
- 3) Nhấn nút Tính hoặc Enter thì tính và hiện thông tin lên các ô tương ứng.
- 4) Nhấn vào Exit hoặc Esc thì thoát chương trình.

Ghi chú: sử dụng biến toàn cục và hằng số cho hợp lý

Bài 6:

Mục đích:

Sử dụng hằng số, biến toàn cục (module variable)

Yêu cầu

Thiết kế giao diện như hình bên dưới.



The screenshot shows a Windows application window titled "Quan Ly Sach". The main title bar is blue with the window title. Below it is a green header bar with the text "CHƯƠNG TRÌNH TÍNH TIỀN BÁN SÁCH". The main area is divided into two sections: "Hóa Đơn:" (Bill:) and "Thông kê:" (Statistics:).

In the "Hóa Đơn:" section, there are four input fields:

- Tên Khách Hàng: (Customer Name) with an empty text box.
- Số lượng Sách: (Quantity) with an empty text box.
- Khách Hàng là SV: (Customer is a student) with an empty checkbox.
- Thành Tiền: (Total Amount) with an empty text box.

In the "Thông kê:" section, there are three input fields:

- Tổng số KH: (Total number of customers) with an empty text box.
- Tổng số KH là SV: (Total number of student customers) with an empty text box.
- Tổng doanh thu: (Total sales) with an empty text box.

At the bottom of the window, there is a horizontal row of four buttons: "Tính TT", "Tiếp", "Thông Kê", and "Kết Thúc".

1. Khi chương trình vừa thực thi:

- Form được hiển thị giữa màn hình. Canh lề phải cho các control chứa dữ liệu là số, thiết lập hệ thống Tab hợp lý.
- Thiết lập nút “Tính TT” nhận phím Enter thay cho sự kiện Click.

2. Yêu cầu cài đặt

Tên Khách hàng không được phép rỗng.

Số lượng sách là số nguyên dương

Mỗi quyển sách đều được bán với giá là 20000.

- Nút “**Tính TT**”: tính thành tiền và xuất kết quả lên label Thành tiền

$$\text{Thành tiền} = \text{Số lượng} * \text{Đơn Giá} ;$$

Lưu ý: Nếu Khách hàng là SV thì được giảm 5% thành tiền.

- Nút “Tiếp”:

- Xóa nội dung các control trong groupbox “Hóa Đơn”
- Đặt focus cho Textbox “Tên Khách Hàng”
- Nút “Thông Kê”: Tính và hiển thị kết quả trên các label trong groupbox “Thông kê”
- Nút “Kết Thúc”: Phát sinh messageBox hỏi lại người dùng có thật sự muốn đóng ứng dụng hay không?

Bài 7:

Viết chương trình nhập vào năm dương lịch và xuất năm âm lịch tương ứng.

Biết rằng : Năm Âm = Can + Chi;

Can = Năm dương % 10 :

0	1	2	3	4	5	6	7	8	9
Canh	Tân	Nhâm	Quý	Giáp	Ất	Bính	Đinh	Mậu	Kỷ

Chi = dương lịch % 12 :

0	1	2	3	4	5	6	7	8	9	10	11
Thân	Dậu	Tuất	Hợi	Tý	Sửu	Dần	Mẹo	Thìn	Tỵ	Ngọ	Mùi



```
//Khai báo biến
int intNamDuong;
string strCan="";
string strChi="";
```

```
//1. kiem tra khac rong  
  
//2. Gan Gia tri  
  
//3. kiem tra so duong  
  
  
//tinh can  
  
switch (intNamDuong % 10)  
{  
    case 0:  
  
        strCan = "Canh";  
  
        break;  
  
    .  
    .  
    .  
  
    case 9:  
  
        strCan = "Kỷ";  
  
        break;  
}  
  
//Tinh Chi  
  
switch (intNamDuong % 12)  
{  
    case 0:  
  
        strChi = "Thân";  
  
        break;  
  
    .  
    .  
    .  
  
    case 11:  
  
        strChi = "Mùi";  
  
        break;  
}  
  
//xuất kết quả ra label
```


Bài 8:

Chương trình quay số may mắn



Thiết kế giao diện như hình trên

- Groupbox chứa 3 label.
- 2 Label dùng để hiển thị kết quả (anh lề phải)
- 3 button như hình vẽ.

Yêu cầu

Khi chương trình vừa bắt đầu

- 3 label sẽ hiển thị 3 số 7.
- Máy tính có 100 đồng
- Người chơi có 100 đồng

Khi Click nút quay số

Luật chơi:

Mỗi lần quay số, người chơi phải trả trước 30 đồng, tương ứng là máy sẽ được cộng thêm 30 đồng. Và nếu người chơi có số tiền ít hơn 30 đồng thì sẽ không được phép chơi tiếp

Nếu mỗi lần quay được số 7, người chơi sẽ được thưởng như sau

- Nếu ô đầu tiên là số 7 (random từ 0 đến 8), được thưởng 100 đồng + 50% số tiền của máy.
- Nếu ô số 2 là số 7 (random từ 0 đến 9), người chơi được thưởng 30 đồng + 50% tiền của máy.
- Nếu ô số 3 là số 7 (random từ 0 đến 10), người chơi được thưởng 10 đồng;

(số tiền người chơi sẽ được cộng dồn nếu cả 3 ô cùng là số 7, tiền máy sẽ giảm đi phần trăm tương ứng)

Nút game mới

Cho phép 1 lượt chơi mới hoàn toàn (set lại tên người chơi và tiền máy là 100)

Hướng dẫn:

//Khai báo biến

```
decimal tienmay = 100;
```

```
decimal tiennguoi = 100;
```

//Xử lý trên sự kiện của timer

```
private void timer1_Tick(object sender, EventArgs e)
```

```
{
```

```
    decimal tienthuong;
```

```
    int so;
```

```
    int so3;
```

```
    int so2;
```

```
    timer1.Interval += 5; //tăng giá trị của timer
```

//tạo số ngẫu nhiên

```
    Random ran = new Random();
```

```
    so = ran.Next(0, 8);
```

```
    so2 = ran.Next(0, 9);
```

```
    so3 = ran.Next(0, 10);
```

//xuất số ngẫu nhiên ra label

```
    lblSo.Text = so.ToString();
```

```
    lblSo2.Text = so2.ToString();
```

```
IblSo3.Text = so3.ToString();
```

```
//kiểm tra số ngẫu nhiên vừa tạo ra có phải là số 7 hay không
```

```
if (timer1.Interval > 200) // điều kiện dừng của timer
```

```
{
```

```
    timer1.Stop();
```

```
    if (so == 7)
```

```
{
```

```
    //tính tiền thưởng
```

```
    //tính tiền người chơi nhận được
```

```
    //tính tiền máy
```

```
    tienthuong = 100 + (tienmay * 50) / 100;
```

```
    tiennguo = tiennguo + tienthuong;
```

```
    tienmay = tienmay - (tienmay * 50) / 100;
```

```
//xuất thông báo
```

```
    MessageBox.Show("Bạn được thưởng " + tienthuong + " đồng", "Lucky Seven !");
```

```
}
```

```
if (so2 == 7)
```

```
{
```

```
    //tính tiền thưởng
```

```
    //tính tiền người chơi nhận được
```

```
    //tính tiền máy
```

```
}
```

```
if (so3 == 7)
```

```
{
```

```
    //tính tiền thưởng
```

```
    //tính tiền người chơi nhận được
```

```
    //tính tiền máy
```

{

//hiển thị kết quả

lblTienNguoiChoi.Text = tiennguo.ToString();

lblTienMay.Text = tienmay.ToString();

}

}

//xử lý nút quay số

private void btnQuaySo_Click(object sender, EventArgs e)

{

if (tiennguo >= 30)//kiểm tra người chơi có đủ tiền không

{

timer1.Enabled = true;//kích hoạt timer

timer1.Interval = 2;//quy định thời gian tái kích hót timer

tienmay = tienmay + 30;//tăng tiền của MÁY

tiennguo = tiennguo - 30;//giảm tiền người chơi

//hiển thị kết quả ra màn hình

}

Else//không còn đủ tiền: over Game

{

// Thông báo không thể chơi tiếp cho không đủ tiền**// Khóa timer****// khóa nút quay số**

}

}

Bài 9:

Viết chương trình cho phép phát sinh 1 số nguyên thuộc [0,9], sau đó cho người dùng đoán số và chương trình sẽ hiện kết quả xem người dùng có đoán đúng số phát sinh đó hay không.

HD: Dùng lớp Random để phát sinh số ngẫu nhiên.

Bài 10:

Thiết kế giao diện trình bày sơ đồ các chỗ ngồi của một rạp chiếu phim giúp người sử dụng chọn vị trí muốn mua (xem hình bên dưới). Trên sơ đồ này cũng thể hiện những vị trí đã bán vé và những vị trí chưa bán vé bằng cách thể hiện màu khác nhau (*ghế chưa bán vé màu trắng, ghế đã bán vé màu vàng, ghế đang chọn màu xanh*).



Khi khởi động, ứng dụng phải nằm giữa màn hình. Không cho phép người dùng thay đổi kích thước form.

Khi người sử dụng nhấp chuột tại một vị trí trên sơ đồ thì:

- Nếu đây là vị trí **chưa bán vé** thì đổi màu của vị trí này sang **màu xanh** để cho biết đây là vị trí **đang chọn**.
- Nếu đây là vị trí **đang chọn** (có màu xanh) thì đổi màu của vị trí này trở về **màu trắng**
- Nếu đây là một vị trí **đã bán vé** thì xuất hiện một thông báo cho người sử dụng biết thông qua **messageBox**.

Sau khi đã chọn các vị trí người sử dụng có thể nhấn vào nút CHỌN hoặc HỦY BỎ.

- Nếu click vào **nút CHỌN (hoặc nhấn phím Enter)** thì:

Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu vàng (cho biết vị trí đã bán vé) và xuất lên label thành tiền phải trả cho số vé đã mua (*giá sử giá vé là 100đ/ghế*)

- Nếu click vào **nút HỦY BỎ (hoặc nhấn phím ESC)** thì:

Đổi màu các vị trí đã chọn (màu xanh) trên sơ đồ sang màu trắng trở lại và xuất lên label thành tiền giá trị 0

Bài 11:

Hãy tạo một Project dùng để quản lý giờ thuê máy của một phòng máy có giao diện và yêu cầu như sau:

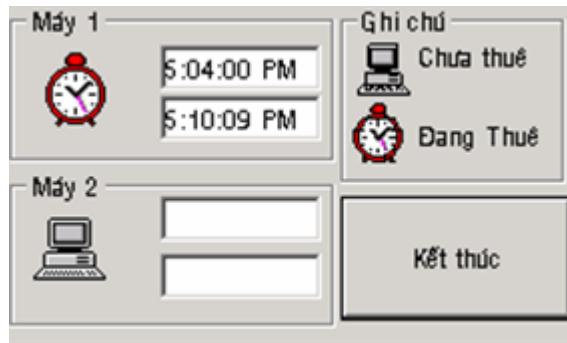
- Khi form vừa load lên màn hình thì
 - các CheckBox **Bắt đầu** có Enabled=True
 - các CheckBox **Kết thúc** có Enabled=False
 - Các Label có Caption= rỗng
- Khi người sử dụng Click vào CheckBox **Bắt Đầu** của một máy và nếu thuộc tính value của nó =1 thì:
 - Hiện Giờ hiện tại lên Label **bắt đầu** của máy tương ứng
 - Cho thuộc tính Enabled của CheckBox Bắt đầu =False
 - Cho thuộc tính Enabled của CheckBox Kết thúc =True
- Khi người sử dụng Click vào CheckBox **Kết thúc** của một máy và nếu thuộc tính value của nó =1 thì:
 - Hiện Giờ hiện tại lên Label **Kết thúc** của máy tương ứng
 - Dưa vào Giờ và Phút (không cần quan tâm đến số giây) trên các label **Bắt đầu** và **Kết thúc** tính ra số tiền phải trả, rồi dùng MsgBox để thông báo cho người sử dụng như hình bên (tiền thuê máy là 3000đ/giờ)
 - Xóa nội dung trong các Label **Bắt đầu** và **Kết thúc**
 - Cho thuộc tính Enabled của CheckBox Bắt đầu =True
 - Cho thuộc tính Enabled của CheckBox Kết thúc =False
 - Cho thuộc tính Value của CheckBox Bắt đầu =0
 - Cho thuộc tính Value của CheckBox Kết thúc =0



Khi người sử dụng Click vào nút **Dừng chương trình** thì kiểm tra nếu có máy đang thuê phải nhắc nhở người sử dụng bằng MsgBox, khi nào người sử dụng thật sự muốn kết thúc thì mới dừng chương trình.

Bài 12:

Hãy tạo một Project dùng để quản lý giờ thuê máy của một phòng máy có giao diện và yêu cầu như sau:



- Khi form vừa load lên màn hình thì
 - Các image trong khung máy 1 và máy 2 có picture là **Máy tính**
 - Các Label có Text = rỗng

- Khi người sử dụng click vào image của **máy 1** hoặc **máy 2** thì:
 Nếu là máy chưa thuê:
 Đổi picture của image thành hình **Đồng hồ**
 Hiện giờ hiện tại vào label **bắt đầu** của máy (label phía trên)
 Nếu là máy đang thuê:
 Hiện giờ hiện tại vào label **Kết thúc** của máy (label phía dưới)
 Dựa vào Giờ và Phút (không cần quan tâm đến số giây) trên các label **Bắt đầu** và **Kết thúc** tính ra số tiền phải trả, rồi dùng MsgBox để thông báo (tiền thuê máy là 3000đ/giờ)
 Đổi picture của image thành hình **Máy tính**
 Xóa nội dung trong các Label **Bắt đầu** và **Kết thúc**
Khi người sử dụng Click vào nút **Kết thúc** thì kiểm tra nếu có máy đang thuê phải nhắc nhở người sử dụng bằng MsgBox, khi nào người sử dụng thật sự muốn kết thúc thì mới dừng chương trình.



Module 4

Nội dung kiến thức thực hành:

- + Xử lý ngoại lệ (Error Handling for the User Interface).
- + Xử lý lỗi khi nhập liệu (VALIDATING USER INPUT)

Bài 1 :

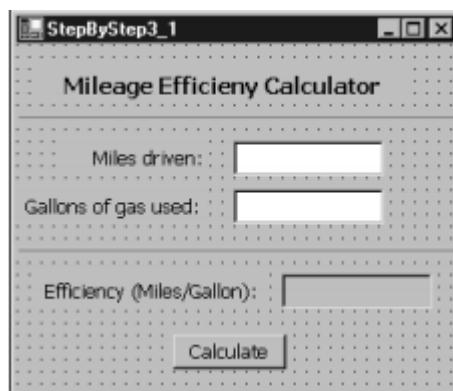
Mục đích:

Sử dụng lớp Exception để bẩy lỗi chương trình.

Sử dụng MessageBox để hiển thị thông báo lỗi

Yêu cầu:

Viết chương trình tính số nhiên liệu tiêu thụ trung bình theo giao diện sau"



Phải bắt lỗi chắc chắn các trường hợp sau:

- Số dặm đường, số gallon tiêu thụ là số dương.
- Lỗi tràn số
- Lỗi chia cho số 0
- ...

Hướng dẫn :

Giáo trình trang

Bài 3 :

Mục đích:

Sử dụng lớp Exception để bẩy lỗi chương trình.

Sử dụng MessageBox để hiển thị thông báo lỗi

Yêu cầu:

Viết lại bài 1, nhưng sử dụng đối tượng ErrorProvider để hiển thị các thông báo lỗi ngay trên control

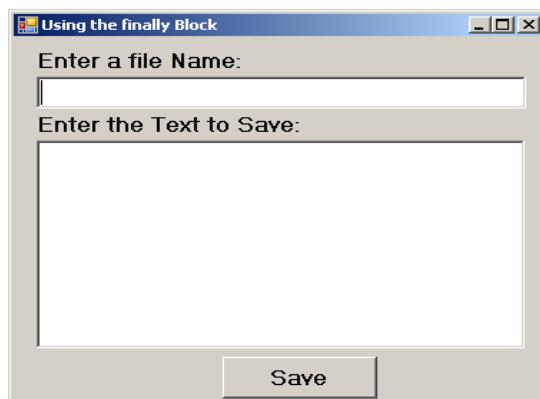
Bài 3 :**Mục đích:**

Sử dụng lớp Exception để bẩy lỗi chương trình.

Sử dụng finally Block

Yêu cầu:

Thiết kế giao diện như hình bên dưới. Khi người dùng Click nút “Save”, chương trình sẽ lưu lại nội dung trong TextBox save với tên được nhập vào trừ textbox name.

**Lưu ý:**

Chương trình phải thực hiện việc **đóng file bất kể có xảy ra lỗi hay không**.

Hướng dẫn:

Giáo trình trang

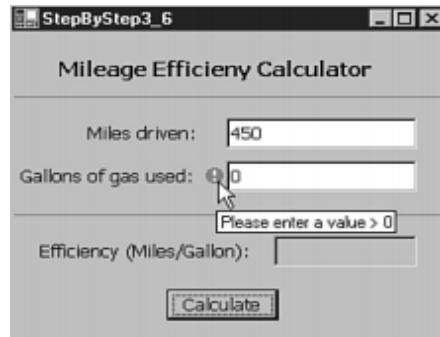
Bài 4 :

Mục đích:

Sử dụng Component ErrorProvider kết hợp kỹ thuật Validation trên các control

Yêu cầu:

Viết lại bài tập một nhưng sử dụng chặn lỗi người dùng bằng **kỹ thuật Validation.**



Lưu ý: Khi có lỗi nhập liệu trên các Textbox thì chỉ giữ focus không cho thực hiện tính toán, nhưng vẫn phải cho người dùng Click nút đóng form.

Hướng dẫn:

Thiết lập thuộc tính CausesValidation của Form là False

Khai báo thêm một biến toàn cục để kiểm tra người dùng Click đóng form

```
private bool closingFlag = false;
```

Trong sự kiện đóng form bổ sung code

```
private void frm_ValidatingUserInput_FormClosing(object sender,
                                                 FormClosingEventArgs e)
{
    closingFlag = true;
    e.Cancel = false;
}
```

Xem thêm giáo trình trang

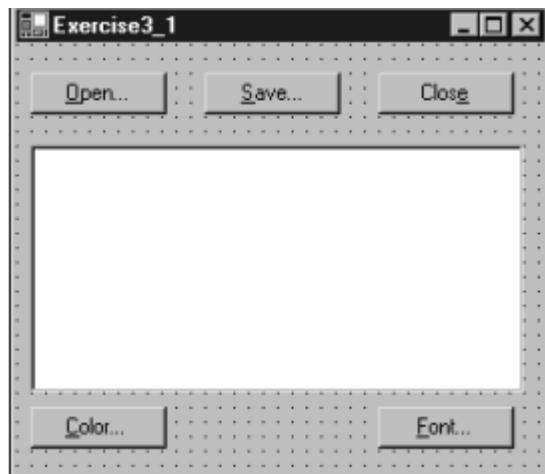
Bài 5 :

Mục đích:

Lập trình với lớp Exception thuộc không gian System.IO.IOException

Sử dụng các component OpenFileDialog, SaveFileDialog, ColorDialog, and FontDialog.

Yêu cầu:



Lưu ý: để đọc/ghi file cần bổ sung không gian “using System.IO”

Xem giáo trình trang

Bài 6 :

Mục đích:

Sử dụng kỹ thuật Validating User Input.

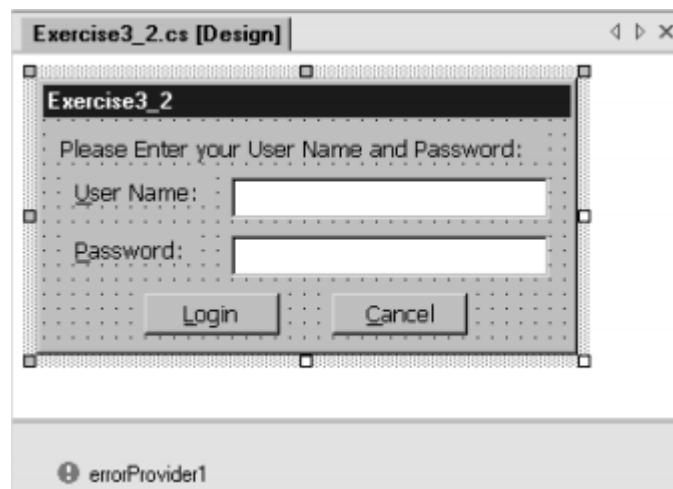
Yêu cầu:

Thiết kế form login như hình bên dưới.

Xuất thông báo login hợp lệ nếu nội dung của 2 textbox “UserName” và “Password” là **khác rỗng và không có bất kỳ khoảng trắng** nào bên trong. Ngược lại xuất thông báo lỗi cụ thể thông qua ErrorProvider.

Cho phép người dùng đóng Form bằng nút “Cancel” ngay cả khi xảy ra lỗi trên bất kỳ trường dữ liệu nào.

Xem thêm hướng dẫn trong giáo trình trang



Bài 7 :

Mục đích:

Sử dụng lớp Exception để bẫy lỗi chương trình

Sử dụng Component ErrorProvider kết hợp kỹ thuật Validation trên các control

Yêu cầu:

Thực hiện lại kiểm lỗi ở các bài tập thuộc Module3

Module 5

Nội dung kiến thức thực hành:

- + Tạo ứng dụng trên consoleApplication và windows form.
- + Sử dụng Mảng và danh sách

Bài 1:

Mục đích:

Xây dựng mảng một chiều trên ConsoleApp

Yêu cầu:

Nhập một mảng 1 chiều có N phần tử ($N < 20$) (các giá trị được phát sinh ngẫu nhiên trong [-100, 100])

- a. Tìm số lớn nhất, nhỏ nhất trong mảng. Xuất ra vị trí các giá trị này.
- b. Đếm xem trong mảng có bao nhiêu số dương, bao nhiêu số âm, bao nhiêu số 0.
- c. Nhập vào số nguyên x, kiểm tra xem x có trong mảng hay không.
- d. Tìm số dương nhỏ nhất trong mảng, số âm lớn nhất trong mảng
- e. In dãy giảm dần.

Bài 2:

Mục đích:

Xây dựng mảng một chiều trên ConsoleApp

Yêu cầu:

Nhập một mảng 2 chiều có $N \times N$ phần tử ($N < 20$) (các giá trị được phát sinh ngẫu nhiên trong [-100, 100])

- a. Tìm số lớn nhất, nhỏ nhất trong mảng. Xuất ra vị trí các giá trị này.
- b. Đếm xem trong mảng có bao nhiêu số dương, bao nhiêu số âm, bao nhiêu số 0.
- c. Đếm số dương, số âm, số 0 ở dưới đường chéo chính, chéo phụ.

- d. Tính tổng các phần tử trong đường chéo chính, chéo phụ.
- e. Tính tổng các phần tử nằm bên dưới đường chéo chính, chéo phụ.
- f. Tìm số lớn nhất trong từng dòng của mảng.

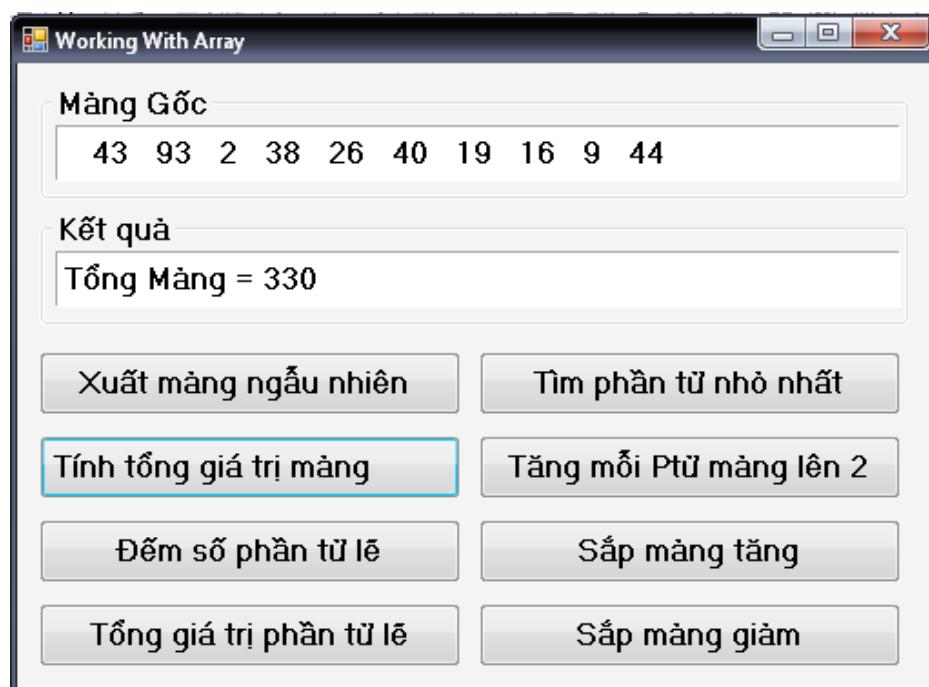
Bài 3:

Mục đích:

Sử dụng mảng một chiều

Yêu cầu:

Thiết kế Form như hình sau:



Trong đó 2 group box sẽ chứa 2 label, để hiển thị mảng và kết quả xử lý trên các button tương ứng.

Hướng dẫn :

1. khai báo mảng toàn cục, với kích thước mảng MAX_SIZE là 1 hằng số.

```
//Khai báo mảng toàn cục
const int MAX_SIZE = 10;
int[] intMyArray;
```

2. KHởi tạo mảng trong Form_Load

```
//Khởi tạo mảng trong Form_Load
intMyArray = new int[MAX_SIZE];
```

3. Xây dựng các phương thức theo yêu cầu

```
//Định nghĩa phương thức nhập mảng (thủ tục, không giá trị trả về)
private void NhapMang()
{
    Random r = new Random();

    for (int i = 0; i < intMyArray.Length; i++)
        intMyArray[i] = r.Next(1, 101);
}

//Định nghĩa phương thức Xuất mảng (thủ tục, không giá trị trả về)
private void XuatMang(int[] a)
{
    for (int i = 0; i < a.Length; i++)
        lblGoc.Text += " " + a[i].ToString();
}

//Định nghĩa phương thức tính tổng mảng (hàm, giá trị trả về kiểu int)
private int TongMang(int[] a)
{
    int Kq = 0;
    for (int i = 0; i < a.Length; i++)
        Kq += a[i];
    return Kq;
}
```

4. Gọi sử dụng các phương thức tại các button.

```
private void btnXuatMang_Click(object sender, EventArgs e)
{
    this.NhapMang(); //gọi thủ tục
    this.XuatMang(intMyArray);
}

private void btnTongMang_Click(object sender, EventArgs e)
{
    //gọi hàm. lấy giá trị trả về của hàm xuất là Label
    lblKetQua.Text = "Tổng Mảng = " + TongMang(intMyArray).ToString();
}
```

Bài 4:

Mục đích:

Sử dụng mảng hai chiều trên ConsoleApp

Yêu cầu:

Nhập một mảng 2 chiều có N x N phần tử (N<20) (các giá trị được phát sinh ngẫu nhiên trong [-100,100]). Tìm số dương nhỏ nhất trong mảng.

Ví dụ:

-9	8	3
-3	5	0
1	-30	7

Số dương nhỏ nhất trong mảng là 1

Bài 4:

Mục đích:

Sử dụng mảng hai chiều trên ConsoleApp

Yêu cầu:

Nhập một mảng 2 chiều có N x N phần tử (N<20) (các giá trị được phát sinh ngẫu nhiên trong [-100,100]). Hãy trích mảng này ra thành hai mảng con: một mảng chứa các số dương, một mảng chứa các số âm (giữ nguyên thứ tự dòng)

Ví dụ:

9	8	3
-3	5	0
1	-30	7

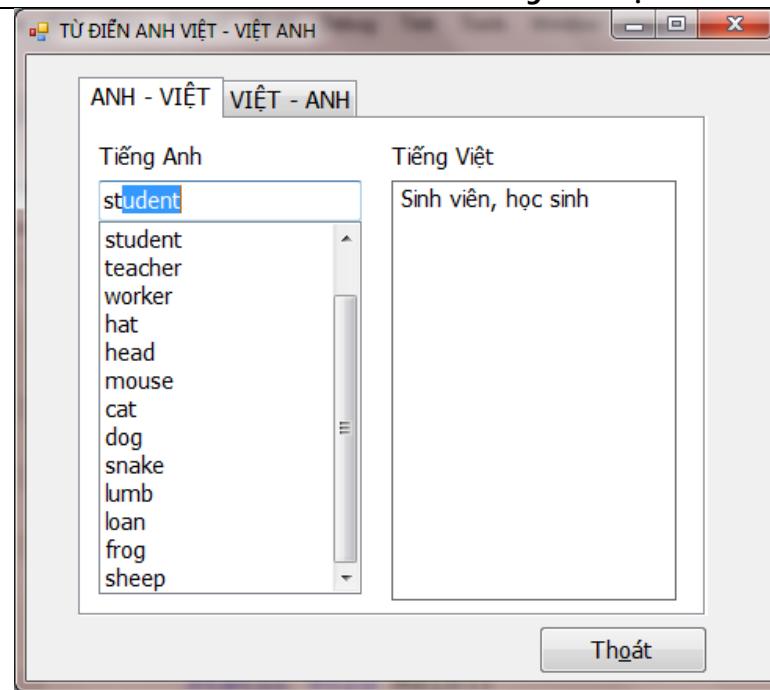
Bài 5:

Mục đích:

Sử dụng ArrayList,

Yêu cầu:

Xây dựng từ điển đơn giản như sau



- Khi nhập vào combobox 1 từ cần tra thì chương trình sẽ dò tìm đến chữ nào khớp với ký tự gần nhất
- Khi nhấn Enter hoặc Doubleclick vào từ cần tra thì nghĩa tương ứng của từ sẽ hiển thị vào textbox bên phải tương ứng.
- Danh sách các từ lưu sẵn vào object (word) → ArrayList

Hướng Dẫn:

Bên trái là Combobox thể hiện dưới dạng Simple, chứa danh sách các từ cần tra cứu.
Bên phải là TextBox thể hiện dưới dạng MultiLine, ghi nghĩa của các từ được chọn bên Combobox.

Module 6

Nội dung kiến thức thực hành:

- + Tạo ứng dụng theo mô hình hướng đối tượng (OOP).

Bài 1:



Yêu cầu:

Thiết kế class Hóa Đơn đáp ứng yêu cầu xử lý của giao diện trên.

Hướng dẫn:

1. Tạo các field private:

Tên hàng, số lượng, đơn giá.

2. Tạo các field static:

Số hóa đơn, doanh thu.

3. Tạo các property thông thường.

4. Tạo property ReadOnly đối với các Field static.

5. Tạo các Contructor

6. Tạo các phương thức

Bài 2:**Yêu cầu:**

1. Thiết kế Form như hình sau:



2. Thiết kế class Hóa Đơn đáp ứng yêu cầu xử lý của giao diện trên.

Hướng dẫn:

- Tạo các field private:

- o Họ tên, địa chỉ, điện thoại, giá tiền, màu xe.

- Tạo các field static:
 - o Số khách hàng, số khách mua xe đen, doanh thu.
- Tạo các property thông thường.
- Tạo property ReadOnly đối với các Field static.
- Tạo các Contructor

Ví dụ:

```
public HoaDon(string hoten, string diachi,
              decimal dienthoai,
              decimal giatien, bool muaxeden)
{
    HOTEN = hoten;
    DIACHI = diachi;
    DIENTHOAI = dienthoai;
    GIATIEN = giatien;
    MUAXEDEN = muaxeden;
    tongkh++; //đếm số khách hàng
    if (MUAXEDEN == true)
        tongkhmuaxeden++; //đếm riêng KH mua xe đen
}
```

- Xây dựng các phương thức xử lý tính toán.

Ví dụ:

```
public decimal phuthu()
{
    if (MUAXEDEN == true)
        return GIATIEN * (decimal)0.05;
    else
        return 0;
}
```

Module 7

Nội dung kiến thức thực hành:

Sử dụng các chuỗi (String)

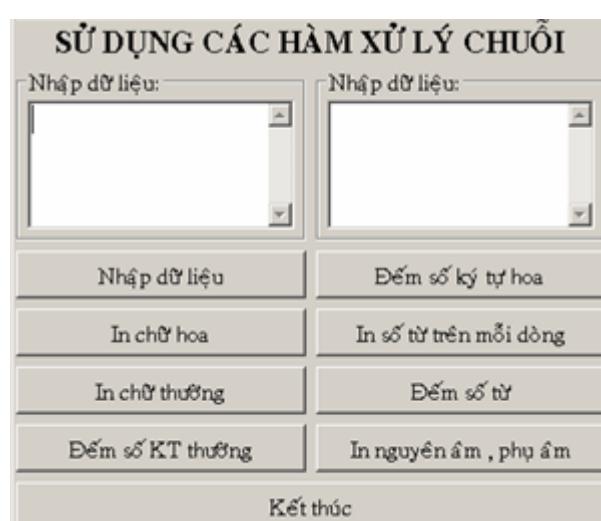
Bài 1:

Viết chương trình có giao diện và thực hiện các yêu cầu sau:

Khi form được load lên thì chỉ có nút nhập liệu , nút kết thúc là sử dụng được, nút nhập dữ liệu được focus.

Khi nút nhập dữ liệu được ấn thì các nút lệnh khác được sáng lên xoá nội dung của Textbox nhập liệu và Setfocus cho Textbox này (sử dụng phương thức Setfocus).

Khi một trong các nút (ngoại trừ nút Nhập Dữ Liệu và nút Kết Thúc) được chọn thì kết quả sẽ hiện thị lên Textbox và các nút lệnh sẽ bị mờ (ngay cả nút vừa nhấp).



Bài 2:

Tạo màn hình đếm số từ như sau



Dựa vào chuỗi nhập, đếm số từ đã nhập theo quy định các từ ngăn cách nhau bằng ký tự trắng (space).

Bài 3:

Tạo một Project như yêu cầu sau:

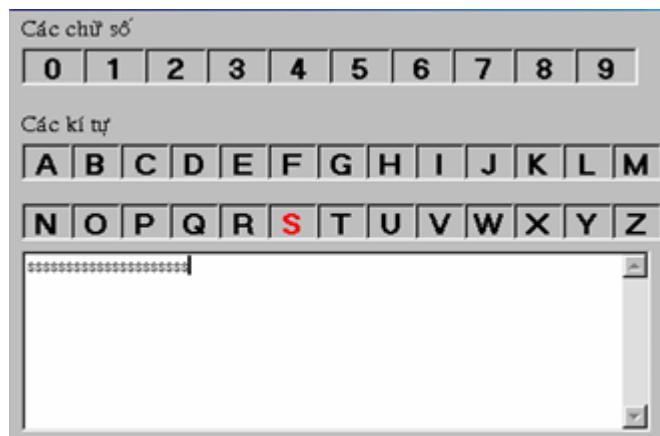
Khi chọn nút Search thì chương trình sẽ thông báo cho người sử dụng biết nội dung nhập trong "Search string" có xuất hiện trong "Main string" hay không

Khi chọn nút Replace thì chương trình sẽ thay thế trong Main String chuỗi giá trị Search String bằng chuỗi Replace String.



Bài 4:

Viết chương trình thực hiện việc thực hành đánh máy chữ, giao diện như hình bên.
Yêu cầu : Khi nhấn 1 kí tự hoặt 1 số nào thì kí tự hoặt số đó chuyển sang màu đỏ, khi thả phím ra thì màu chữ trở lại màu đen. Đồng thời ký tự vừa được nhấn sẽ hiện bên dưới Text box.



Bài 1:

Thiết kế giao diện và viết code để thực hiện những yêu cầu sau:

- Cho phép nhập một chuỗi Họ Tên
- Xuất Họ, Chữ lót, Tên của chuỗi họ tên vào các label tương ứng khi click chuột vào một nút lệnh
- Tạo điều kiện để người sử dụng thuận lợi khi nhập họ tên mới.
- Có nút lệnh để đóng chương trình và phải hỏi người sử dụng trước khi thật sự kết thúc chương trình