**BÀI TẬP LẬP TRÌNH WINDOWS**

GV: PHẠM MẠNH CƯƠNG

**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 hòan 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 trên windows form.

Mục đích:

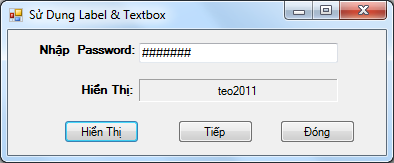
Sử dụng các properties, methods và event của các đối tượng trong Windows

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 “Hiễn Thị’ thì nội dung của password sẽ được hiễn thị ra trên label bên dưới

**Hướng dẫn:**

*Trong sự kiện Click của button “Hiễn Thị”, gõ vào câu lệnh sau*

**lblHienThi.Text = txtPassWord.Text;**

###### **+** Người sử dụng nhấn nút “Tiếp” để xoá nội dung textbox, label và đặt con trỏ vào texttbox 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*

**lblHienThi.Text = ""; txtPassWord.Clear();**

**txtPassWord.Focus();**

**+** Nhấp 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();

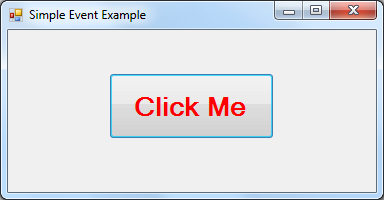
**Bài 2:**

##### Mục Đích:

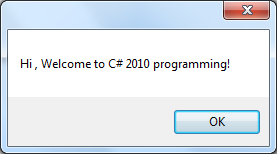
###### Sử dụng các sự kiện (event) cơ bản trên Form

**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 sủ dụng Click và button hay vào form củng sẽ xuất hiện các message box hiể 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

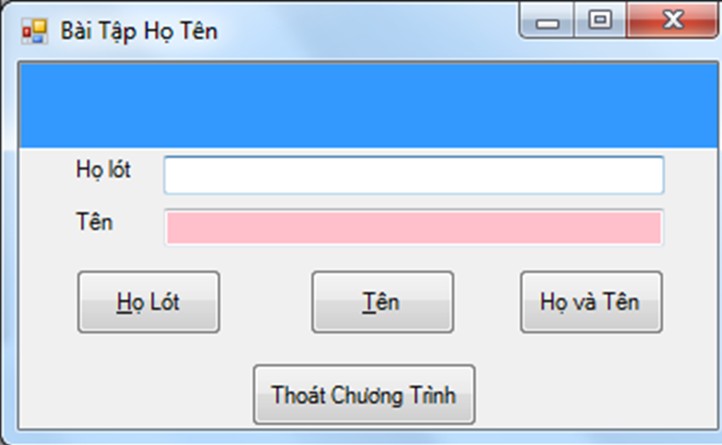
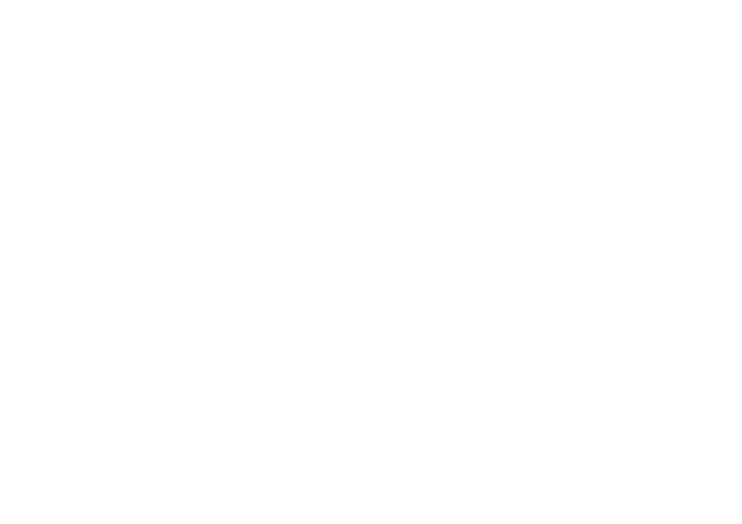
private void frmSimpleEvent \_Click(object sender, EventArgs e)

{

MessageBox.Show("Form was clicked.");

}

**Bài 3:**



Thiết kế giao diện như hình sau:

lblHoTen

txtHo

txtTen

btnHo

btnHoTen

###### 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:**

txtNhapTen

Thiết kế giao diện như hình bên dưới

chkBold

radRed

lblLapTrinh

Yêu cầu

1. khi Form load.
   1. Radiobutton Red được chọn mặc định
   2. Con trỏ xuất hiện ngay tại ô Nhập tên.
2. Khi gõ tên vào ô Nhập tên (gõ ký tự nào thì ***lable 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

**}**

1. Nhấn nút thoát hoặc Esc thì thoát chương trình.
2. 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;

**}**

1. 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)**

**{**

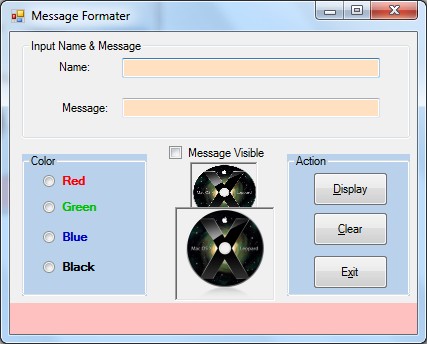
**lblLapTrinh.Font =** new Font**(lblLapTrinh.Font.Name, lblLapTrinh.Font.Size,**

**lblLapTrinh.Font.Style ^** FontStyle.Bold);

**}**

**Bài 6:**

###### Thiết kế giao diện như hình bên dưới



txtName

txtMess**e**age

chkVisible

btnDisplay

radRed

btnClear

picSmall

picBig

btnExit

Viết lệnh để chương trình hoạt động như sau:

lblMessage

* Khi form vừa load lên thì:

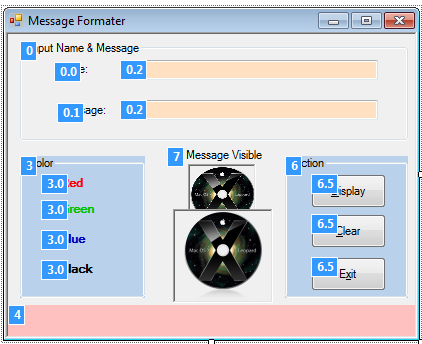
 Con trỏ được đặt tại vị trí ô Name

 Hình CDRom lớn hiện ngược lại hình nhỏ ẩn.

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

 Đưa chụôt vào hình CDRom (lớn, nhỏ) hiện lên tooltip “Click Me”

* Hiệu chỉnh TabIndex 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;***

* Ần 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;***

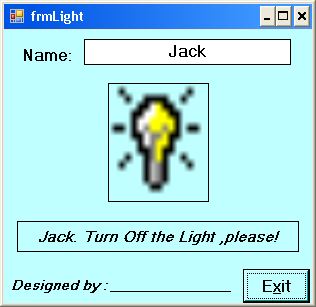
###### Khi chương trình chạy hòan chỉnh hãy hiệu chỉnh hai hình CDRom chồng lên nhau sao cho đồng tâm.

**Bài 7:**

**Mục đích:**

Kết hợp dử dụng component ToolTip trên các control

**Yêu cầu**



**Yêu cầu:**

1. Khi chương trình vừa thực thi.
   1. Form được hiển thị giữa màn hình
   2. 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”
   3. 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**

###### Hiện tên mặc định vào ô Name

* 1. Lable 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!”*"**;**

**}**

###### 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!”*"**;**

##### }

###### Nút Exit cho phép đóng ứng dụng

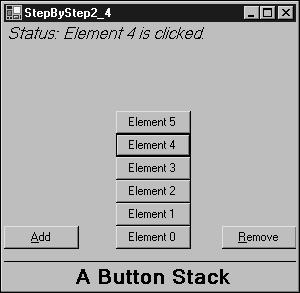
**Bài 8:** Adding Controls Programmatically

**Mục đích:**

Lập trình biến cố trên các đối tượng được tạo động lúc thực thi(run time)

**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 trang thái của Element đó lên một label ở phía trên form. (xem hình)

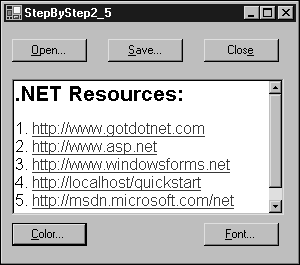
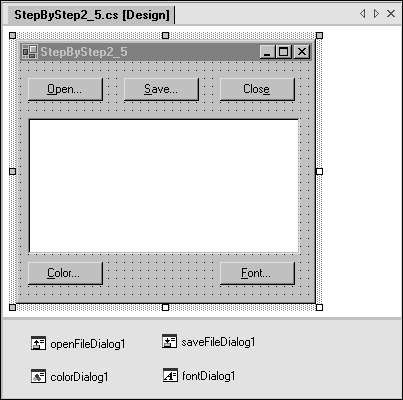
**Bài 9:**

##### Mục đích:

###### Sử dụng các dialog được cung cấp bởi C#. Sử dụng RichTextBox

**Yêu cầu**

Ex1: Using Common Dialog Boxes

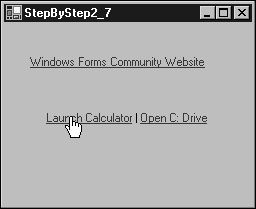


**Bài 10:**

##### Mục đích:

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

**Yêu cầu:** Using LinkLabel Controls



**Bài 11:**

**Mục đích:** Sử dụng GroupBox và Panel

**Yêu cầu :** Using GroupBox and Panel Controls



**Bài 12:**

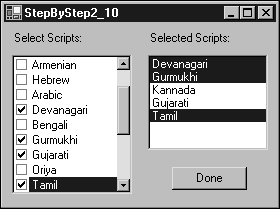
##### Mục đích:

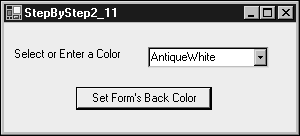
###### Sử dụng ListBox, CheckListBox, Combobox

**Yêu cầu**

Step by Step

Ex1: Using ListBox and CheckedListBox Controls



Ex2: Using ComboBox Controls

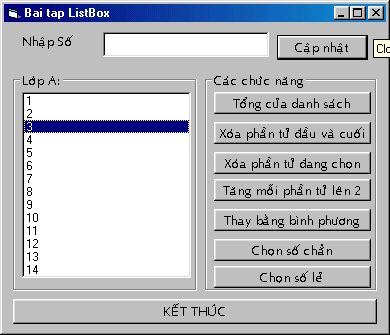
**Bài 13:**

**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 hiễn 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 mặt 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:**

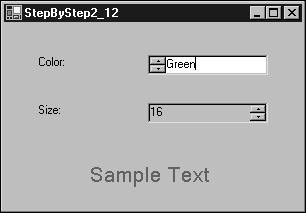
**Mục đích:**

Sử dụng các control của Visual Studio

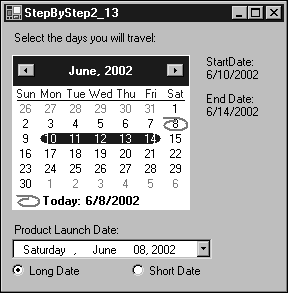
**Yêu cầu**

Step by step

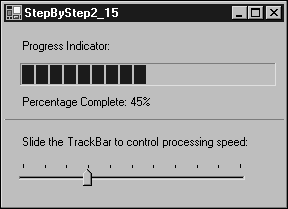
Ex1: Using DomainUpDown and NumericUpDown Controls



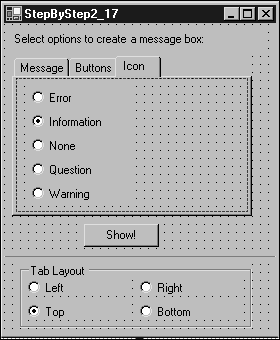
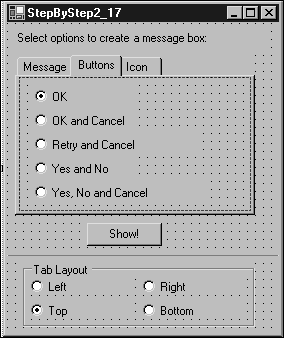
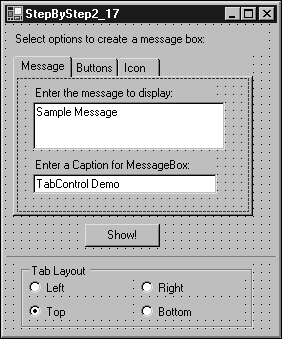
Ex2: The MonthCalendar and DateTimePicker Controls



Ex3:The Timer, TrackBar, and ProgressBar Controls



Ex4: Using TabControl Controls



**Bài 16:**

##### Mục đích:

###### Xây dựng hệ thống menu : MainMenu và ContextMenu(menu cảm ngữ cảnh)

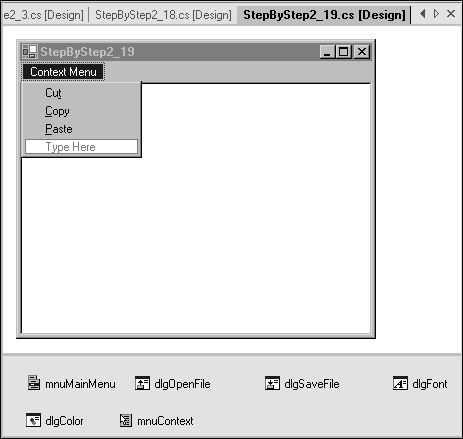
**Yêu cầu**

Step by Step

Ex: Creating a Main Menu for a Form



Ex : Creating a Context Menu for a Form



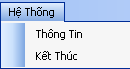
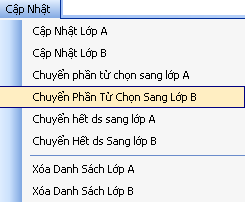
**Bài 17:**

**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 thức hiện chương trình.

##### Gợi ý:

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

**Bài 18:**

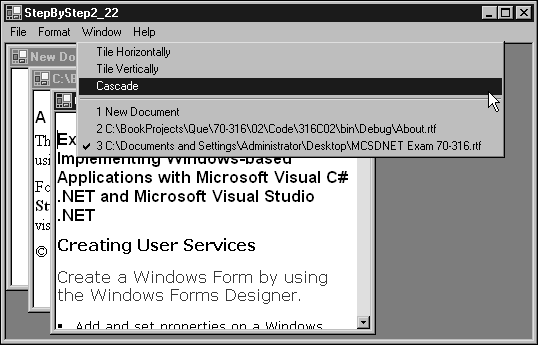
##### Mục đích:

###### 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:**

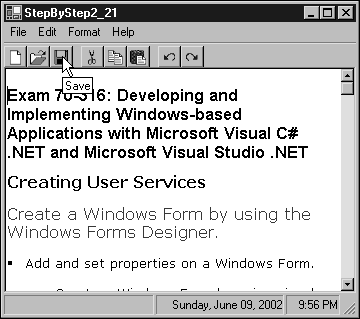
##### Mục đích:

###### Xây dựng ứng dụng kết hợp đối tượng statusBar

**Yêu cầu**

Step by step

Creating a Status Bar for a Form



**Bài 20:**

##### Mục đích:

###### Xây dựng ứng dụng liên kết nhiềm form

**Bài 21:**

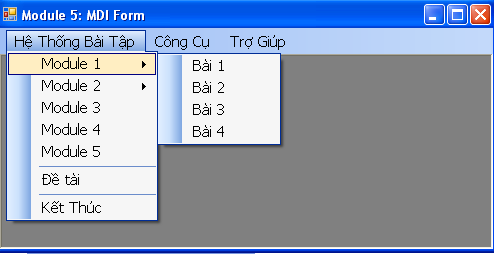
##### Mục đích:

###### 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

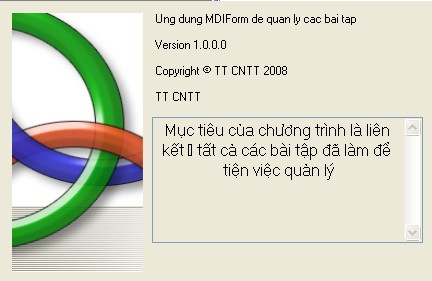
*Vào menu Project chọn <Add Existing Items> Chọn File cần Add vào chương trình (\*.cs)*

*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)

# Module 2

Nội dung kiến thức thực hành:

+ Các kiểu dữ liệu trên C#

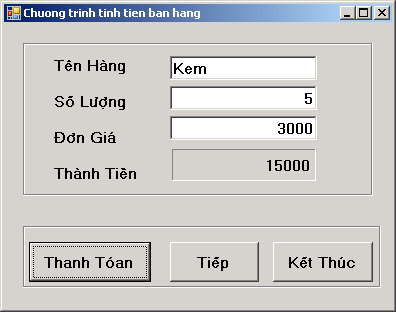
+ Cấu trúc điều khiển

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 hiễn thị giữa màn hình. (StartPosition =CenterScreen )
    - Thiết lập nút Thanh Tóan 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 Tóan”.
    - 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.

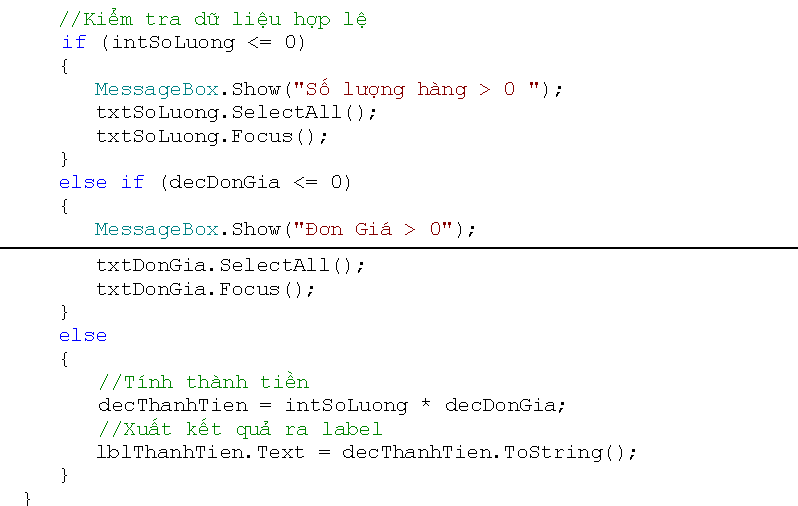
* + - Tính toán cho mặt hàng hiện tại xuất kết quả ra label Thành Tiền.

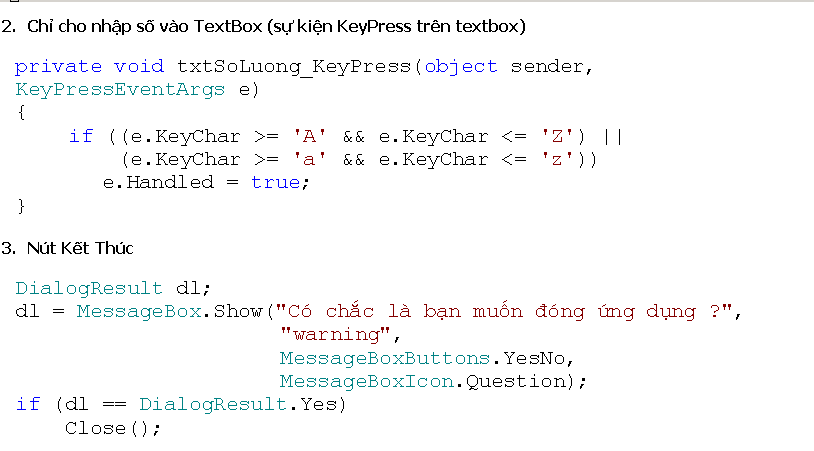
Thành tiền = số lượng \* đơn giá;

* **Nút Tiếp**:
  + Xóa nội dung các textbox và label
  + Đặt focus vào textbox đầu tiên
* Nút Kết Thúc:
  + Phát sinh Messagebox hỏi lại việc đóng form có chính xác không, nếu người sử dụng xác nhận bằng nút Yes thì cho thóat chương trình

##### Hướng Dẫn :







**Bài 2:**

##### 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 3:**

##### Mục đích:

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

**Yêu cầu**

Thiết kế giao diện như hình bên dưới.



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 lable Thành tiền

Thành tiền = Số lượng \* Đơn Giá ;

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

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

**Bài 4:**

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 bao bien int intNamDuong;

string strCan="";

string strChi="";

**//1. kiem tra khac rong**

**//2. Gan Gia tri**

**//3. kiem tra số dương**

//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 5:**

Chương trinh quay số may mắn



Thiết kế giao diện như hình trên

* Groupbox chứa 3 labe.
* 2 Label dùng để hiễn thị kết quả (canh lề pải)
* 3 button như hình vẽ.

Yêu cầu

**Khi chương trình vừa bắt đầu**

* 3 label sẽ hiễn 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 lược chơi mới hòan tòan (set lại tền ngươi 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();

lblSo3.Text = so3.ToString();

//kiểm tra số ngẫu nhiên vừa tạo ra có phải lá 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; tiennguoi = tiennguoi + 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

}

//hiễn thị kết quả lblTienNguoiChoi.Text = tiennguoi.ToString(); lblTienMay.Text = tienmay.ToString();

}

}

//xử lý nút quay số

private void btnQuaySo\_Click(object sender, EventArgs e)

{

if (tiennguoi >= 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

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

//hiễn 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**

}

}

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.

**Bài 6:**

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

**Bài 7:**

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 sụng 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 ( *giả sử giá vé là 100đ/ghế*)

* + Nếu click vào **nút HỦY BỎ (hoặc nhấn phín 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 8:**

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= rổng



* Khi người sử dụng Click vào ChechBox **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

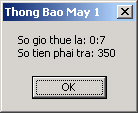
Cho thuộc tính Enabled của CheckBox Bắt đằu

=False

Cho thuộc tính Enabled của CheckBox Kết thúc

=True

o Khi người sử dụng Click vào ChechBox **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 Dựa vào Giờ và Phút (không cần quan tâm đến số giây) trên các label **Bắt đầu** và **Kết thúc** tính ra số tiến 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 đằu =True Cho thuộc tính Enabled của CheckBox Kết thúc =False Cho thuộc tính Value của CheckBox Bắt đằu =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 9:**

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



o Khi form vừa load lên màn hình thì

Các image trong khung máy 1 và máy 2 có picture là **Máy tính**

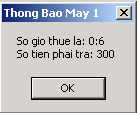
Các Label có Text = rổng

* 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 tiến 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 3**

Nội dung kiến thức thực hành:

+ Xử lý ngọai lệ (Error Handling for the User Interface).

+ Xử lý lỗi khi nhập liệu (VALIDATING USER INPUT)

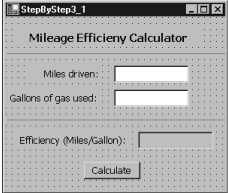
**Bài 1 :**

Mục đích:

Sử dụng lớp Exception đề 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 chặc chẽ các trường hợp sau:

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

- …

**Bài 3 :**

Mục đích:

Sử dụng lớp Exception đề bẩy lỗi chương trình. Sử dụng MessageBox để hiển thị thông báo lỗi

Yêu cầu:

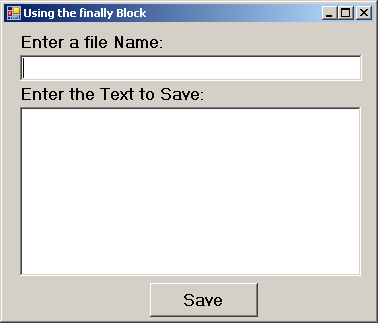
Viết lại bài 1, nhưng sử dụng đối tượng ErrorProvider để hiệ thị các thông báo lỗi ngay trên control

**Bài 4 :**

Mục đích:

Sử dụng lớp Exception đề 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:

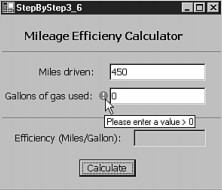
Chương trình phải thức hiện việc **đóng file bất kẻ có xảy ra lỗi hay không**.

**Bài 5:**

Mục đích:

Sử dụng Component ErrorProvider kết hợp kỷ 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 tóan, nhưng vẫn phải cho người dùng Click nút đóng form.

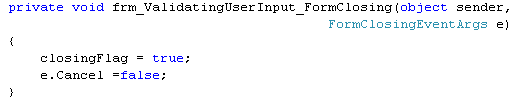
Hướng dẫn:

Thiết lập thuộc tính CausesValidation của Form là False

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

private bool closingFlag = false;

Trong sự kiện đóng form bổ sung code



Xem thêm giáo trình trang

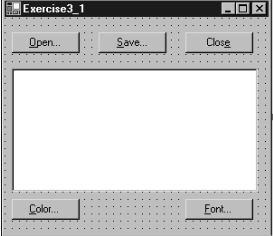
**Bài 6 :**

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”

**Bài 7 :**

Mục đích:

Sử dụng kỹ thuật Validating User Input.

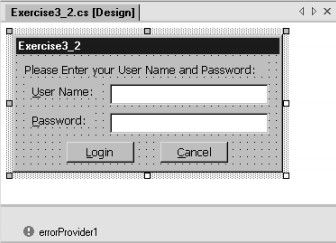
Yêu cầu:

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 khỏang trắng** nào bên trong. Ngược lại xuất thông báo lỗi cụ thể thông qua ErrorProvider.

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

Xem thêm hướng dẫn trong giáo trình trang



**Bài 8 :**

Mục đích:

Sử dụng lớp Exception đề 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 4**

Nội dung kiến thức thực hành:

+ windows form.

+ Sử dụng Mảng và danh sách

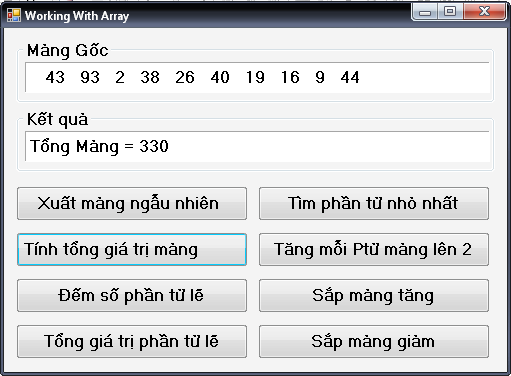
**Bài 1:**

Mục đích:

Sử dụng mảng một chiều

Yêu cầu:

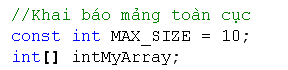
Thiết kế Form như hình sau:



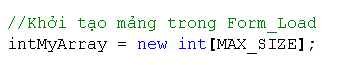
Trong đó 2 group box sẽ chứa 2 label, để hiễn thị mảng và kết quả xử lý trên các button tương ứng.

Hướng dẫn :

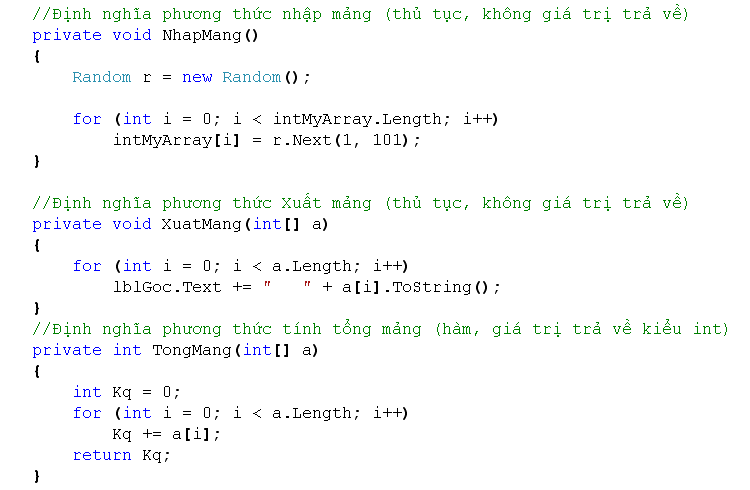
* 1. khai báo mảng tòan cục, với kích thước mảng MAX\_SIZE là 1 hằng số.



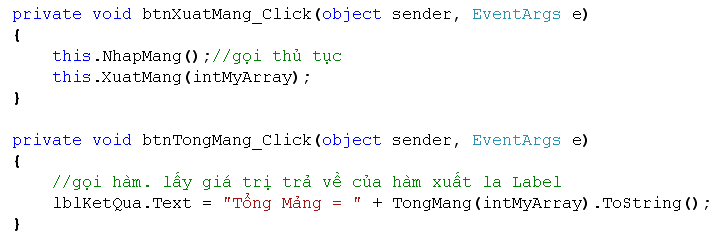
* 1. KHởi tạo mảng trong Form\_Load



* 1. Xây dựng các phương thức theo yêu cầu



* 1. Gọi sử dụng các phương thức tại các button.



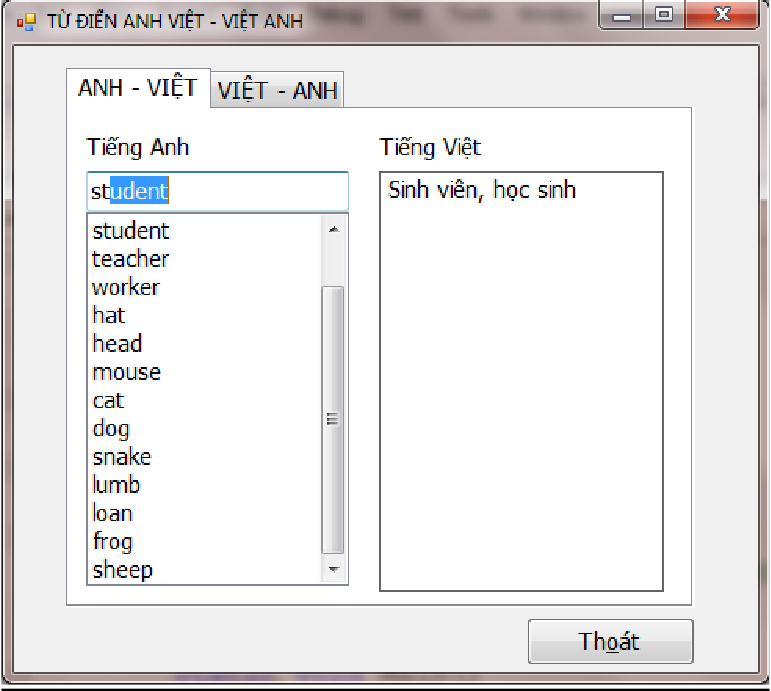
|  |  |  |  |
| --- | --- | --- | --- |
| **Bài** | **2:** |  |  |

Mục đích:

Sử dụng Arraylist,

Yêu cầu:

Xây dựng từ điễn 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 nút Enter hoặc Doubleclick vào từ cần tra thì nghĩa tương úng 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 5

Nội dung kiến thức thực hành:

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

**Bài 1:**



Yêu cầu:

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

Hướng dẫn:

1. 1. Tạo các field private:

*Tên hàng, số lượng, đon giá.*

1. Tạo các field static:

Số hóa đon, doanh thu.

1. Tạo các property thông thường.
2. Tạo property ReadOnly đối với các Field static.
3. Tạo các Contructor
4. Tạo các phương thức

**Bài 2:**

Yêu cầu:

1. Thiết kế Form như hình sau:



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

Hướng dẫn:

* + Tạo các field private:

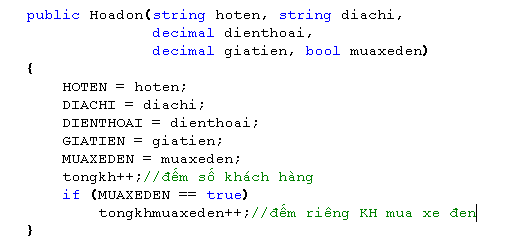
o Họ tên, địa chỉ, điện thoại, giá tiền, màu xe.

* Tạo các field static:

o Số khách hàng, số khách mua xe đen, doanh thu.

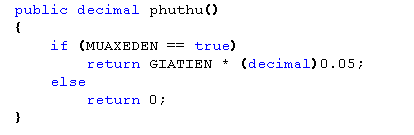
* Tạo các property thông thường.
* Tạo property ReadOnly đối với các Field static.
* Tạo các Contructor

Ví dụ:



* Xây dựng các phương thức xử lý tính toán.

Ví dụ:



# Module 6

Nội dung kiến thức thực hành:

Sử dụng các chuỗi (String)

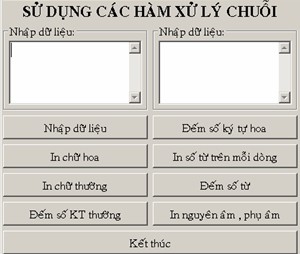
**Bài 1:**

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.

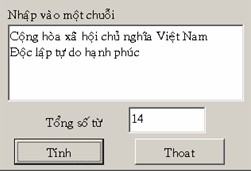
Khi nút nhập dữ liệu được ấn thì các nút lệnh khác được sáng lên xoá nội dung của Textbox nhập liệu và Setfocus cho Textbox này ( sử dụng phương thức Setfocus).

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 nhấn).



**Bài 2:**

Tạo màn hình đếm số từ như sau



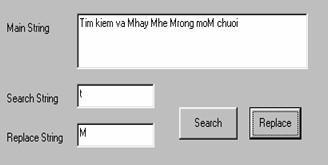
###### Dựa vào chuỗi nhập , đếm số từ đã nhập theo quy định các từ ngăn cách nhau bằng ký tự trắng (space).

**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 5:**

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

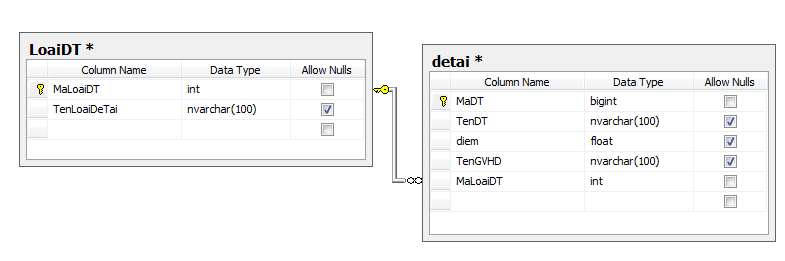
**Module 7**

**Bài 01:**

Sinh viên tạo thư mục trên đĩa D:\ để chứa bài thi theo đúng định dạng sau: **D:\SoThuTu\_HoVaTen**. Ví dụ: D:\01\_NguyenTatThang (Không gõ dấu tiếng Việt, không khoảng trắng).

* Tạo cơ dữ liệu với tên **detai** được tạo trên SQL Server 2005 lưu trong thư mục **SoThuTu\_HoVaTen**.

Gồm các bảng sau:



Yêu cầu:

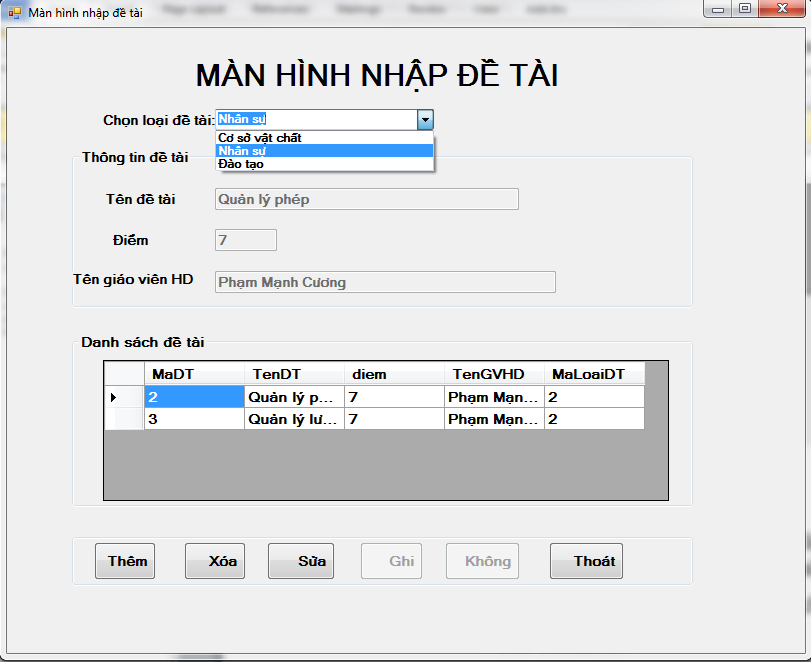
- MaDT: Tự động tăng

- Sinh viên tạo các thủ tục sau:

* spud\_ChonDeTaiTheoLoaiDeTai: Chọn đề tài theo loại đề tài
* spud\_ChonDeTaiTheoMaDeTai: Chọn đề tài theo mã đề tài.
* spud\_SuaDeTai: Sửa đề tài.
* spud\_ThemDeTai: Thêm đề tài mới.
* spud\_XoaDeTai: Xóa đề tài

*\*Sinh viên tự nhập dữ liệu mẫu, ít nhất 3 dòng dữ liệu.*

**-** Sinh viên tạo chương trình với tên **AppDetai** lưu trong thư mục **SoThuTu\_HoVaTen** như sau:



**Yêu cầu:**

*Lưu ý:* Form không yêu cầu phần kiểm tra kiểu dữ liệu khi nhập liệu.

1. **Phần khởi tạo màn hình**

* Khi chạy chương trình sẽ hiển thị dữ liệu vào comboBox “cbLoaiDeTai” có các dòng (Item): Theo dữ liệu bảng LoaiDT.
* Khi màn hình vừa được load, không cho phép người dùng thao tác.
* Các TextBox.
* Các nút: Ghi, Không.
* Gán dữ liệu dòng đầu tiên trong dataGridView lên phần thông tin đề tài.
* Khi click lên 1 dòng trên dataGridView “danh sách đề tài” gán dữ liệu dòng đang chọn trong dataGridView lên phần thông tin đề tài, nếu số đề tài trong từng loại đề tài là rỗng (tức dataGridview trống) thì xóa trống các TexBox.

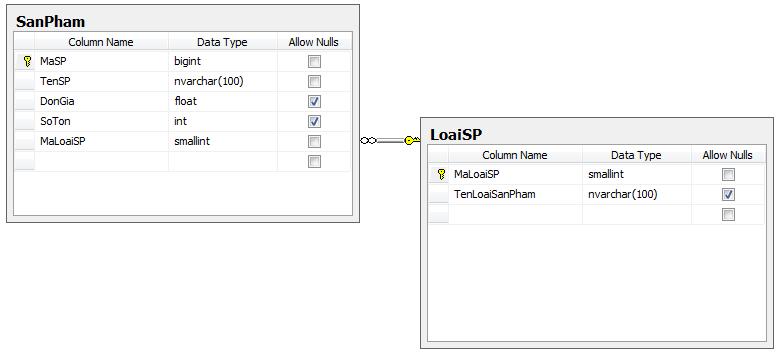
1. **Các xử lý**

* **Xử lý nút lệnh “Thêm”: Thêm mới đề tài.**
* Cho phép nhập thông tin trong các TextBox.
* Disable các nút lệnh “Thêm”, “Xóa”, “Sửa”, “Thoát”.
* Enable các nút lệnh “Ghi”, “Không”.
* **Xử lý nút lệnh “Xóa”: Hiển thị thông báo hỏi “Bạn có muốn xóa không?”**
* Nếu chọn “Yes” thì thực hiện xóa và load lại danh sách đề tài.
* Ngượi lại (chọn “No”): quay lại màn hình quản lý đề tài.
* Chỉ được xóa khi số đề tài trong từng loại đề tài >0.
* **Xử lý nút lệnh “Sửa”: Cho phép sửa thông tin đề tài.**
* Cho phép sửa thông tin trong các TextBox.
* Disable các nút lệnh “Thêm”, “Xóa”, “Sửa”, “Thoát”.
* Enable các nút lệnh “Ghi”, “Không”.
* Chỉ được xóa khi số đề tài trong từng loại đề tài >0.
* **Xử lý nút lệnh “Không”.**
* Hủy bỏ thao tác thêm/sửa thông tin đề tài.
* Hiển thị thông tin đề tài đã chọn trước đó.
* **Xử lý nút lệnh “Ghi”: Cập nhật CSDL (thêm/sửa) thông tin đề tài và load lại danh sách đề tài.**
* **Khi nhấn vào nút “Thoát” Kết thúc chương trình.**

**Bài 02:**

Tạo cơ dữ liệu với tên là **banhang** được tạo trên SQL Server 2005 lưu trong thư mục **SoThuTu\_HoVaTen** (*Cơ sở dữ liệu được dùng cho phần* ***B*** *và* ***C***).

Gồm các bảng sau:

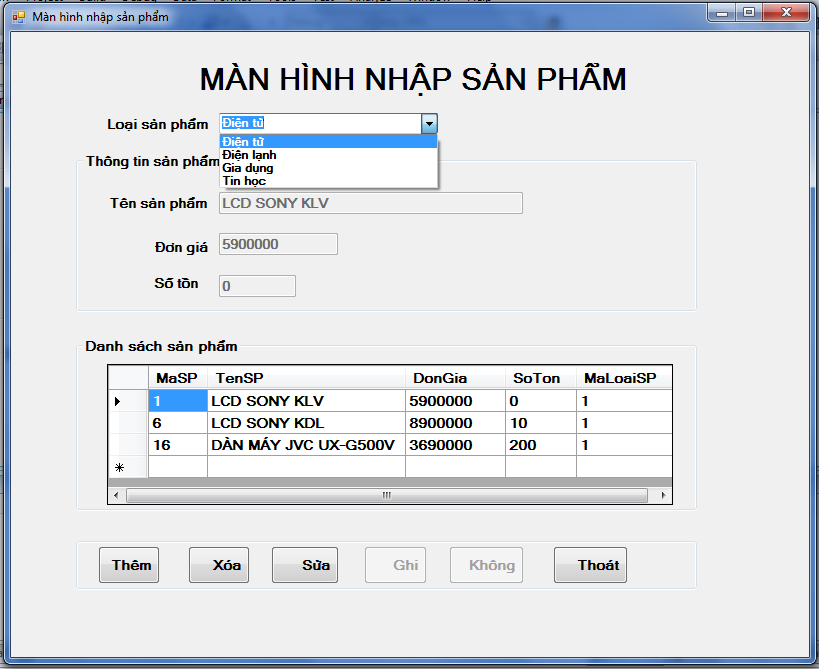
Yêu cầu:

- MaSP: Tự động tăng.

- Sinh viên tạo các thủ tục sau:

* spud\_ChonSanPhamTheoLoai: Chọn sản phẩm theo loại.
* spud\_ChonSanPhamTheoMaSP: Chọn sản phẩm theo mã sản phẩm.
* spud\_SuaSanPham: Sửa sản phẩm.
* spud\_ThemSanPham: Thêm sản phẩm mới.
* spud\_XoaSanPham: Xóa sản phẩm.

**-** Sinh viên tạo chương trình với tên **Appbanhang** lưu trong thư mục **SoThuTu\_HoVaTen** như sau:



**Yêu cầu:**

*Lưu ý:* Form không yêu cầu phần kiểm tra kiểu dữ liệu khi nhập liệu.

1. **Phần khởi tạo màn hình**

* Khi chạy chương trình sẽ hiển thị dữ liệu vào comboBox “cbLoaiSanPham” có các dòng (Item): Theo dữ liệu bảng Loại sản phẩm.
* Khi màn hình vừa được load, không cho phép người dùng thao tác.
* Các TextBox.
* Các nút: Ghi, Không.
* Gán dữ liệu dòng đầu tiên trong dataGridView lên phần thông tin sản phẩm.
* Khi click lên 1 dòng trên dataGridView “danh sách sản phẩm” gán dữ liệu dòng đang chọn trong dataGridView lên phần thông tin sản phẩm, nếu số sản phẩm trong từng loại sản phẩm là rỗng (tức dataGridview trống) thì xóa trống các TextBox.

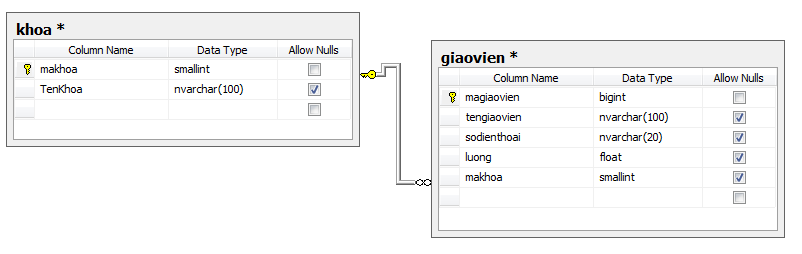
1. **Các xử lý.**

* **Xử lý nút lệnh “Thêm”: Thêm mới sản phẩm.**
* Cho phép nhập thông tin trong các TextBox.
* Disable các nút lệnh “Thêm”, “Xóa”, “Sửa”, “Thoát”.
* Enable các nút lệnh “Ghi”, “Không”.
* **Xử lý nút lệnh “Xóa”: Hiển thị thông báo hỏi “Bạn có muốn xóa không ?”**
* Nếu chọn “Yes” thì thực hiện xóa và load lại danh sách sản phẩm.
* Ngượi lại (chọn “No”): Quay lại màn hình nhập sản phẩm.
* Chỉ được xóa khi số sản phẩm trong từng loại sản phẩm >0.
* **Xử lý nút lệnh “Sửa”: Cho phép sửa thông tin sản phẩm.**
* Cho phép sửa thông tin trong các TextBox.
* Disable các nút lệnh “Thêm”, “Xóa”, “Sửa”, “Thoát”.
* Enable các nút lệnh “Ghi”, “Không”.
* Chỉ được sửa khi số sản phẩm trong từng loại sản phẩm >0.
* **Xử lý nút lệnh “Không”:**
* Hủy bỏ thao tác thêm/sửa thông tin sản phẩm.
* Hiển thị thông tin sản phẩm đã chọn trước đó.
* **Xử lý nút lệnh “Ghi”: Cập nhật CSDL (thêm/sửa) thông tin sản phẩm và load lại danh sách sản phẩm.**
* **Khi nhấn vào nút “Thoát” kết thúc chương trình.**

**Bài 03:**

Tạo cơ dữ liệu với tên là **daotao** được tạo trên SQL Server 2005 lưu trong thư mục **SoThuTu\_HoVaTen** (*Cơ sở dữ liệu được dùng cho phần* ***B*** *và* ***C***).

Gồm các bảng sau:



Yêu cầu:

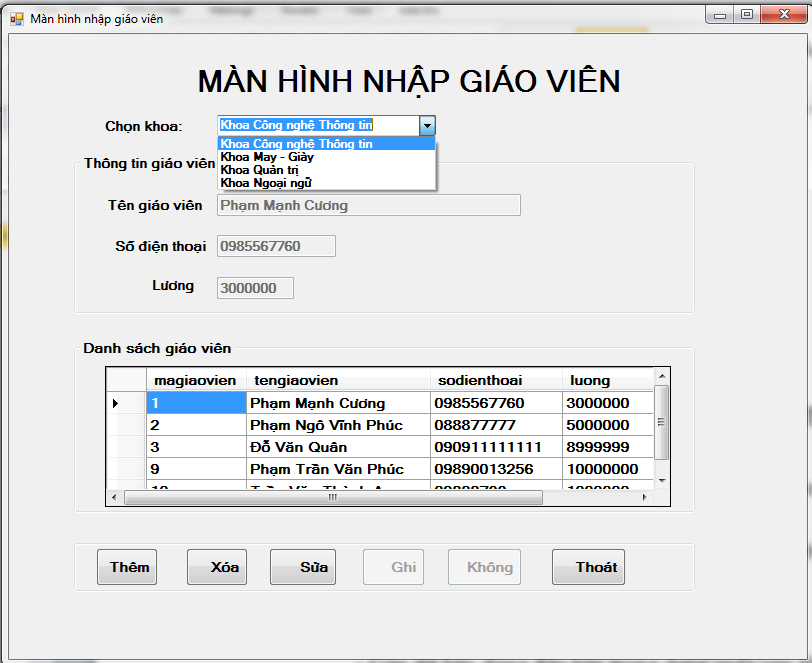
- magiaovien: Tự động tăng

- Sinh viên tạo các thủ tục sau:

* spud\_ChonGiaoVienTheoKhoa: Chọn giáo viên theo khoa.
* spud\_ChonGiaoVienTheoMaGiaoVien: Chọn giáo viên theo mã giáo viên.
* spud\_SuaGiaoVien: Sửa giáo viên.
* spud\_ThemGiaoVien: Thêm giáo viên mới.
* spud\_XoaGiaoVien: Xóa giáo viên.

*\*Sinh viên tự nhập dữ liệu mẫu, ít nhất 3 dòng dữ liệu.*

* Sinh viên tạo chương trình với tên **Appdaotao** lưu trong thư mục **SoThuTu\_HoVaTen** như sau:



**Yêu cầu:**

*Lưu ý:* Form không yêu cầu phần kiểm tra kiểu dữ liệu khi nhập liệu.

1. **Phần khởi tạo màn hình.**

* Khi chạy chương trình sẽ hiển thị dữ liệu vào comboBox “cbKhoa” có các dòng (Item): Theo dữ liệu bảng Khoa.
* Khi màn hình vừa được load, không cho phép người dùng thao tác .
* Các TextBox.
* Các nút: Ghi, Không.
* Gán dữ liệu dòng đầu tiên trong dataGridView lên phần thông tin giáo viên.
* Khi click lên 1 dòng trên dataGridview “danh sách giáo viên” gán dữ liệu dòng đang chọn trong dataGridView lên phần thông tin giáo viên, nếu số giáo viên trong từng khoa là rỗng (tức dataGridview trống) thì xóa trống các TextBox.

1. **Các xử lý.**

* **Xử lý nút lệnh “Thêm”: Thêm mới giáo viên.**
* Cho phép nhập thông tin trong các TextBox.
* Disable các nút lệnh “Thêm”, “Xóa”, “Sửa”, “Thoát”.
* Enable các nút lệnh “Ghi”, “Không”.
* **Xử lý nút lệnh “Xóa”: Hiển thị thông báo hỏi “Bạn có muốn xóa không ?”**
* Nếu chọn “Yes” thì thực hiện xóa và load lại danh sách giáo viên.
* Ngượi lại (chọn “No”): Quay lại màn hình nhập giáo viên.
* Chỉ được xóa khi số giáo viên trong từng khoa >0.
* **Xử lý nút lệnh “Sửa”: Cho phép sửa thông tin giáo viên.**
* Cho phép sửa thông tin trong các TextBox.
* Disable các nút lệnh “Thêm”, “Xóa”, “Sửa”, “Thoát”.
* Enable các nút lệnh “Ghi”, “Không”.
* Chỉ được sửa khi số giáo viên trong từng khoa >0.
* **Xử lý nút lệnh “Không”.**
* Hủy bỏ thao tác thêm/sửa thông tin giáo viên.
* Hiển thị thông tin giáo viên đã chọn trước đó.
* **Xử lý nút lệnh “Ghi”: Cập nhật CSDL (thêm/sửa) thông tin giáo viên và load lại danh sách giáo viên.**
* **Khi nhấn vào nút “Thoát” kết thúc chương trình.**

HẾT