

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT HƯNG YÊN

KHOA CÔNG NGHỆ THÔNG TIN

BỘ MÔN CÔNG NGHỆ PHẦN MỀM

-----o0o-----

BÀI TẬP THỰC HÀNH

Tên môn học: Lập trình nâng cao với C#

Số tin chỉ/ĐVHT: 3TC

Hệ đào tạo: Đại học chính qui

Ngành: KTPM

Khoa: Công nghệ thông tin

Họ tên giảng viên: **Nguyễn Hữu Đông**

Đơn vị công tác: Bộ môn CNPM – Khoa CNTT

Hưng yên, tháng 1 năm 2016

BÀI TẬP THỰC HÀNH 1: LẬP TRÌNH C# WIN FORM

Mục tiêu kiến thức:

- + Tạo ứng dụng trên Windows Form.
- + Sử dụng các thuộc tính, phương thức và sự kiện của các control: Label, Textbox, Button, ErrorProvider, RadioButton, CheckBox, GroupBox, Panel, PictureBox, ListBox, CheckedListBox, ComboBox.

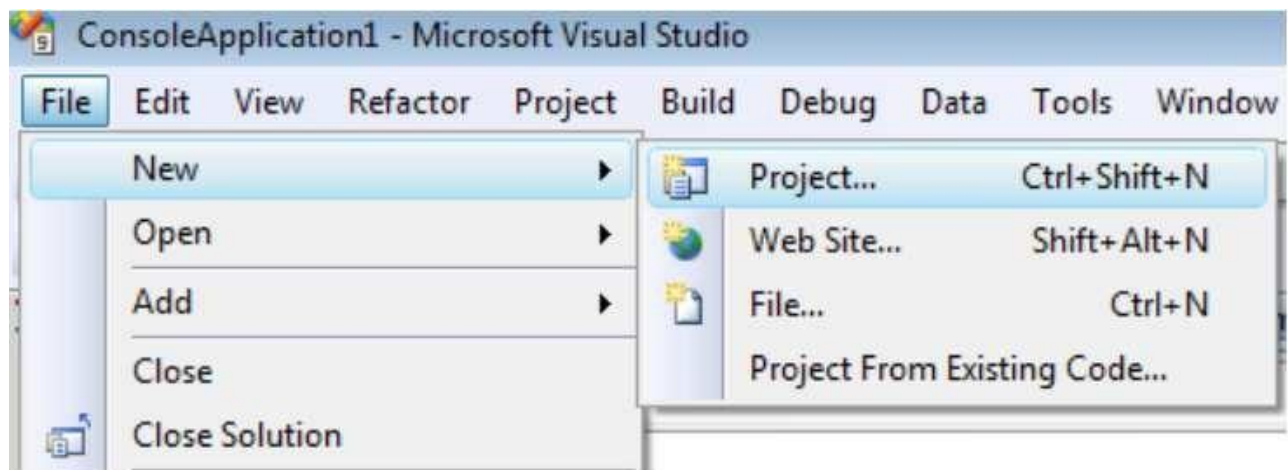
Mục tiêu kỹ năng:

- Có kỹ năng tốt trong thực hành xây dựng ứng dụng trên Winform với các điều khiển cơ bản.

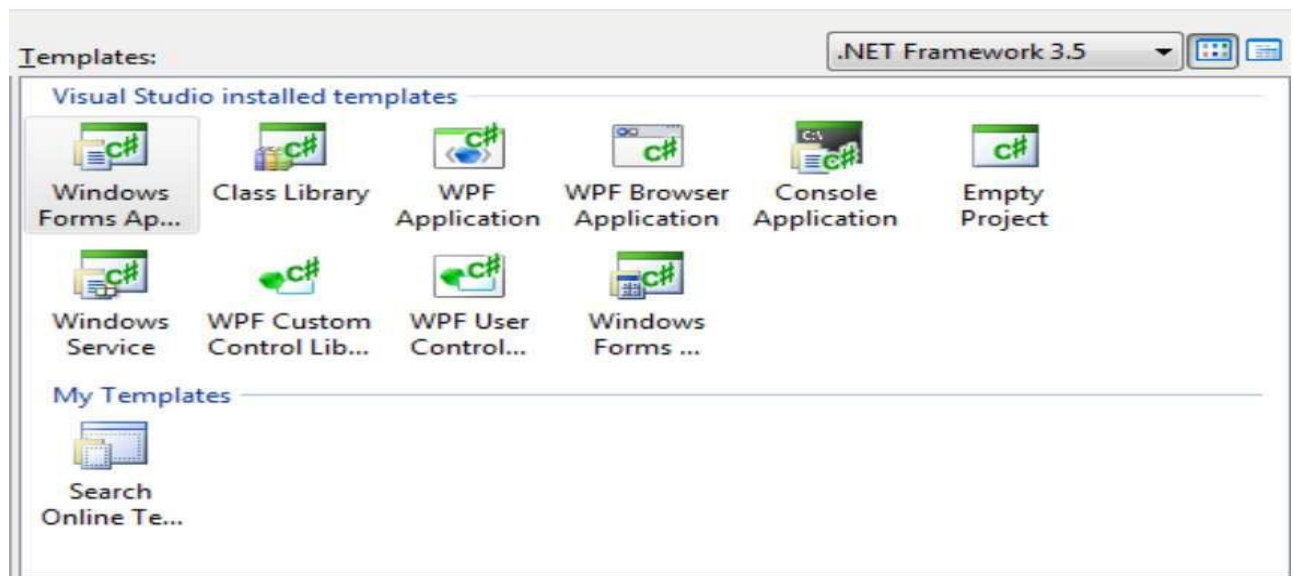
Bài tập mở đầu

Bước 1: Khởi tạo Project

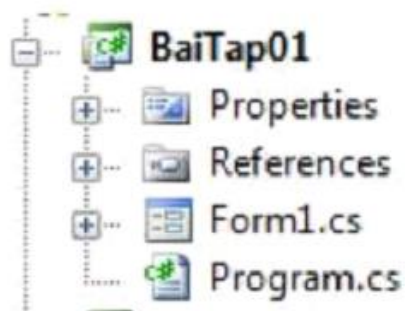
Từ Visual Studio 2005/2008/2010/... ta chọn File -> New -> Project.



Một hộp thoại xuất hiện, ta chọn Template là **Windows Form Application**.

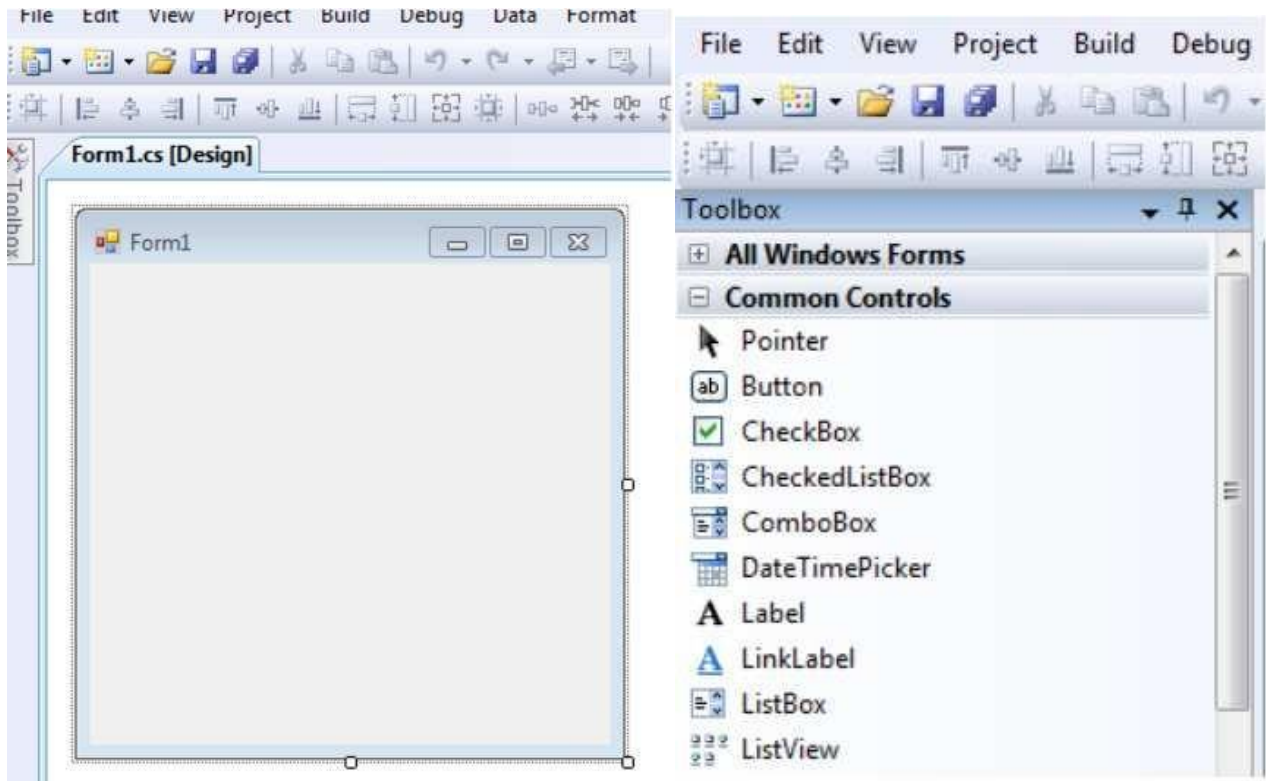


Ứng dụng được tạo ra như sau:



Bước 2: Ứng dụng mở đầu:

Project BaiTap01 gồm có thành phần Program.cs giống trong ứng dụng Console Application và còn có một thành phần mới là Form1.cs đang ở chế độ Design với giao diện như sau:

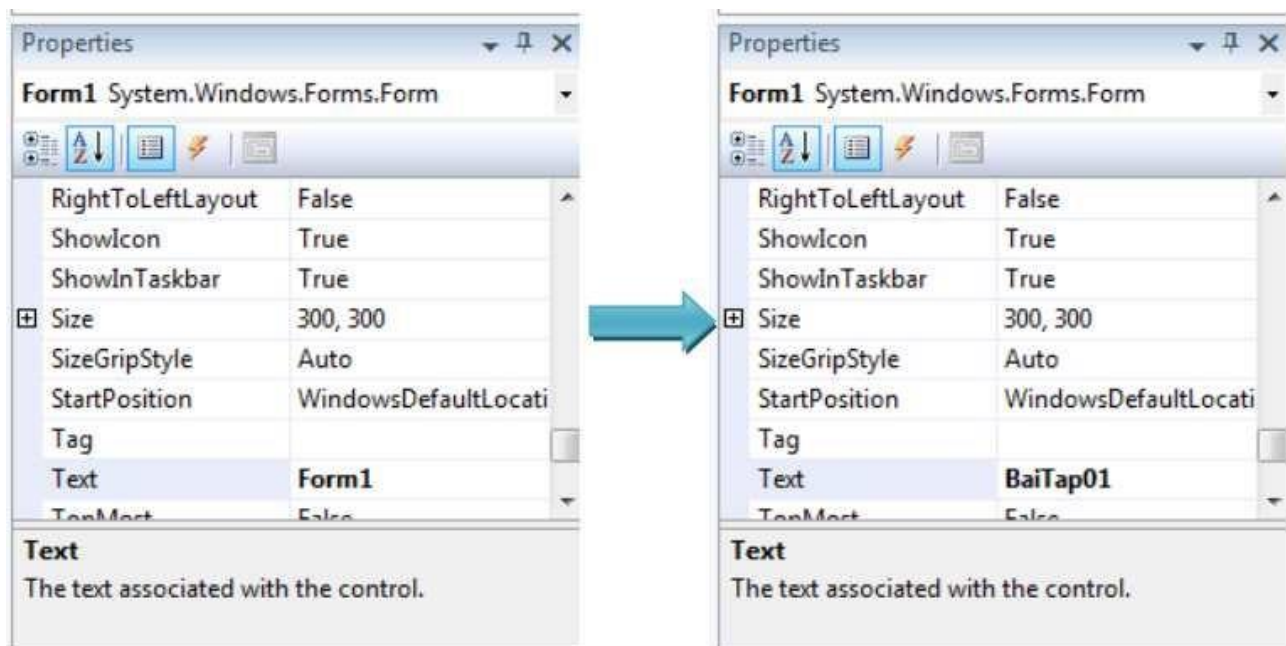


Chú ý cửa sổ Toolbox chứa các công cụ để thiết kế:

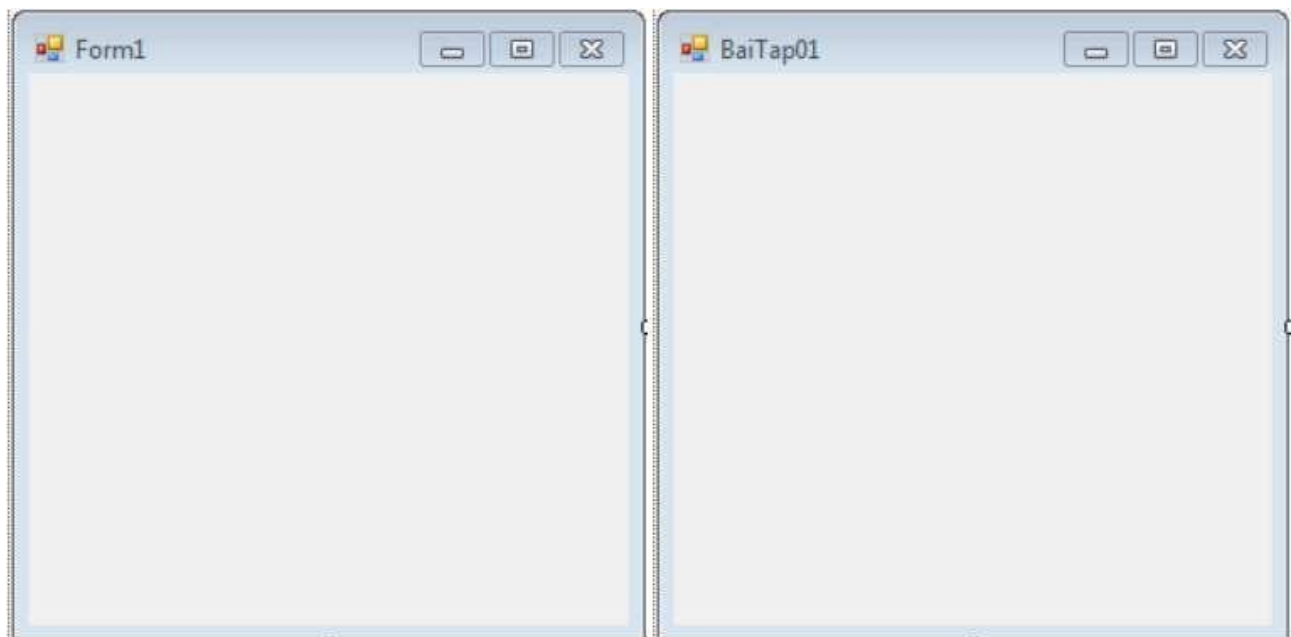
Nếu không thấy cửa sổ này, ta chọn menu View -> Toolbox.

Bước 3: Đổi tên form:

Click lên Form1 ở cửa sổ Design, quan sát trên cửa sổ Properties, ta thấy có thuộc tính Text, giá trị mặc định là Form1, ta đổi thành BaiTap01.

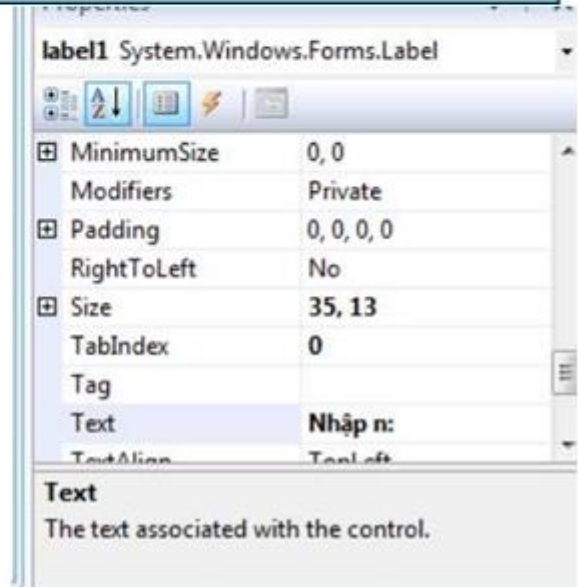
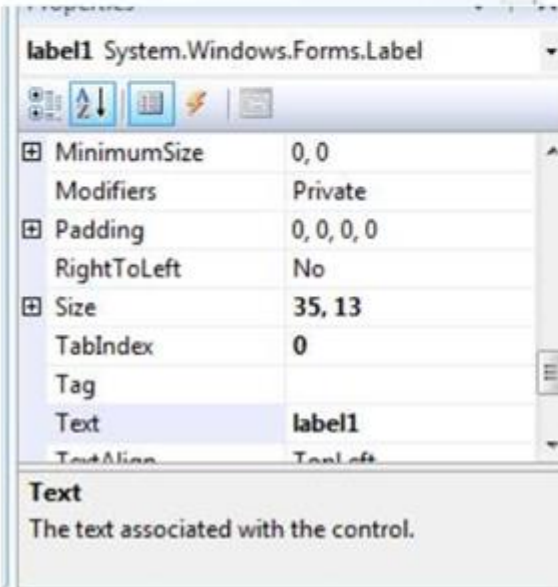


Ta thấy thanh tiêu đề Form1 đổi sang BaiTap01.

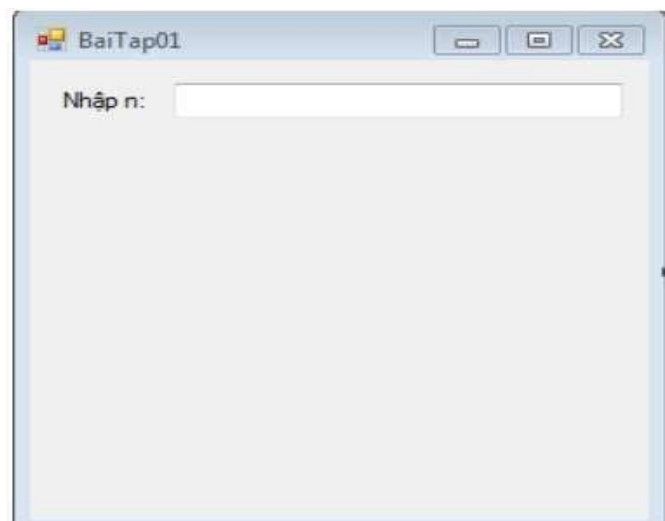
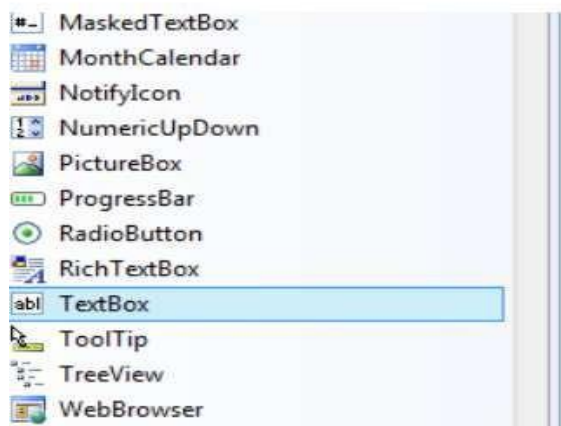


Bước 4: Kéo thả các control từ Toolbox vào Form

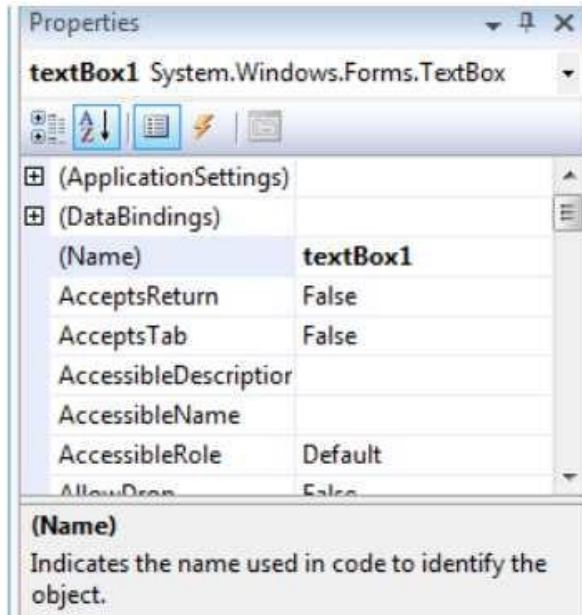
Từ thanh Toolbox, kéo thả một Label bỏ vào Form, sau đó di chuyển chuột đến cửa sổ Properties, chọn thuộc tính Text, đổi từ **label1** sang **Nhập n.**



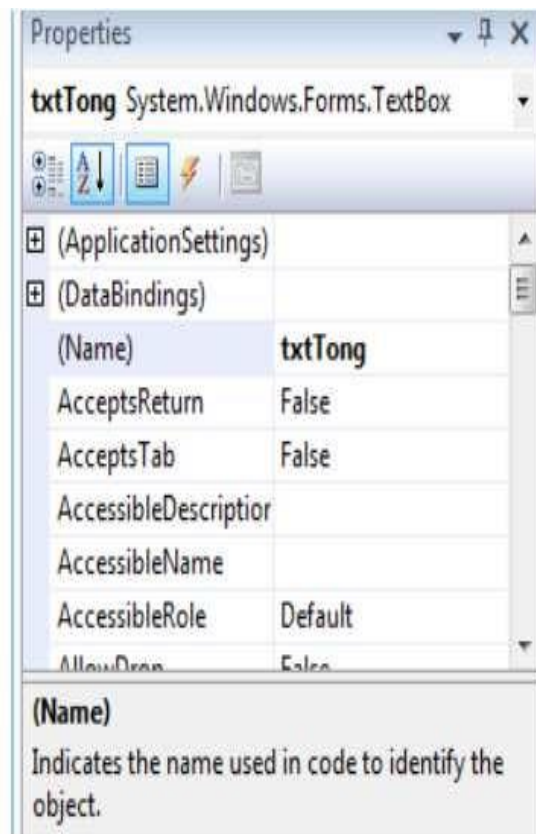
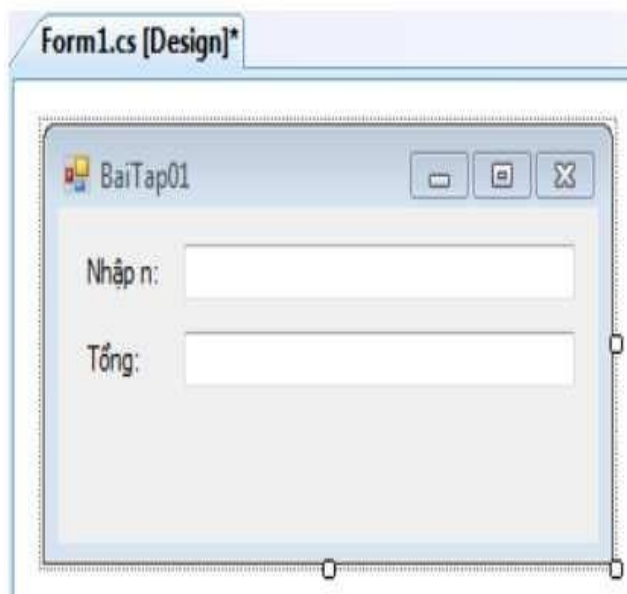
Tiếp theo kéo thả một Textbox bỏ vào form.



Trong cửa sổ Properties của textbox này, thuộc tính Name ta đổi từ **textBox1** thành **txtN**.

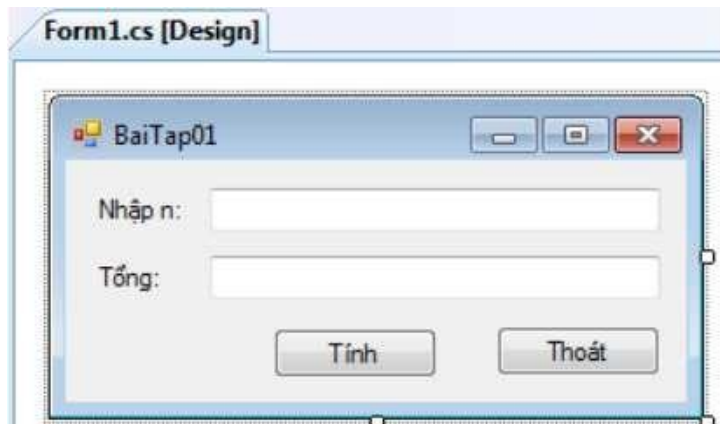


Tương tự thêm vào form một label có thuộc tính Text đổi thành **Tổng** và một textbox có thuộc tính Name đổi thành **txtTong**.



Tiếp theo, kéo một Button từ Toolbox và thả vào Form, đổi thuộc tính Name của button này là **btnTinh**, text là **Tính**.

Kéo thả thêm một Button nữa tương tự trên, Name = **btnThoat**, Text = **Thoát**.



Đưa chuột về các góc của Form để thay đổi kích thước cửa sổ.

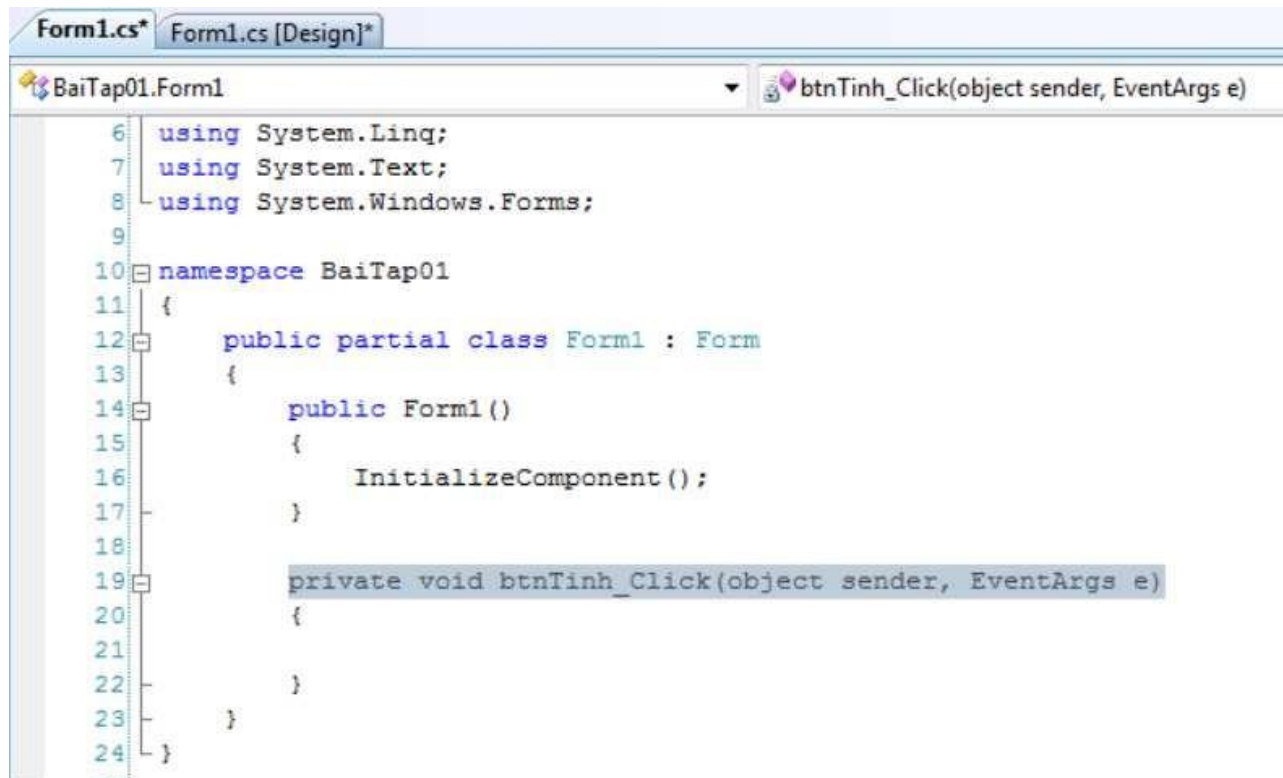
Bước 5: Xử lý sự kiện click chuột lên button

Double click vào button Tính (btnTinh), một cửa sổ soạn thảo xuất hiện với nội dung như sau:

```
6 using System.Linq;
7 using System.Text;
8 using System.Windows.Forms;
9
10 namespace BaiTap01
11 {
12     public partial class Form1 : Form
13     {
14         public Form1()
15         {
16             InitializeComponent();
17         }
18
19         private void btnTinh_Click(object sender, EventArgs e)
20         {
21         }
22     }
23 }
24 }
```

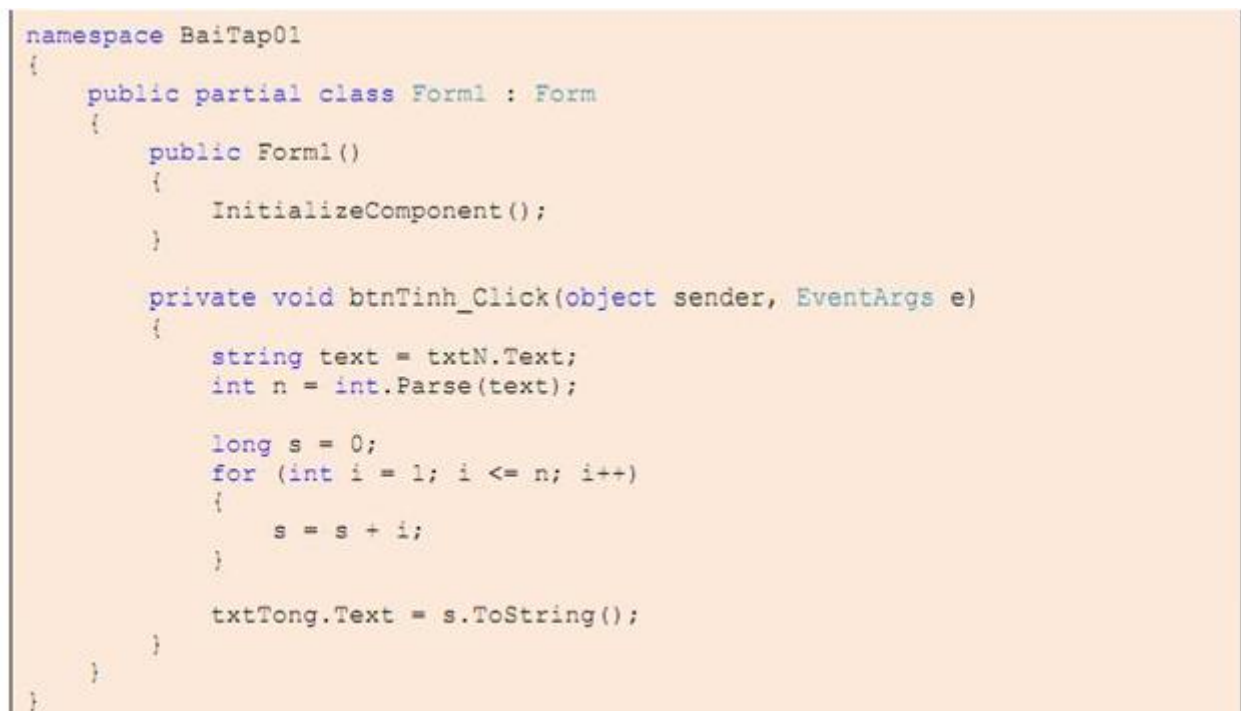
Chú ý: Cửa sổ này là Form1.cs tương ứng với Form1.cs[Design].

Phương thức btnTinh_Click được phát sinh do ta vừa double click vào btnTinh.



```
6 using System.Linq;
7 using System.Text;
8 using System.Windows.Forms;
9
10 namespace BaiTap01
11 {
12     public partial class Form1 : Form
13     {
14         public Form1()
15         {
16             InitializeComponent();
17         }
18
19         private void btnTinh_Click(object sender, EventArgs e)
20         {
21         }
22     }
23 }
24
```

Ta định nghĩa cho lớp này như sau:



```
namespace BaiTap01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnTinh_Click(object sender, EventArgs e)
        {
            string text = txtN.Text;
            int n = int.Parse(text);

            long s = 0;
            for (int i = 1; i <= n; i++)
            {
                s = s + i;
            }

            txtTong.Text = s.ToString();
        }
    }
}
```

Tiếp theo ta xử lý sự kiện click chuột cho button btnThoat.

Quay lại Form1.cs[Design], double click lên button Thoát, Form1.cs có nội dung thay đổi như sau:

```

namespace BaiTap01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnTinh_Click(object sender, EventArgs e)
        {
            string text = txtN.Text;
            int n = int.Parse(text);

            long s = 0;
            for (int i = 1; i <= n; i++)
            {
                s = s + i;
            }

            txtTong.Text = s.ToString();
        }

        private void btnThoat_Click(object sender, EventArgs e)
        {
        }
    }
}

```

Ta xử lý cho sự kiện này như sau:

```

private void btnThoat_Click(object sender, EventArgs e)
{
    Close();
}

```

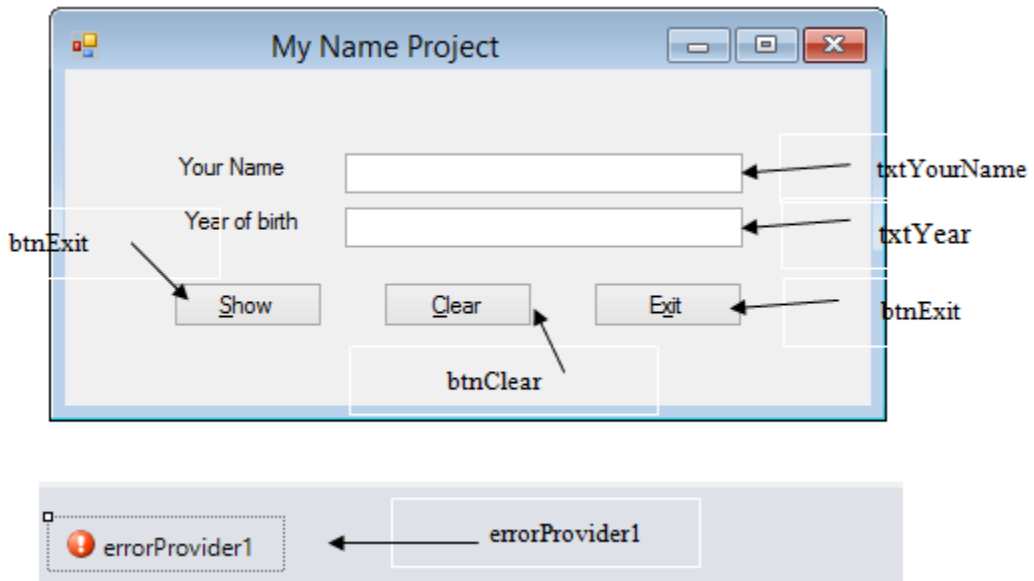
Chạy thử chương trình (F5):

Nhập $n = 4$ và click vào button Tính. Kết quả xuất hiện ở txtTong:



Click vào button Thoát để kết nối chương trình.

Bài 1: Thiết kế Form sau:



Yêu cầu:

- Chương trình cho phép nhập tên, năm sinh vào Textbox YourName và Year of birth tương ứng. Nếu YourName không nhập đủ liệu, Year of birth không phải là số thì phải thông báo lỗi (dùng ErrorProvider). Người dùng nhấn nút Show sẽ hiển thị thông tin nhập vào MessageBox bao gồm: tên, tuổi (năm hiện tại – năm sinh).

- Người dùng nhấn nút Clear sẽ xóa hết thông tin đã nhập trên các Textbox, đồng thời đặt con trỏ văn bản vào Textbox YourName.

- Nút Exit xác nhận người dùng có thực sự muốn thoát khỏi chương trình không? (Yes: thoát, No: không).

Hướng dẫn:

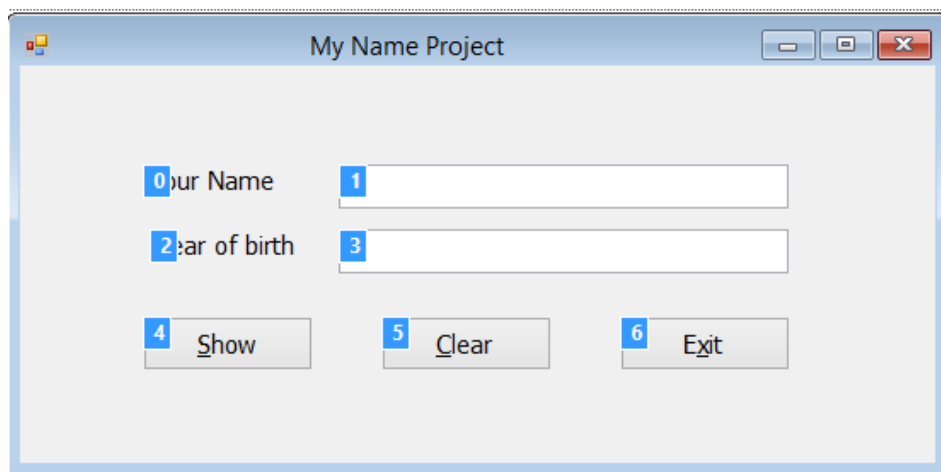
Danh sách các thuộc tính của các object:

Object	Properties	Events
frmMain	Name: frmMain Text: My name Project FontName: Tahoma FontSize: 11 AcceptButton: btnShow (nhận sự kiện click chuột khi nhấn Enter) CancelButton: btnExit (nhận sự kiện click chuột khi nhấn Esc)	FormClosing

txtYourName	Name: txtYourName BorderStyle: FixSingle	Leave (mất tiêu điểm)
TxtYear	Name: txtYear BorderStyle: FixSingle	TextChanged
btnShow	Name: btnShow Text: &Show	Click
btnClear	Name: btnClear Text: &Clear	Click
BtnExit	Name: btnExit Text: E&xit	Click
errorProvider	Name: errorProvider1	

Thứ tự nhận tiêu điểm trên Form: chọn menu View → Tab Order

Lần lượt thực hiện click chọn từng phần tử trên Form theo thứ tự nhận tiêu điểm:



Các sự kiện:

```
private void btnClear_Click(object sender, EventArgs e)
{
    txtYourName.Clear();
    txtYear.Clear();
    txtYourName.Focus();
}
```

```
private void btnShow_Click(object sender, EventArgs e)
{

```

```

        int age = DateTime.Now.Year - Convert.ToInt32(txtYear.Text);
        string s = "My name is: " + txtYourName.Text + "\n" +
age.ToString();
        MessageBox.Show(s);
    }

    private void txtYourName_Leave(object sender, EventArgs e)
    {
        Control ctr = (Control)sender;
        if (ctr.Text.Trim().Length == 0)
            this.errorProvider1.SetError(txtYourName, "You must enter
Your name");
        else
            this.errorProvider1.Clear();
    }

    private void txtYear_TextChanged(object sender, EventArgs e)
    {
        Control ctr = (Control)sender;
        if (ctr.Text.Trim().Length > 0 && !char.IsDigit(ctr.Text,
ctr.Text.Length - 1))
            this.errorProvider1.SetError(txtYear, "This is not invalid
number");
        else
            this.errorProvider1.Clear();
    }

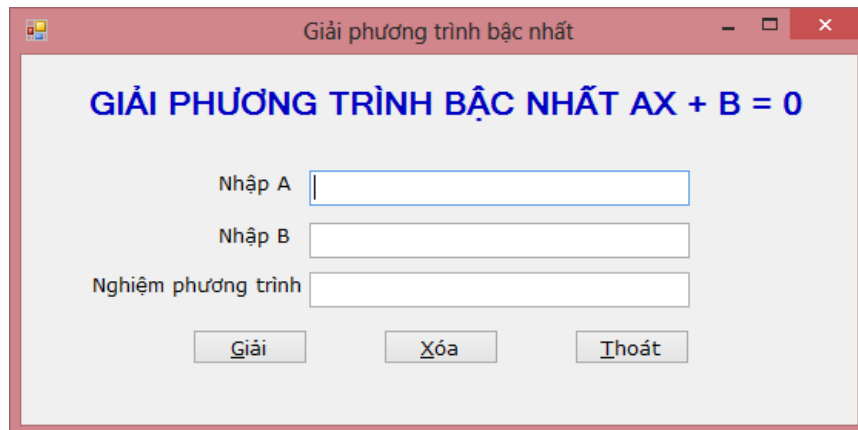
    private void frmMain_FormClosing(object sender,
FormClosingEventArgs e)
    {
        DialogResult r;
        r = MessageBox.Show("Do you want to close?", "Exit",
            MessageBoxButtons.YesNo,
            MessageBoxIcon.Question,
            MessageBoxDefaultButton.Button1);
        if (r == DialogResult.No)
            e.Cancel = true;
    }

```

```
}
```

```
private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Bài 2: Thiết kế chương trình hiện thực bài toán giải phương trình bậc 1:



Yêu cầu:

- Khi form hiện lên thì nút Giải và nút Xóa bị mờ (Enabled=false). Nếu dữ liệu nhập không hợp lệ thì thông báo lỗi (dùng errorProvider). Sau khi nhập dữ liệu hợp lệ và đầy đủ thì nút Giải có tác dụng (Enabled=true).

- Khi nhấn nút Tính: tính nghiệm phương trình (xét tất cả các trường hợp xảy ra: PT có 1 nghiệm, vô nghiệm, vô số nghiệm) và hiện kết quả vào Nghiệm PT. Khi đó nút Xóa có tác dụng, nút Tính bị mờ.

- Khi nhấn nút Xóa: xóa các Textbox và Label, đặt con trỏ vào Textbox A, nút Xóa bị mờ

- Khi nhấn nút Thoát: xác nhận người dùng có chắc chắn thoát khỏi ứng dụng hay không?

Bài 3: Thiết kế giao diện như hình. Khi nhấn chọn vào phép tính nào thì sẽ hiện kết quả của phép tính đó vào ô Kết quả.

Trước khi tính cần kiểm tra dữ liệu nhập phải là số.

Bài 4: Thiết kế giao diện như hình sau:

Yêu cầu:

- Khi chương trình hiện lên:
 - Radiobutton Red được chọn mặc định (đổi màu chữ ô lblLapTrinh và ô txtNhapTen)
 - Con trỏ văn bản xuất hiện ngay tại ô txtNhapTen.
- Khi gõ vào ô txtNhapTen thì Label lblLapTrinh chạy song song cùng nội dung.
- Nhấn nút "Thoát" hoặc Esc thì thoát chương trình.

- Nhấn Radiobutton Red, Green, Blue, Black thì đổi màu chữ tương ứng trong ô lblLapTrinh và ô txtNhapten.

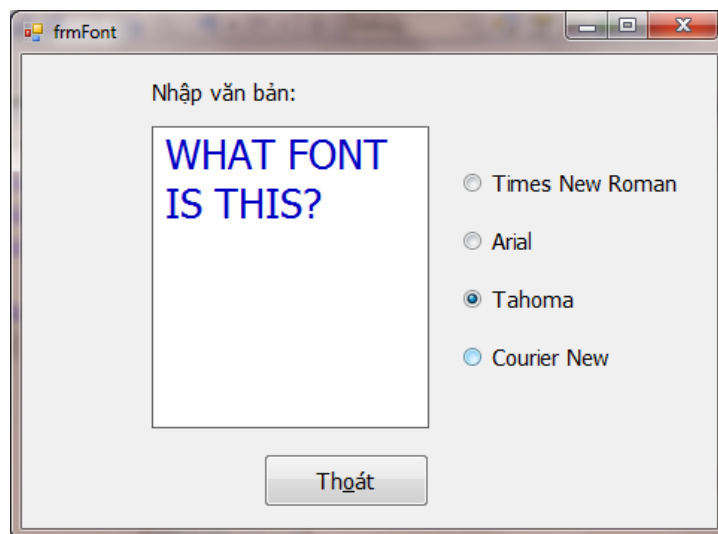
- Nhấn các checkbox chữ đậm, nghiêng, gạch chân thì đổi style chữ trong ô lblLapTrinh và ô txtNhapten tương ứng.

Hướng dẫn: đổi style chữ trong ô lblLapTrinh

Viết trong sự kiện CheckedChanged của từng checkbox:

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

Bài 5: Viết chương trình định dạng Textbox theo các font chữ tương ứng với từng Radiobutton



Bài 6: Viết chương trình hiển thị vào PictureBox lá cờ tương ứng với nước được chọn trên Radio Button tương ứng.



BÀI TẬP THỰC HÀNH 2: LẬP TRÌNH C# WIN FORM(tiếp)

Mục tiêu kiến thức:

+ Tạo ứng dụng trên Windows Form.

+ Sử dụng các thuộc tính, phương thức và sự kiện của các control: Label, Textbox, Button, ErrorProvider, RadioButton, CheckBox, GroupBox, Panel, PictureBox, ListBox, CheckedListBox, ComboBox.

Mục tiêu kỹ năng:

Có kỹ năng tốt trong thực hành xây dựng ứng dụng trên Winform với các điều khiển cơ bản.

Bài 1. Thiết kế giao diện như sau:

The screenshot shows a Windows Form titled "Combobox". It contains the following elements:

- Nhập Số** (Input Number): A group box containing a text box with the value "20" and a button labeled "Cập nhật" (Update).
- Danh sách các ước số** (List of divisors): A list box containing the numbers 1, 2, 4, 5, and 10.
- Tổng các ước số** (Sum of divisors): A button.
- Số lượng các ước số chẵn** (Number of even divisors): A button.
- Số lượng các ước số nguyên tố** (Number of prime divisors): A button.
- Thoát** (Exit): A button.

Yêu cầu:

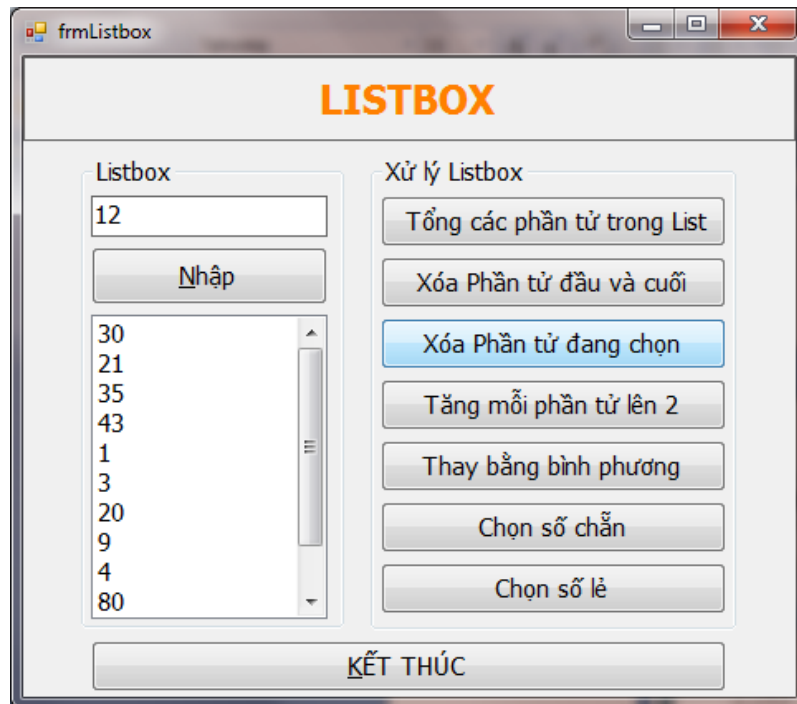
Khi Form vừa hiện lên, các Textbox, Combobox, Listbox chưa có dữ liệu, con trỏ đặt tại Textbox (thiết lập Tab Order hợp lý).

Nhấn nút "Cập nhật" hoặc Enter: thêm số vừa nhập ở Textbox vào Combobox (nhớ kiểm tra dữ liệu nhập), đồng thời xóa nội dung Textbox và đặt con trỏ lại Textbox.

Khi chọn 1 số trên Combobox thì danh sách các ước số của số này sẽ hiển thị vào Listbox bên phải tương ứng.

Khi nhấn các nút: "Tổng các ước số", "Số lượng các ước số chẵn", "Số lượng các ước số nguyên tố" thì sẽ hiển thị thông tin tương ứng vào MessageBox dựa vào các ước số trên Listbox.

Bài 2. Thiết kế giao diện như sau:



Yêu cầu:

Khi Form vừa hiện lên, các Textbox, Listbox để trống, con trỏ đặt tại Textbox (thiết lập Tab Order hợp lý).

Khi người sử dụng nhập một số vào Textbox rồi Enter hoặc nhấn nút "Nhập" thì số đó được thêm vào Listbox, đồng thời nội dung trong Textbox bị xóa và con trỏ được chuyển về Textbox.

Người dùng nhấn vào nút nào thì thực hiện chức năng tương ứng của nút đó. Hiện kết quả ra MessageBox (nếu có).

Thiết lập thuộc tính Anchor hợp lý cho các control.

Thiết lập MinimumSize cho form.

Bài 3. Viết chương trình nhập danh sách sinh viên theo yêu cầu sau: (xem hình bên dưới).

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, Ctrl và chuột)

Khi người dùng nhập Họ và tên của sinh viên vào Textbox, click nút Cập Nhật (hoặc Enter) thì tên sinh viên đó sẽ được đưa vào danh sách lớp A (không chấp nhận dữ liệu rỗng).

>, < chuyển các tên đang chọn từ Listbox trái sang Listbox phải và ngược lại.

>>, << chuyển hết toàn bộ các tên từ Listbox trái sang Listbox phải và ngược lại.

Nút Xóa: cho phép xóa các tên đang chọn trong danh sách lớp A.

Thêm vào giao diện 1 combobox Lớp, trong đó có 2 lớp: Lớp A, Lớp B, theo đó người sử dụng có thể chọn lớp để cập nhật sinh viên vào lớp mong muốn.

Hướng dẫn:

// copy selected strings in the source list to the destination list

```
for (int i = 0; i < SourceListbox.SelectedItems.Count; i++)
```

```
{
```

```
    DestinationListbox.Items.Add(SourceListbox.SelectedItems[i]);
```

```
}
```

// remove selected strings from the source list

```
for(int j = SourceListbox.SelectedItems.Count - 1; j >= 0; j --)
```

```
{
```

```
    SourceListbox.Items.Remove(SourceListbox.SelectedItems[j]);
```

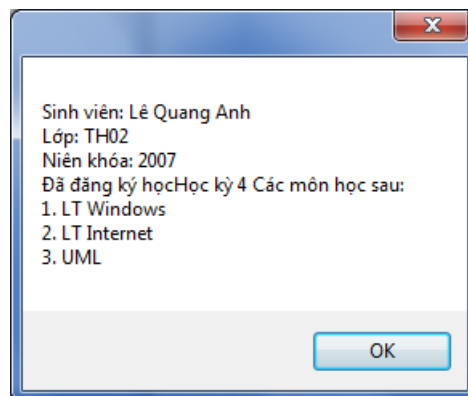
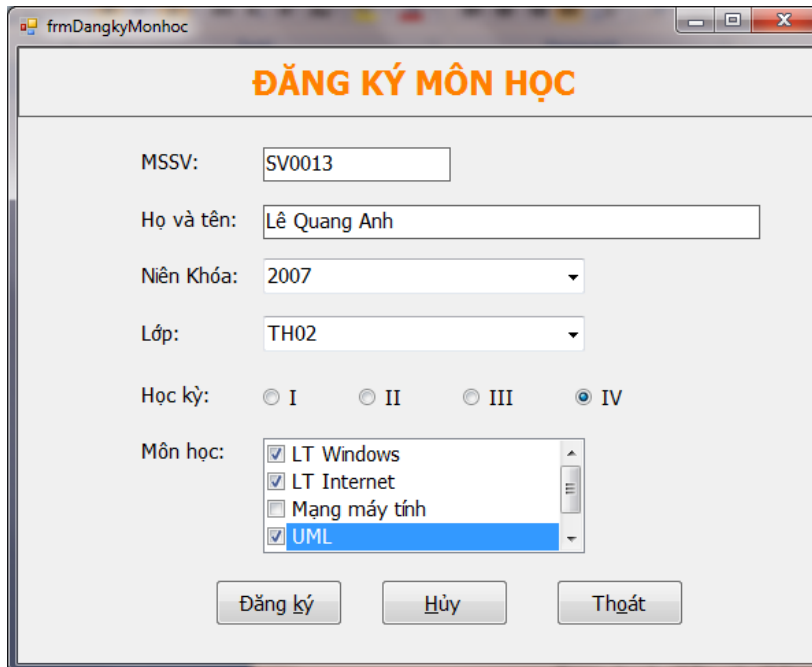
```
}
```


Bài 4. Viết chương trình cho phép sinh viên đăng ký học các môn học trong học kỳ:

Yêu cầu:

Khi Form hiện lên, các ô nhập đều để trống (thiết lập tab hợp lý).

Nút Đăng ký: Hiện thị các thông tin mà sinh viên đã đăng ký lên MessageBox như hình:



Nút Hủy: trả lại trạng thái ban đầu của Form.

Nút Thoát: thoát khỏi ứng dụng.

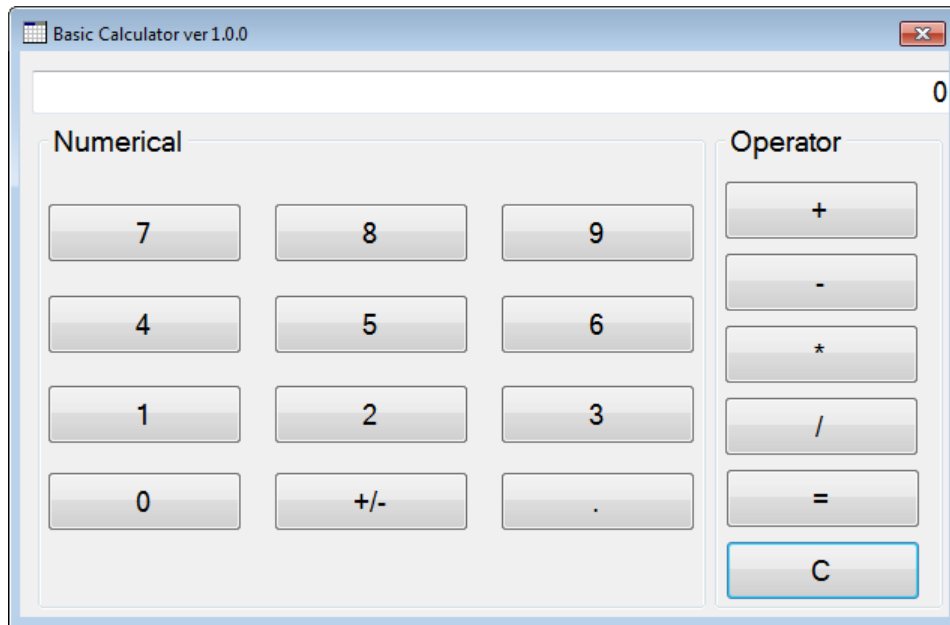
Bài 5. Thiết kế giao diện cho phép đổi màu Panel như sau:



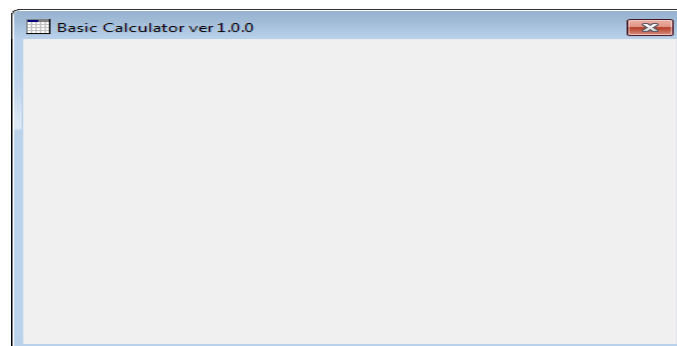
Để đổi màu Panel, người dùng có thể kéo các TrackBar để thay đổi các giá trị màu red(R), green(G) và blue(B). Yêu cầu khi TrackBar nào kéo đến đầu thì giá trị của nó phải được hiện ra tương ứng trên các Label. Các TrackBar có giá trị trong khoảng 0 đến 255.

Bài 6: Máy tính đơn giản thực hiện các chức năng +, -, *, /.

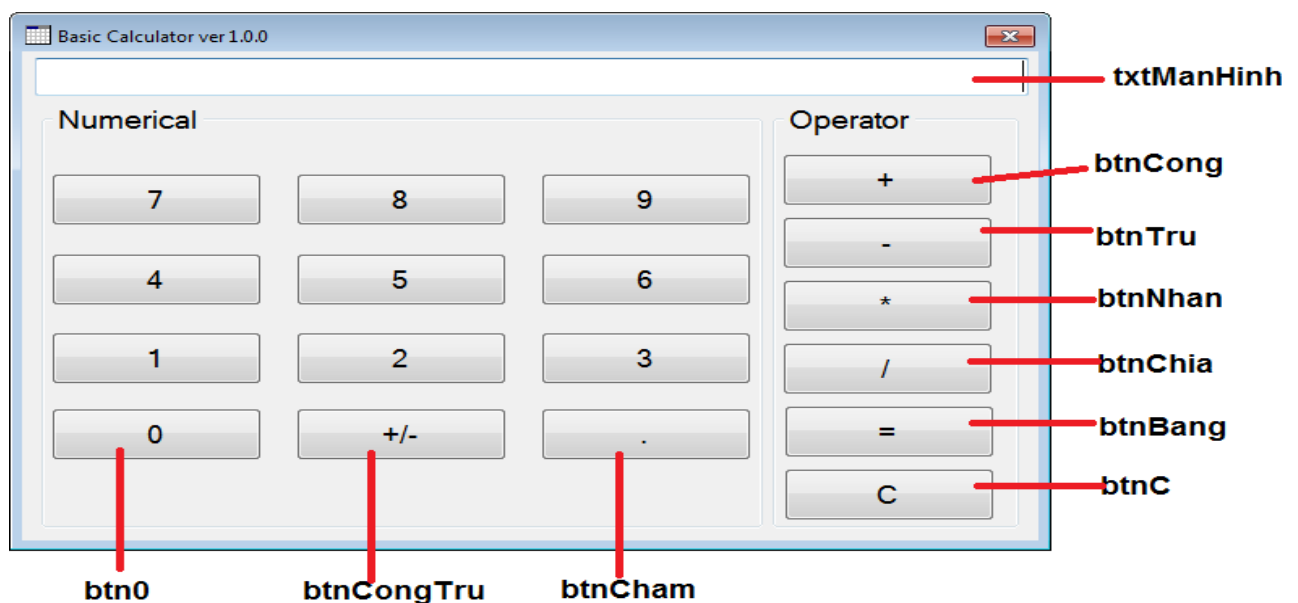
MÁY TÍNH ĐƠN GIẢN



Bước 1: Tạo Project, ghi tiêu đề form, icon form, ẩn nút Minimize, Maximize



Bước 2: Thiết kế màn hình và đặt tên như hình sau:



Trong đó các phím số đặt tên từ btn0 -> btn9.

Bước 3: Khai báo các biến cần thiết

- `string` chuoitam = ""; //Dùng để lưu giữ tạm giá trị của số nhập vào
- `char` toantu; //Dùng lưu kí hiệu toán tử của phép toán, vd: +, -, *, /
- `double[]` toanhang = `new double[2]`; //Lưu 2 toán hạng (số hạng) của phép toán
- `double` ketqua; //Dùng lưu kết quả của phép toán
- `int` buoc = 1; //Dùng lưu giữ số bước thực hiện phép toán

//Khai báo biến

```
string chuoitam = "";
char toantu;

double[] toanhang = new double[2];
double ketqua;

int buoc = 1;
```

Bước 4: Viết hàm cho chung cho các phím số (0 €9)

Đặt tên cho phím vừa bấm là b. Nếu b rỗng, kết thúc hàm. Ngoài ra chúng ta cũng loại trừ trường hợp có rất nhiều số 0 đứng trước một số khác (ví dụ 000000001 chính là 1). Do đó `b.Text == "0" && chuoitam.Length == 0` thì kết thúc hàm.

//Hàm bấm phím số

```
private void NumberButtons(object sender, EventArgs e)
{
    Button b = sender as Button;
    if ((b == null) || (b.Text == "0" && chuoitam.Length == 0)) return;
    chuoitam += b.Text;

    txtManHinh.Text = chuoitam;
}
```

Lưu ý: Cho tất cả các phím số gọi hàm này ứng với sự kiện click.

Bước 5: Viết hàm xử lý cho nút +/- (dương/âm)

//Hàm xử lý cho nút +/- (dương/âm)

```
private void btnCongTru_Click(object sender, EventArgs e)
{
    if (chuoitam.Contains('-'))
        chuoitam = chuoitam.Replace("-", "");
    else
        chuoitam = "-" + chuoitam;
    txtManHinh.Text = chuoitam;
}
```

Bước 6: Viết hàm xử lý cho phím .

//Hàm xử lý phím chấm

```
private void btnCham_Click(object sender, EventArgs e)
{
    if (!chuoitam.Contains('.'))
    {
        chuoitam = chuoitam + ".";
        txtManHinh.Text = chuoitam;
    }
}
```

Bước 7: Viết hàm xử lý cho nút C

//Hàm xử lý nút C

```
private void btnC_Click(object sender, EventArgs e)
{
    if (chuoitam.Length == 0)
    {
        buoc = 1;
        toanhang[0] = toanhang[1] = 0.0;
        toantu = ' ';
        ketqua = 0.0;
    }
    else
        chuoitam = "0";
    txtManHinh.Text = chuoitam;
}
```

Bước 8: Viết hàm xử lý cho phím bằng

//Hàm bấm phím số

```
private void btnBang_Click(object sender, EventArgs e)
{
    if (chuoitam.Length != 0)
        toanhang[1] = Double.Parse(chuoitam);
    switch (toantu)
    {
        case '+': ketqua = toanhang[0] + toanhang[1]; break;
        case '-': ketqua = toanhang[0] - toanhang[1]; break;
        case '*': ketqua = toanhang[0] * toanhang[1]; break;
        case '/': ketqua = toanhang[0] / toanhang[1]; break;
    }
    txtManHinh.Text = ketqua.ToString();
    //Gán lại giá trị ban đầu
    buoc = 1;
    chuoitam = "";
}
```

Bước 9: Viết hàm xử lý các phím thực hiện phép toán

//Hàm bấm phím số

```
private void Operations(object sender, EventArgs e)
{
    Button b = sender as Button;
    if(buoc == 1)
        toantu = b.Text[0]; //Lấy kí hiệu toán tử
    if (chuoitam.Length == 0)
        toanhang[buoc - 1] = ketqua;
    else
        toanhang[buoc - 1] = Double.Parse(chuoitam);
    if (buoc == 2)
    {
        btnBang_Click(null, null);
        toantu = b.Text[0];
        toanhang[0] = ketqua;
        buoc = 2;
    }
    else
    {
        txtManHinh.Text = toanhang[0].ToString();
        buoc++;
    }
    chuoitam = "";
}
```


BÀI TẬP THỰC HÀNH 3: LẬP TRÌNH C# WIN FORM (tiếp)

Mục tiêu kiến thức:

+ Sử dụng các thuộc tính, phương thức và sự kiện của các control: TrackBar, NumericUpDown, MaskedTextBox, DateTimePicker, MonthCalendar, Timer, ProgressBar, ToolTip, MenuStrip.

+ Viết ứng dụng dạng MDI

+ ListView, ImageList, TreeView

Mục tiêu kỹ năng:

- Có kỹ năng tốt trong lập trình Winform với các điều khiển nâng cao.

Bài 1. Thiết kế giao diện như sau:

The screenshot shows a WinForm titled "Employee Details" with the following fields and controls:

- Employee Name:
- Date of Birth: (mm/dd/yyyy)
- Address:
- City: (Ho Chi Minh, Nha Trang, Ha Noi)
- Country: (Select)
- Qualification: (University, Master, Ph D)
- Phone: (-)
- E-Mail:
- Date of Joining: 11/ 6/2009 (mm/dd/yyyy)

Annotations on the right side of the form:

- A blue box labeled "TextBox" has an arrow pointing to the "Employee Name" field.
- A blue box labeled "MaskedTextBox" has an arrow pointing to the "Phone" field.
- A blue box labeled "MaskedTextBox" has an arrow pointing to the "E-Mail" field.

Yêu cầu:

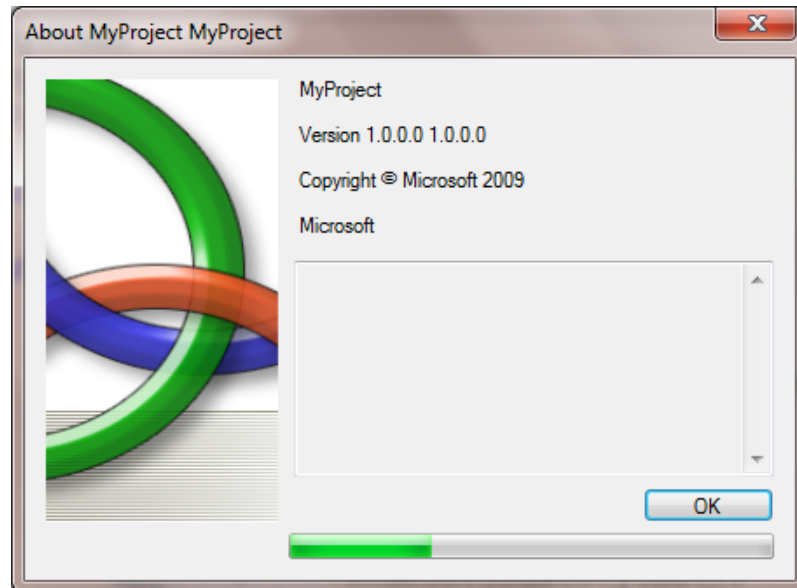
Quy định của Masktextbox Phone là 000-0000000.

ComboBox Country chỉ chứa 2 nước VietNam và Thailan (chứa 3 thành phố Pattaya, ChiangMai và Bangkok).

Khi đang nhập 1 ô mà bỏ trống và focus đến ô khác thì sẽ có thông báo lỗi và cho focus về ô cần nhập.

Khi nhấn Submit sẽ có một MessageBox hiển thị đầy đủ thông tin vừa nhập.

Bài 2. Tạo giao diện Form Splash như hình: chứa Progressbar và một nút OK.



Khi khởi động chương trình thì Form Splash xuất hiện. Form này dừng trong thời gian là 15s.

Thanh Progressbar sẽ thể hiện tiến trình load form này, sau 15s Form Splash sẽ tự động tắt và khởi động Form trong bài 1.

Trong khi Form Splash đang hoạt động, nếu người dùng nhấn vào nút OK thì Form này ngưng hoạt động, đồng thời Form bài 1 được gọi hoạt động.

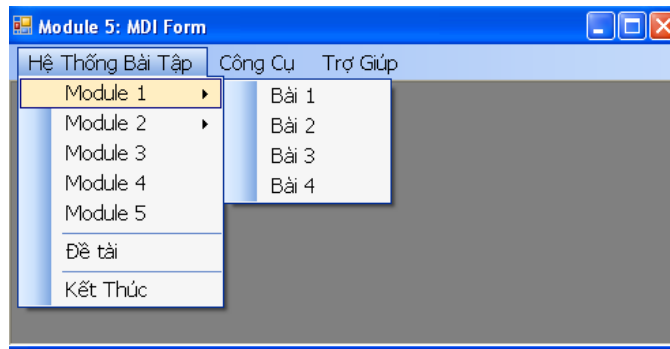
Hướng dẫn: kéo thả thanh Progressbar vào Form. Bắt sự kiện cho nút OK

Bài 3.

- + Tạo một ứng dụng liên kết các ứng dụng đã tạo ra từ các project trước.
- + Tạo form Splash
- + Tạo form About

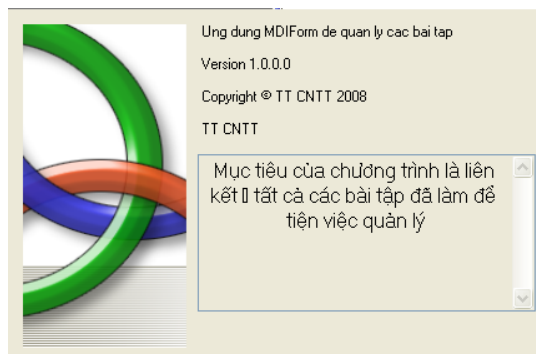
Yêu cầu:

Tổ chức Form chính như mẫu sau:



Yêu cầu liên kết các project đã có sẵn vào menu.

Thiết kế Form Splash cho chương trình (tùy ý):



Thiết kế Form About cho chương trình, tùy ý nhưng phải mang thông tin về chương trình như: tên chương trình, phiên bản, tác giả,...

Hướng dẫn:

Thiết lập một số thuộc tính của Form About:

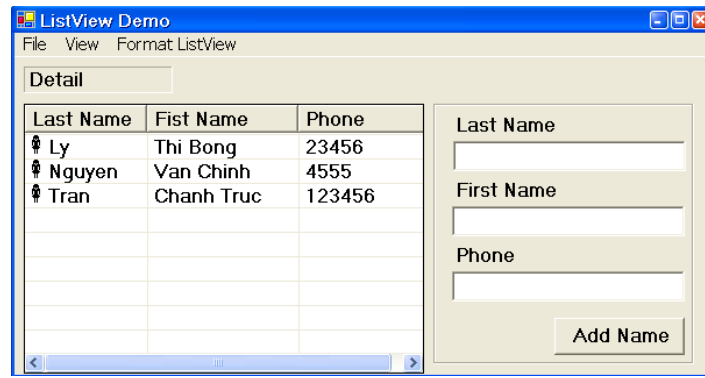
ControlBox → False

FormBorderStyle → FixedDialog

ShowInTaskbar → False

Bài 5. Thêm vào bài 1 chức năng cho phép người dùng mở dialog chọn màu để chọn màu cho Panel.

Viết chương trình nhập dữ liệu vào Listview như hình:



Yêu cầu:

Người sử dụng nhập thông tin Last name, First name, Phone và sử dụng nút Add Name để nhập vào Listview.

Các dòng trong Listview có biểu tượng (icon) hiển thị như hình.

Người sử dụng có thể thay đổi chế độ view của Listview bằng menu View.

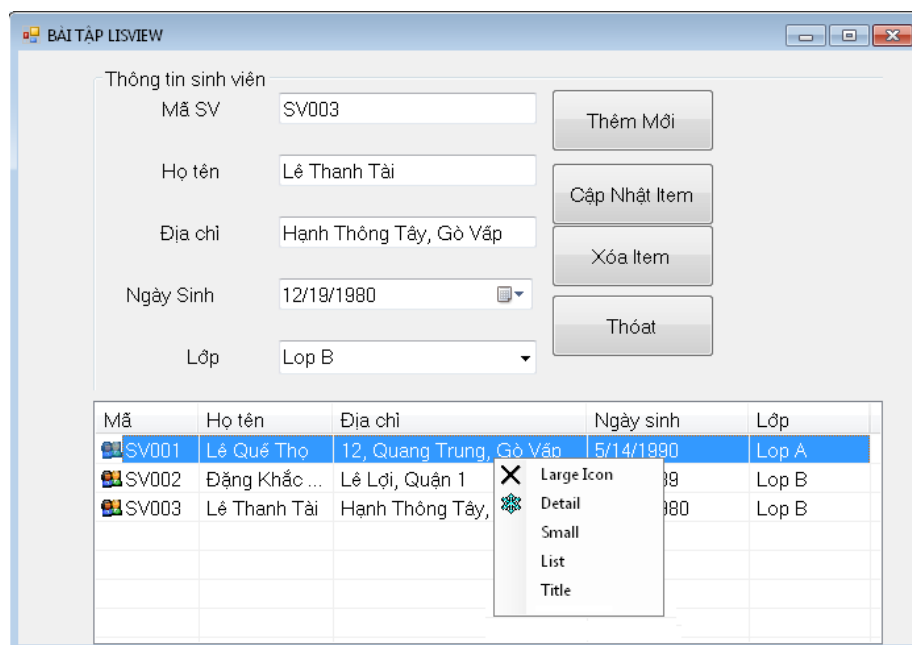
Menu FormatListView hiển thị hộp thoại chọn màu dùng để thay đổi dạng grid của Listview.

Gợi ý:

Sử dụng ListViewItem để thêm một dòng mới cho Listview.

Sử dụng ImageList để chứa thư viện icon cho Listview. Kết nối Listview với ImageList.

Bài 6. Thiết kế chương trình quản lý SV, cho phép nhập thông tin SV vào các Textbox như hình:



Yêu cầu:

Thêm vào Form hai Imagelist là *ilsNho* có kích thước mặc định 16 x 16, *ilsLon* có kích thước mặc 48 x 48 phục vụ cho ListView.

Nhấn nút Cập Nhật Item thì đưa thông tin sinh viên vào Listview theo các cột như hình.

Nhấn nút Xóa Item là xóa item đang chọn trên Listview (có thể chọn nhiều). Trước khi xóa cần xác nhận đã chọn Item nào chưa, xác nhận có chắc xóa không.

Nhấn nút Thêm Mới thì xóa thông tin sinh viên đang nhập và cho phép nhập thông tin sinh viên mới.

Nếu chọn một sinh viên nào trong Listview thì hiện lại thông tin Sinh Viên đó lên các Textbox tương ứng.

Click phải vào Listview cho phép hiện menu ngữ cảnh để chọn chức năng view.

Bài 7. Thiết kế giao diện như sau:

The screenshot shows a Windows application window titled "Tính tiền điện". Inside the window, there is a form titled "BÁO CÁO TIÊU THU ĐIỆN" (Electricity Billing Report). The form has several input fields: "HỌ TÊN KH:" (Customer Name), "KHU VỰC:" (Area) with a dropdown arrow, "ĐỊNH MỨC:" (Rate), "SỐ CŨ:" (Old Meter Reading), "SỐ MỚI:" (New Meter Reading), "TIÊU THU:" (Usage), and "THÀNH TIỀN:" (Amount). Below these fields are three buttons: "TÍNH TIỀN" (Calculate), "NHẬP MỚI" (New Entry), and "THOÁT" (Exit). To the right of the form is a list view with five columns: "Họ tên", "Khu vực", "Định mức", "Tiêu thụ", and "Thành tiền". At the bottom of the window, there is a "TỔNG TIỀN:" (Total Amount) label followed by a text box showing the value "0" and a "XÓA" (Delete) button.

Thực hiện các yêu cầu sau:

Thiết lập thuộc tính cho phép chọn nhiều dòng trên Listview.

Combobox có 3 khu vực: Khu vực 1 (định mức là 50), khu vực 2 (định mức là 100), khu vực 3 (định mức là 150). Khi chọn khu vực nào thì hiện định mức tương ứng.

Nút TÍNH TIỀN (hoặc Enter trên các textbox): kiểm tra dữ liệu nhập, nếu hợp lệ thì tính và xuất kết quả ra ô *Tiêu thụ* và *Thành tiền*, đồng thời thêm một dòng tương ứng vào Listview và cập nhật ô tổng tiền.

Đơn giá điện: trong định mức là 500, ngoài định mức là 1000.

Nút NHẬP MỚI: Xóa nội dung các textbox và label, đồng thời đặt con trỏ vào textbox đầu tiên

Nút XÓA: cho phép xóa 1 dòng đang chọn trong Listview, phải xác nhận lại trước khi xóa và cập nhật lại ô tổng tiền.

Nút THOÁT (hoặc nhấn Esc): thoát chương trình

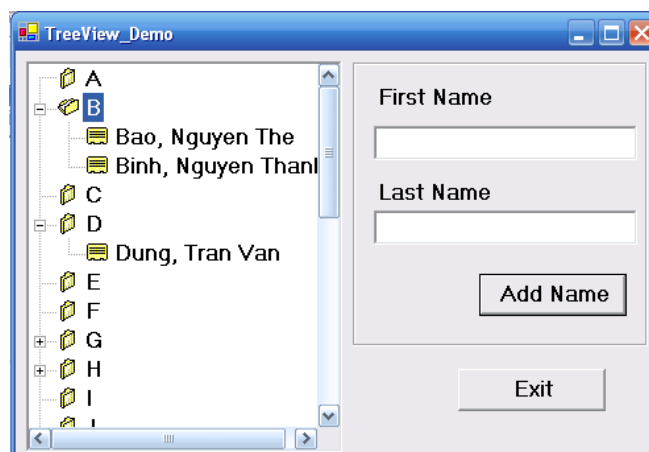
Quy định Form hiển thị giữa màn hình.

Quy định việc di chuyển tab hợp lý.

Thiết lập thuộc tính Anchor hợp lý cho các control.

Thiết lập MinimumSize cho form.

Bài 8. Viết chương trình nhập danh bạ với yêu cầu giao diện như hình dưới.



Yêu cầu:

Khi chương trình vừa hiển thị, Treeview chứa tất cả các chữ cái từ A->Z.

Nhằm mục đích tiện lợi cho người sử dụng khi tìm tên, khi người sử dụng nhập tên của một người nàp đó, chương trình sẽ đưa tên người này vào Treeview ở vị trí node có tương ứng với chữ các đầu của tên (xem hình).

Bài 9. Viết chương trình xem danh sách SV của Khoa Tin học như hình:

[illegible]

Yêu cầu:

Khi Form hiện lên, Treeview hiển thị danh sách các lớp – sinh viên như hình, chưa có nút nào được chọn. Con trỏ đặt tại ô Nhập tên.

Khi người dùng chọn nút cấp Khoa, chương trình hiện toàn bộ danh sách SV thuộc Khoa vào Listview.

Khi người dùng chọn một lớp bất kỳ thì chương trình hiện toàn bộ danh sách SV thuộc lớp đang chọn vào Listview.

Khi chọn 1 SV bất kỳ thì chỉ hiện thị SV đó vào Listview.

Nút Tìm: cho phép tìm SV (trong cấp đang chọn trên Treeview) có họ tên chứa chuỗi nhập trong Textbox. Hiện kết quả ra Listview.

Bài 10. Thiết kế giao diện như sau:

The screenshot shows a Windows application window titled "Quản lý sinh viên". The interface is split into two main panes. The left pane, titled "Danh sách lớp" (Class List), contains a tree view with a root node "NCTH3KA" which is currently selected. Below it are two sub-nodes: "9912578-Nguyễn Văn A" and "Đồng Nai". The right pane, titled "Thông tin sinh viên" (Student Information), displays the details for the selected student. It contains three text input fields: "Mã sv" (Student ID) with the value "9912578", "Họ Tên" (Full Name) with the value "Nguyễn Văn A", and "Địa chỉ" (Address) with the value "Đồng Nai". At the bottom of the right pane are two buttons: "Cập Nhật" (Update) and "Xóa" (Delete).

Thực hiện các yêu cầu sau:

Thiết lập HideSelection = False.

Khi Form hiện lên, đã có sẵn 1 số lớp trong danh sách lớp ở Treeview.

Nút Cập Nhật: Thêm 1 SV vào lớp đang chọn trên Treeview với nội dung các nút như hình. Trước khi thêm phải kiểm tra thông tin nhập gồm: các ô nhập không được để trống, không được trùng mã SV. Ngoài ra còn phải kiểm tra nút chọn trên Treeview có phải là nút lớp không (chỉ được thêm vào nút lớp).

Nút Xóa: cho phép xóa nút đang chọn trong Treeview, phải xác nhận lại trước khi xóa và chỉ được xóa khi chọn nút chứa mã SV.

Khi click chọn nút mã SV hoặc địa chỉ thì hiện thông tin sv đó qua các Textbox.

Quy định Form hiển thị giữa màn hình.

Quy định việc di chuyển tab hợp lý.

Thiết lập thuộc tính Dock hợp lý cho Treeview.

Thiết lập MinimumSize cho form.

BÀI TẬP THỰC HÀNH 5: TRUY CẬP DỮ LIỆU VỚI ADO.NET

A. Đối tượng Connection, Command, DataReader

1. Mục tiêu

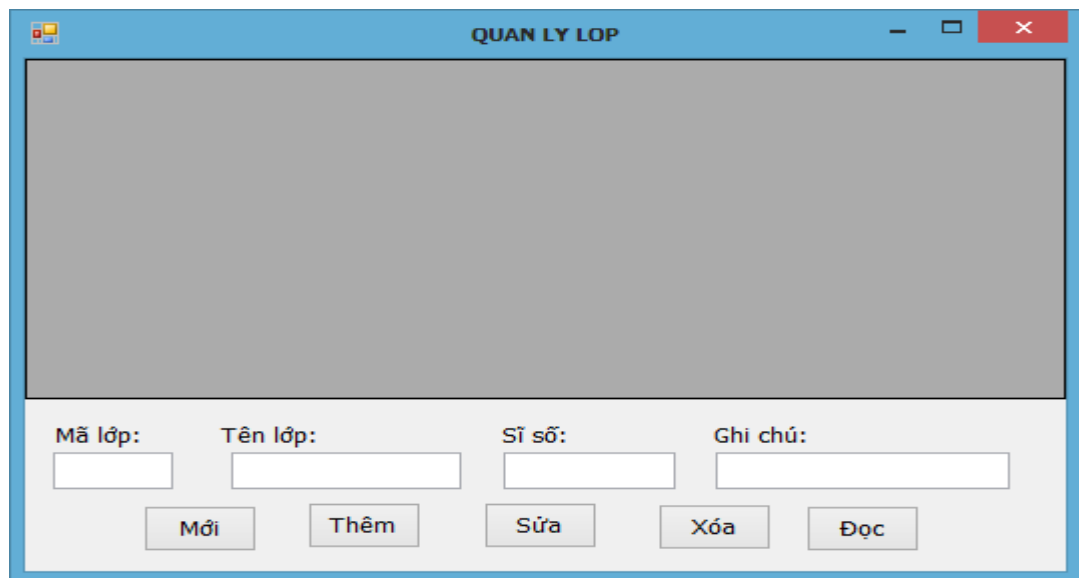
- Khai báo và sử dụng được đối tượng Connection.
- Sử dụng được đối tượng Connection để kết nối tới csdl SQL Server và viết được ứng dụng cụ thể kết nối tới SQL Server.
- Nắm vững cách khai báo và sử dụng đối tượng Command và DataReader.
- Sử dụng được hai đối tượng này để viết được ứng dụng quản lý.

2. Nội dung

- Tạo Cơ sở dữ liệu QLSV trong SQL Server → tạo bảng tbLop:

	Column Name	Data Type	Allow Nulls
?	MaLop	nchar(10)	<input type="checkbox"/>
	TenLop	nchar(10)	<input type="checkbox"/>
	SiSo	int	<input type="checkbox"/>
	GhiChu	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- Tạo Form có giao diện như sau (DataGridView và các Button có tên lần lượt là: dgLop, btMoi, btThem):



- Viết code thực hiện các chức năng:
 - o Khi form được Load thì Load dữ liệu của bảng tbLop lên dgLop.
 - o Viết chức năng tương ứng cho các nút Mới, Thêm, Sửa, Xóa, Đọc (Đọc toàn bộ dữ liệu bảng tbLop).

Hướng dẫn thực hiện:

- Cách 1: Viết chung các phương thức vào trong Form.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Database
{
    public partial class frmQLLop : Form
    {
        public frmQLLop()
        {
            InitializeComponent();

            //Khai báo các đối tượng
            string strKetNoi = @"Data Source=VIETHQTK1\VIETHQTK1;Initial
Catalog=QLSV;Integrated Security=True";

            SqlConnection sqlCon;
            SqlCommand sqlCom;
            SqlDataReader sqlRe;
            SqlDataAdapter sqlAdap;
            DataTable dt;

            //Các phương thức
            //Phương thức 1: Mở kết nối đến CSDL
            void KetNoi()
            {
                sqlCon = new SqlConnection(strKetNoi);
                if (sqlCon.State == ConnectionState.Closed)
                {
```

```
        sqlCon.Open();
    }
}
```

//Phương thức 2: Phương thức đóng kết nối

```
void NgatKetNoi()
{
    sqlCon.Close();
}
```

//Phương thức 3: kiểm tra mã trùng dạng tổng quát

```
int KiemTraMaTrung(string strSelect)
{
    int sbg = 0;
    KetNoi();
    sqlCom = new SqlCommand(strSelect, sqlCon);
    sbg = (int)sqlCom.ExecuteScalar();
    NgatKetNoi();
    //return
    return sbg;
}

//Phương thức 4: thực thi câu lệnh SQL (thêm, sửa, xóa)
void ThucThi(string strSQL)
{
    //B1:
    KetNoi();
    //B2:
    sqlCom = new SqlCommand(strSQL, sqlCon);
    sqlCom.ExecuteNonQuery();
    //B3:
    NgatKetNoi();
}
```

//Phương thức 5: Load dữ liệu vào DataGridView

```
void LoadDuLieuDataGridView(string strSelect)
{
    dt = new DataTable();
    sqlAdap = new SqlDataAdapter(strSelect, strKetNoi);
    sqlAdap.Fill(dt);
    dgLop.DataSource = dt;
}
```

//Form_Load

```
private void frmQLLop_Load(object sender, EventArgs e)
{
    LoadDuLieuDataGridView("select * from tbLop");
}
```

//Sự kiện Cell Click: chọn bản ghi cần sửa trên DataGridView

//Hiển thị hàng ghi được chọn xuống các Textbox

```
private void dgLop_CellClick(object sender, DataGridViewCellEventArgs e)
{
    int HangChon = e.RowIndex;
    txtMa.Text = dgSinhVien[0, HangChon].Value.ToString();
    txtTenLop.Text = dgSinhVien[1, HangChon].Value.ToString();
    txtSiSo.Text = dgSinhVien[2, HangChon].Value.ToString();
    txtGhiChu.Text = dgSinhVien[3, HangChon].Value.ToString();
}
```

//Mới

```
private void btMoi_Click(object sender, EventArgs e)
{
    txtMa.Text = "";
    txtTenLop.Text = "";
    txtSiSo.Text = "";
    txtGhiChu.Text = "";
}
```

```

//Thêm
private void btThm_Click(object sender, EventArgs e)
{
    //ThucThi("Insert INTO TBLOP values('201222','TK11LC2','8','ít') ");//Dữ liệu cố định
    //Dữ liệu từ các Textbox
    //Kiểm tra mã có trùng hay không?
    if (KiemTraMaTrung("select count(MaLop) from tblop where MaLop='" + txtMa.Text +
        """) == 1)
        MessageBox.Show("Mã: " + txtMa.Text + " đã tồn tại. Vui lòng chọn mã khác!");
    else if (KiemTraMaTrung("select count(MaLop) from tblop where MaLop='" +
        txtMa.Text + """) == 0)
    {
        ThucThi("Insert INTO TBLOP values('" + txtMa.Text + "'," + txtTenLop.Text + "'," +
        txtSiSo.Text + "',N'" + txtGhiChu.Text + "')");
        LoadDuLieuDataGridView("select * from tbLop");
    }
}

//Sửa
private void btSua_Click(object sender, EventArgs e)
{
    //Update
    string strUpdate = "update tbLop set TenLop='" + txtTenLop.Text + "',
        SiSo='" + txtSiSo.Text + "',GhiChu='" + txtGhiChu.Text + "' where MaLop='" + txtMa.Text + "' ";
    ThucThi(strUpdate);
    //Load lại dữ liệu vào DataGridView
    LoadDuLieuDataGridView("select * from tbLop");
}

//Xóa
private void btXoa_Click(object sender, EventArgs e)
{
    //Delete
    string strDelete = "Delete from tbLop where Malop='" + txtMa.Text + "' ";

```

```

        ThucThi(strDelete);
        //Load lại dữ liệu vào DataGridView
        LoadDuLieuDataGridView("select * from tbLop");
    }

    //Đọc dữ liệu
    private void btDoc_Click(object sender, EventArgs e)
    {
        //B1:
        KetNoi();
        //B2:
        sqlCom = new SqlCommand("select * from tbLop", sqlCon);
        sqlRe = sqlCom.ExecuteReader();
        while (sqlRe.Read())
        {
            MessageBox.Show(sqlRe[1].ToString());
        }
        //B3:
        NgatKetNoi();
    }
}
}

```

- Cách 2: Viết lớp clsDatabase chuyên thao tác với CSDL → sinh viên tự làm.

B. Đối tượng DataAdapter, DataSet

1. Mục tiêu:

- Nắm vững cách khai báo và sử dụng đối tượng DataAdapter và DataSet.
- Nắm vững mối liên hệ giữa hai đối tượng.
- Sử dụng được hai đối tượng này để hoàn thiện ứng dụng csdl

2. Nội dung

- Tạo Cơ sở dữ liệu QLSV trong SQL Server → tạo bảng tbLop:

	Column Name	Data Type	Allow Nulls
▶	MaLop	nchar(10)	<input type="checkbox"/>
	TenLop	nchar(10)	<input type="checkbox"/>
	SiSo	int	<input type="checkbox"/>
	GhiChu	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

- Tạo Form có giao diện như sau (DataGridView và các Button có tên lần lượt là: dgLop, btMoi, bt Them):

- Viết code thực hiện các chức năng:
 - o Khi form được Load thì Load dữ liệu của bảng tbLop lên dgLop.
 - o Viết chức năng tương ứng cho các nút Mới, Thêm, Sửa, Xóa, Đọc (Đọc toàn bộ dữ liệu bảng tbLop).

Hướng dẫn thực hiện:

- Cách 1: Viết chung các phương thức thao tác với CSDL vào trong Form.

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
```

```
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Database
{
    public partial class frmQLLop2 : Form
    {
        public frmQLLop2()
        {
            InitializeComponent();

            //Khai báo
            string strKetNoi = @"Data Source=VIETHQTK1\VIETHQTK1;Initial
Catalog=QLSVCNTT;Integrated Security=True";
            SqlDataAdapter sqlAdap;
            DataTable dt;

            //Các phương thức
            //Phương thức 1: thực thi câu lệnh SQL (insert, update, delete)
            void ThucThi(string strSQL)
            {
                DataTable dtSQL=new DataTable();
                sqlAdap = new SqlDataAdapter(strSQL, strKetNoi);
                sqlAdap.Fill(dtSQL);
            }
            //Phương thức 2: Kiểm tra mã trùng
            int KiemTraMaTrung(string strSelect)
            {
                int sbg = 0;
                DataTable dtSelect = new DataTable();
                sqlAdap = new SqlDataAdapter(strSelect, strKetNoi);
                sqlAdap.Fill(dtSelect);
                //
            }
        }
    }
}
```

```

        sbg = dtSelect.Rows.Count;
        return sbg;
    }

    //Phương thức 3: Load dữ liệu vào DataGridView
    void LoadDuLieuDataGridView(string strSelect)
    {
        dt = new DataTable();
        sqlAdap = new SqlDataAdapter(strSelect, strKetNoi);
        sqlAdap.Fill(dt);
        dgSinhVien.DataSource = dt;
    }

    //Load
    private void frmQLLop2_Load(object sender, EventArgs e)
    {
        LoadDuLieuDataGridView("select * from tbLop");
    }

    //Sự kiện Cell Click: chọn bản ghi cần sửa trên DataGridView
    //Hiển thị hàng ghi được chọn xuống các Textbox
    private void dgSinhVien_CellClick(object sender, DataGridViewCellEventArgs e)
    {
        int HangChon = e.RowIndex;
        txtMa.Text = dgSinhVien[0, HangChon].Value.ToString();
        txtTenLop.Text = dgSinhVien[1, HangChon].Value.ToString();
        txtSiSo.Text = dgSinhVien[2, HangChon].Value.ToString();
        txtGhiChu.Text = dgSinhVien[3, HangChon].Value.ToString();
    }

    //Mới
    private void btMoi_Click(object sender, EventArgs e)
    {
        txtMa.Text = "";
    }

```

```

        txtTenLop.Text = "";
        txtSiSo.Text = "";
        txtGhiChu.Text = "";
    }

    //Thêm
    private void btThm_Click(object sender, EventArgs e)
    {
        //ThucThi("Insert INTO TBLOP values('201222','TK11LC2','8','ít') "); //Dữ liệu cố định
        //Dữ liệu từ các Textbox
        //Kiểm tra mã có trùng hay không?
        if (KiemTraMaTrung("select * from tblop where MaLop='" + txtMa.Text + "'" ) == 1)
            MessageBox.Show("Mã: " + txtMa.Text + " đã tồn tại. Vui lòng chọn mã khác đi!");
        else if (KiemTraMaTrung("select * from tblop where MaLop='" + txtMa.Text + "'" ) ==
0)
        {
            ThucThi("Insert INTO TBLOP values('" + txtMa.Text + "', '" + txtTenLop.Text + "', '" +
txtSiSo.Text + "', 'N' + txtGhiChu.Text + "') ");
            LoadDuLieuDataGridView("select * from tbLop");
        }
    }

    //Sửa
    private void btSua_Click(object sender, EventArgs e)
    {
        //Update
        string strUpdate = "update tbLop set TenLop='" + txtTenLop.Text + "', SiSo='" +
txtSiSo.Text + "', GhiChu='" + txtGhiChu.Text + "' where MaLop='" + txtMa.Text + "' ";
        ThucThi(strUpdate);
        //Load lại dữ liệu vào DataGridView
        LoadDuLieuDataGridView("select * from tbLop");
    }

    //Xóa
    private void btXoa_Click(object sender, EventArgs e)

```

```

{
    //Delete
    string strDelete = "Delete from tbLop where Malop='" + txtMa.Text + "' ";
    ThucThi(strDelete);
    //Load lại dữ liệu vào DataGridView
    LoadDuLieuDataGridView("select * from tbLop");
}
}
}

```

- Cách 2: Viết lớp clsDatabase chuyên thao tác với CSDL.

Lớp clsThaoTac:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace Database
{
    class clsThaoTac
    {
        //LỚP NÀY CÓ NHIỆM VỤ CHUYÊN THAO TÁC CSDL: LOAD DỮ LIỆU LÊN DATAGRIDVIEW,
        THÊM ,SỬA, XÓA....

        //SỬ DỤNG 2 CÁCH: (CON, COM, READER) HOẶC (ADAPTER, DATATABLE)

        //Khai báo dữ liệu

        string strKetNoi = @"Data Source=VIETHQTK1\VIETHQTK1;Initial
        Catalog=QLSV;Integrated Security=True";

        SqlDataAdapter sqlAdap;

        /// <summary>

        /// Phương thức 1: Load dữ liệu từ CSDL lên DataGridView
    }
}

```

```

/// </summary>
/// <param name="dg">Tên điều khiển DataGridView</param>
/// <param name="strSelect">Chuỗi string mô tả câu lệnh Select</param>
public void Data2DataGridView(DataGridView dg, string strSelect)
{
    DataTable dtLoad = new DataTable();
    sqlAdap = new SqlDataAdapter(strSelect, strKetNoi);
    sqlAdap.Fill(dtLoad);
    //
    dg.DataSource = dtLoad;
}

/// <summary>
/// Phương thức 2: Kiểm tra mã trùng
/// </summary>
/// <param name="strSelect">Câu lệnh Select</param>
/// <returns></returns>
public int TongBanGhi(string strSelect)
{
    DataTable dtTong = new DataTable();
    sqlAdap = new SqlDataAdapter(strSelect, strKetNoi);
    sqlAdap.Fill(dtTong);
    //
    int sbg = dtTong.Rows.Count;
    return sbg;
}

/// <summary>
/// Phương thức 3: Thực thi câu lệnh Insert, Update, Delete...
/// </summary>
/// <param name="strSQL">Câu lệnh SQL: Insert, update, Delete</param>
public void ThucThi(string strSQL)
{

```

```

        DataTable dtSQL = new DataTable();
        sqlAdap = new SqlDataAdapter(strSQL, strKetNoi);
        sqlAdap.Fill(dtSQL);
    }
    //End
}

```

Form:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Database
{
    public partial class frmQLLop3 : Form
    {
        public frmQLLop3()
        {
            InitializeComponent();
        }
        //Tạo đối tượng lớp
        clsThaoTac c = new clsThaoTac();

        //Load Form
        private void frmQLLop3_Load(object sender, EventArgs e)
        {
            c.Data2DataGridView(dgLop, "select * from tbLop");
        }
    }
}

```

//Sự kiện Cell Click

```
private void dgLop_CellClick(object sender, DataGridViewCellEventArgs e)
```

```
{  
    int hang = e.RowIndex;  
    txtMaLop.Text = dgLop[0, hang].Value.ToString();  
    txtTenLop.Text = dgLop[1, hang].Value.ToString();  
    txtSS.Text = dgLop[2, hang].Value.ToString();  
    txtGhiChu.Text = dgLop[3, hang].Value.ToString();  
}
```

//Mới

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

```
{  
    txtMaLop.Text = "";  
    txtTenLop.Text = "";  
    txtSS.Text = "";  
    txtGhiChu.Text = "";  
}
```

//Thêm

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

```
{  
  
    if (c.TongBanGhi("select * from tbLop where MaLop='" + txtMaLop.Text + "'" ) == 0)  
    {  
        string strThem = "insert into tbLop values('" + txtMaLop.Text + "','" +  
txtTenLop.Text + "','" + txtSS.Text + "','" + txtGhiChu.Text + "')";  
        c.ThucThi(strThem);  
        c.Data2DataGridView(dgLop, "select * from tbLop");  
        btMoi_Click(sender, e);  
    }  
  
    else MessageBox.Show("Mã: " + txtMaLop.Text + " đã tồn tại. Hãy chọn mã khác",  
"Message", MessageBoxButtons.OK, MessageBoxIcon.Information);  
}
```



```

//Sửa
private void btSua_Click(object sender, EventArgs e)
{
    c.ThucThi("update tbLOp set TenLOp=" + txtTenLop.Text + ",SS=" + txtSS.Text + ",
GhiChu=" + txtGhiChu.Text + " where MaLop=" + txtMaLop.Text + " ");//Đúng
    c.Data2DataGridView(dgLop, "select * from tbLop");
}

//Xóa
private void btXoa_Click(object sender, EventArgs e)
{
    //Xóa
    if (MessageBox.Show("Bạn có thực sự muốn xóa hay không?", "Delete",
MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
    {
        c.ThucThi("delete from tbLOp where MaLop=" + txtMaLop.Text + "");
        c.Data2DataGridView(dgLop, "select * from tbLop");
    }
}

}
}

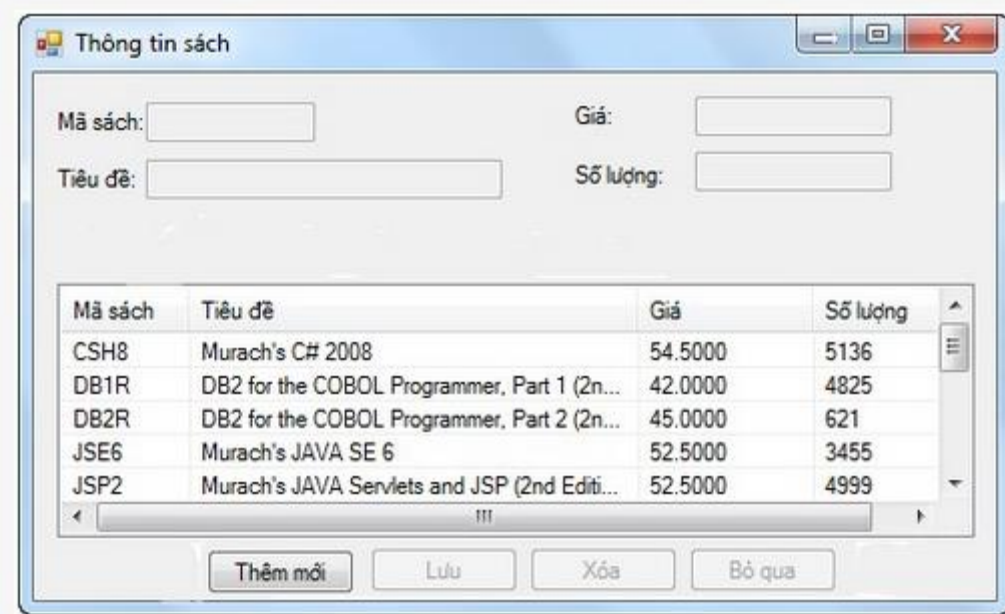
```

BÀI TẬP THỰC HÀNH 6: TRUY CẬP DỮ LIỆU VỚI ADO.NET(tiếp)

Nội dung kiến thức thực hành:

- Tìm hiểu một số khái niệm liên quan đến ADO.NET: Data Provider, Kiến trúc của ADO.NET.
- Sử dụng các đối tượng ADO.NET: SqlConnection, SqlCommand, SqlDataReader, SqlDataAdapter, DataSet, DataTable

Bài 1: Lập trình phần mềm Quản lý thông tin sách có thêm các chức năng thêm mới, xóa thông tin sách từ CSDL



Mã sách	Tiêu đề	Giá	Số lượng
CSH8	Murach's C# 2008	54.5000	5136
DB1R	DB2 for the COBOL Programmer, Part 1 (2n...	42.0000	4825
DB2R	DB2 for the COBOL Programmer, Part 2 (2n...	45.0000	621
JSE6	Murach's JAVA SE 6	52.5000	3455
JSP2	Murach's JAVA Servlets and JSP (2nd Editi...	52.5000	4999

Chức năng của ứng dụng được mô tả như sau:

- Khi load ứng dụng
 - Hiển thị thông tin về tất cả các loại sách trong ứng dụng trên ListView
 - Các TextBox đều bị vô hiệu hóa
 - Button Thêm mới được kích hoạt cho phép thêm thông tin sách mới
 - Button Lưu, Xóa, Bỏ qua bị vô hiệu hóa
 - Khi một cuốn sách trên ListView được chọn
 - Thông tin về cuốn sách được hiển thị trên các TextBox vẫn đang bị vô hiệu hóa
 - Button Thêm mới, Xóa được kích hoạt
 - Button Lưu, Bỏ qua bị vô hiệu hóa

The screenshot shows a Windows application window titled "Thông tin sách". It has four input fields at the top: "Mã sách:" with value "DB2R", "Giá:" with value "45.0000", "Tiêu đề:" with value "DB2 for the COBOL Programmer, Par", and "Số lượng:" with value "621". Below these is a table with four columns: "Mã sách", "Tiêu đề", "Giá", and "Số lượng". The table contains five rows of book data. The row with "Mã sách" DB2R is highlighted in blue. At the bottom of the window are four buttons: "Thêm mới", "Lưu", "Xóa", and "Bỏ qua".

Mã sách	Tiêu đề	Giá	Số lượng
CSH8	Murach's C# 2008	54.5000	5136
DB1R	DB2 for the COBOL Programmer, Part 1 (2n...	42.0000	4825
DB2R	DB2 for the COBOL Programmer, Part 2 (2n...	45.0000	621
JSE6	Murach's JAVA SE 6	52.5000	3455
JSP2	Murach's JAVA Servlets and JSP (2nd Editi...	52.5000	4999

Hình 2. Form khi một item trên ListView được chọn .

- Khi button Thêm mới được nhấn
- Các TextBox được kích hoạt cho phép nhập thông tin
- Button Lưu được kích hoạt cho phép lưu thông tin
- Button Bỏ qua được kích hoạt cho phép bỏ qua thao tác Thêm mới
- Button Thêm mới, Xóa, bị vô hiệu hóa
- Khi button Xóa được nhấn
- Hiện thị thông báo hỏi người dùng có chắc chắn xóa không
- Xóa thông tin

Yêu cầu: Thực hiện thao tác thêm hàng, xóa hàng trên DataSet, sau đó cập nhật lại CSDL

1. DATABASE

```
use master
go
drop database sachdb
go

CREATE DATABASE SachDB
go
use SachDB
go
create table categories
(
categoryid int identity(1,1) primary key,
categoryname varchar(50)
)
go

insert into categories(categoryname) values ('Sách .net')
```

```

go
insert into categories(categoryname) values ('Sách java')
go
CREATE TABLE [dbo].[Products](

[ProductCode] [char](10) NOT NULL PRIMARY KEY,

[Description] [varchar](50) NOT NULL,

[UnitPrice] [money] NOT NULL,

[OnHandQuantity] [int] NOT NULL,

categoryid int

constraint fk_c_p foreign key(categoryid) references
categories(categoryid)

)

GO

INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity],categoryid) VALUES (N'A3CS
', N'Murach"s ASP.NET 3.5 Web Programming with C# 2008',
54.5000, 4637,1)

INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity],categoryid) VALUES (N'A3VB
', N'Murach"s ASP.NET 3.5 Web Programming with VB 2008',
54.5000, 3974,1)

INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity],categoryid) VALUES (N'ADC3
', N'Murach"s ADO.NET 3.5, LINQ, and EF with C# 2008', 54.5000,
5244,2)

INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity],categoryid) VALUES (N'ADV3
', N'Murach"s ADO.NET 3.5, LINQ, and EF with VB 2008', 54.5000,
4538,2)

INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'CRFC
',
N'Murach"s CICS Desk Reference', 50.0000, 1865)

INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'CSH8
',
N'Murach"s C# 2008', 54.5000, 5136)

```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'DB1R      \, N'DB2 for
the COBOL Programmer, Part 1 (2nd Edition)', 42.0000, 4825)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'DB2R      \, N'DB2 for
the COBOL Programmer, Part 2 (2nd Edition)', 45.0000, 621)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'JSE6      \,
N'Murach"s JAVA SE 6', 52.5000, 3455)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'JSP2      \,
N'Murach"s JAVA Servlets and JSP (2nd Edition)', 52.5000, 4999)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'MCBL      \,
N'Murach"s Structured COBOL', 62.5000, 2386)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'MCCP      \,
N'Murach"s CICS for the COBOL Programmer', 54.0000, 2368)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'MDOM      \,
N'Murach"s JavaScript and DOM Scripting', 54.5000, 6937)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'SQL8      \,
N'Murach"s SQL Server 2008', 52.5000, 2465)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'VB08      \,
N'Murach"s Visual Basic 2008', 54.5000, 2193)
```

```
INSERT [dbo].[Products] ([ProductCode], [Description],
[UnitPrice], [OnHandQuantity]) VALUES (N'ZJLR      \,
N'Murach"s OS/390 and z/os JCL', 62.5000, 677)
```

```
go
select * from categories
go
select * from products
go
```

2. CODING

```
public partial class Form1 : Form
{
public Form1()
```

```

{
InitializeComponent();
}

private void load_data()
{
    SqlConnection con = new
    SqlConnection("server=.;database=sachdb;integrated
    security=true;");
    SqlDataAdapter da = new SqlDataAdapter("select * from
    products",con);
    DataTable tb = new DataTable();
    da.Fill(tb);
    dataGridView1.DataSource = tb;

    //databinding
    textBox1.DataBindings.Clear();
    textBox2.DataBindings.Clear();
    textBox3.DataBindings.Clear();
    textBox4.DataBindings.Clear();

    textBox1.DataBindings.Add("Text",
    dataGridView1.DataSource, "productcode");
    textBox2.DataBindings.Add("Text",
    dataGridView1.DataSource, "Description");
    textBox3.DataBindings.Add("Text",
    dataGridView1.DataSource, "UnitPrice");
    textBox4.DataBindings.Add("Text",
    dataGridView1.DataSource, "OnHandQuantity");

}

private void Form1_Load(object sender, EventArgs e)
{
    load_data();
    textBox1.Enabled = false;
    textBox2.Enabled = false;
    textBox3.Enabled = false;
    textBox4.Enabled = false;

    button2.Enabled = false;
    button3.Enabled = false;
    button4.Enabled = false;
}

private void button1_Click(object sender, EventArgs e)
{
    textBox1.Enabled = true;

    textBox2.Enabled = true;
    textBox3.Enabled = true;
    textBox4.Enabled = true;
}

```

```

        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
        textBox1.Focus();

        button1.Enabled = false;
        button3.Enabled = false;
        button2.Enabled = true;
        button4.Enabled = true;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        SqlConnection con = new
        SqlConnection("server=.;database=sachdb;integrated
        security=true;");
        SqlCommand cmd = new SqlCommand("insert into products
        values('" + textBox1.Text + "\",'" + textBox2.Text + "\",'"
        + textBox3.Text + "\",'" + textBox4.Text + "\"", con);
        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();
        load_data();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        DialogResult kq = MessageBox.Show("ban muon xoa khong
        ?", "tieu de", MessageBoxButtons.YesNo);

        if (kq == System.Windows.Forms.DialogResult.Yes)
        {
            SqlConnection con = new
            SqlConnection("server=.;database=sachdb;integrated
            security=true;");
            SqlCommand cmd = new SqlCommand("delete from products
            where productcode = '" + textBox1.Text + "'", con);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            load_data();
        }
    }

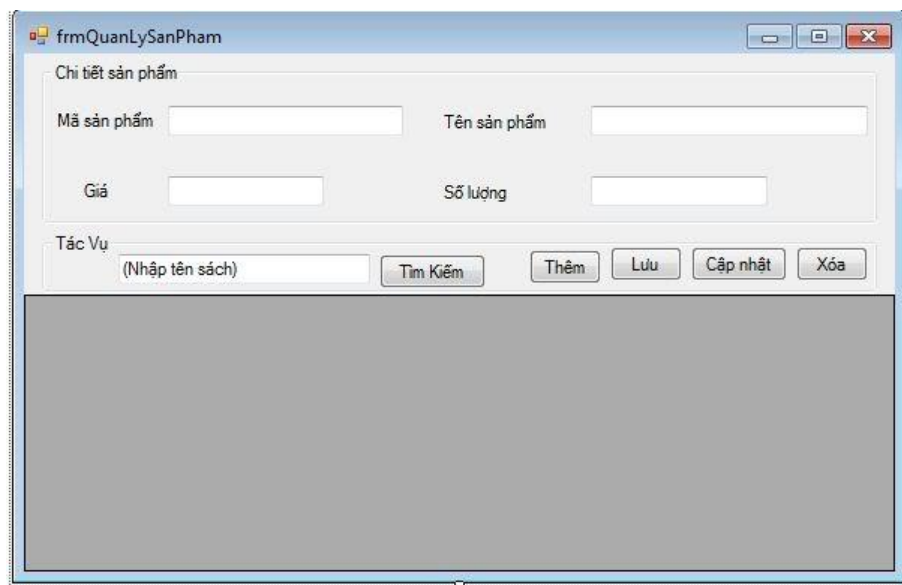
    private void dataGridView1_CellClick(object sender,
    DataGridViewCellEventArgs e)
    {
        button3.Enabled = true;
        button4.Enabled = true;
    }

```

```
private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    dataGridView1_CellClick(sender, e);
}
}
```

Bài 2. Thực hiện các thao tác với DataGridView: hiển thị dữ liệu từ CSDL (SQL), thêm dữ liệu, xóa dữ liệu, cập nhật dữ liệu, tìm kiếm dữ liệu.

Thực hiện việc bắt lỗi nếu số lượng không phải là số nguyên và >100. Demo chương trình bằng hình ảnh.



Bài 3: Viết chương trình quản lý Sinh Viên C# với databse SQL trong đó

- Load dữ liệu từ file SQL vào Visual Studio
- Click hiện thông tin từ bảng DataGridView lên các TextBox
- Sử dụng được các nút Thêm, Xóa, Sửa, Lưu, Hủy, Thoát
- Có sử dụng việc bắt lỗi

Form1

www.tamga.tk

QUẢN LÝ THÔNG TIN SINH VIÊN

Cách 2

Mã SV

T05

Họ SV

Tam

Tên SV

Ga

Mã Khoa

CN

Ngày sinh

09/23/1985

Giới tính

Nam

www.c10mt.tk

www.c10maytinh.tk

www.tamga.tk

tamgaalbum@yahoo.com

Phone: 01283.98.69.98

	Mã SV	Họ SV	Tên SV	Ngày Sinh	Giới tính	Mã Khoa
	C04	Nguyễn Hoàng ...	Hưng ...	03/19/1990	Nam	CN
	T00	Lê	Tuấn	02/15/1991	Nam	TO
	T01	Bùi Minh	Khánh ...	04/09/1990	Nam	TO
	T02	Trần Thị	Lan ...	03/04/1990	Nữ	TO
	T03	Lê	Thiện	05/18/1990	Nam	TO
	T04	Lê Thị	Thảo ...	03/27/1990	Nữ	TO
▶	T05	Tam	Ga	09/23/1985	Nam	CN

Thêm

Sửa

Xóa

Lưu

Hủy

Thoát

BÀI TẬP THỰC HÀNH 7: TRUY CẬP DỮ LIỆU VỚI ADO.NET (tiếp)

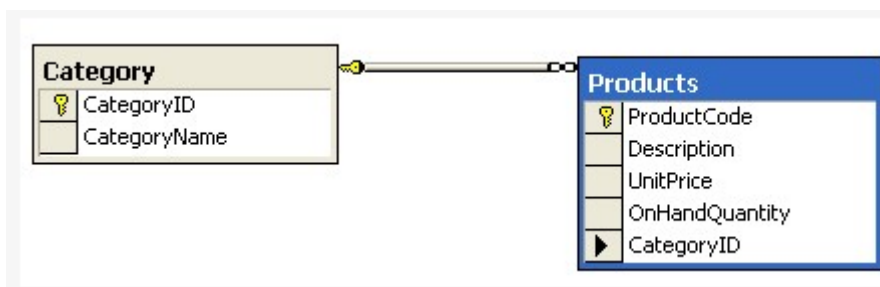
Mục tiêu kiến thức:

Sinh viên vẫn tiếp tục làm các bài thực hành liên quan tới cơ sở dữ liệu nhưng tập trung vào phần thống kê đưa ra danh sách dựa vào các điều kiện có trước. Ngoài ra, hướng dẫn một số cách làm khác để sinh viên học tập.

Mục tiêu kỹ năng:

- Có kỹ năng tốt trong thực hành kết nối cơ sở dữ liệu ADO.NET như hiển thị dữ liệu, thêm dữ liệu,...

Bài 1:



Thông tin sách

Loại sách:

Mã sách:

Tiêu đề:

Giá:

Số lượng:

Mã sách	Tiêu đề	Giá	Số lượng
A3CS	Murach's ASP.NET 3.5 Web Programming ...	54.5000	4637
A3VB	Murach's ASP.NET 3.5 Web Programming ...	54.5000	3974
ADC3	Murach's ADO.NET 3.5, LINQ, and EF with...	54.5000	5244
ADV3	Murach's ADO.NET 3.5, LINQ, and EF with...	54.5000	4538
CRFC	Murach's CICS Desk Reference	50.0000	1865
CSH8	Murach's C# 2008	54.5000	5136

Thêm mới Sửa Lưu Xóa Bỏ qua

Hãy lập trình để thực hiện sau

- Đưa dữ liệu từ bảng Category vào Combobox tên là loại sách

b. Thực hiện chức năng nhập mới sách theo Category.

c. Hiển thị dữ liệu của bảng Product ra DataGridView

Hướng dẫn:

```
public partial class Form2 : Form
{
    public Form2()
    {
        InitializeComponent();
    }

    private void load_data()
    {
        SqlConnection con = new
        SqlConnection("server=.;database=sachdb;integrated
        security=true;");
        SqlDataAdapter da = new SqlDataAdapter("select * from
        products", con);
        DataTable tb = new DataTable();
        da.Fill(tb);
        dataGridView1.DataSource = tb;
        //databinding
        textBox1.DataBindings.Clear();
        textBox2.DataBindings.Clear();
        textBox3.DataBindings.Clear();
        textBox4.DataBindings.Clear();
        textBox1.DataBindings.Add("Text",
        dataGridView1.DataSource, "productcode");
        textBox2.DataBindings.Add("Text",
        dataGridView1.DataSource, "Description");
        textBox3.DataBindings.Add("Text",
        dataGridView1.DataSource, "UnitPrice");
        textBox4.DataBindings.Add("Text",
        dataGridView1.DataSource, "OnHandQuantity");
    }
    private void Form2_Load(object sender, EventArgs e)
    {
        load_data();
        textBox1.Enabled = false;
        textBox2.Enabled = false;
        textBox3.Enabled = false;
        textBox4.Enabled = false;
        button2.Enabled = false;
        button3.Enabled = false;
        button4.Enabled = false;
        // load dữ liệu lên comboBox
        SqlConnection con = new
        SqlConnection("server=.;database=sachdb;integrated
        security=true;");
```

```

        SqlDataAdapter da = new SqlDataAdapter("select * from
categories", con);
        DataTable tb = new DataTable();
        da.Fill(tb);
        comboBox1.DataSource = tb;
        comboBox1.DisplayMember = "categoryname";
        comboBox1.ValueMember = "categoryid";
    }

private void button1_Click(object sender, EventArgs e)
{
    textBox1.Enabled = true;
    textBox2.Enabled = true;
    textBox3.Enabled = true;
    textBox4.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox1.Focus();
    button1.Enabled = false;
    button3.Enabled = false;
    button2.Enabled = true;
    button4.Enabled = true;
}

private void button2_Click(object sender, EventArgs e)
{
    SqlConnection con = new
SqlConnection("server=.;database=sachdb;integrated
security=true;");
    SqlCommand cmd = new SqlCommand("insert into products
values('" + textBox1.Text + "\",'" + textBox2.Text +
"','" + textBox3.Text + "','" + textBox4.Text +
"','"+comboBox1.SelectedValue.ToString()+"')", con);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
    load_data();
}

private void button3_Click(object sender, EventArgs e)
{
    DialogResult kq = MessageBox.Show("ban muon xoa khong
?", "tieu de", MessageBoxButtons.YesNo);
    if (kq == System.Windows.Forms.DialogResult.Yes)
    {
        SqlConnection con = new
SqlConnection("server=.;database=sachdb;integrat
ed security=true;");
    }
}

```

```

        SqlCommand cmd = new SqlCommand("delete from
products where productcode = '" + textBox1.Text
+ "'", con);
con.Open();
cmd.ExecuteNonQuery();
con.Close();
load_data();
    }
}

private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    button3.Enabled = true;
    button4.Enabled = true;
}

private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
    dataGridView1_CellClick(sender, e);
}

private void button5_Click(object sender, EventArgs e)
{
    SqlConnection con = new
SqlConnection("server=.;database=sachdb;integrated
security=true;");

    //MessageBox.Show(textBox1.Text);
    //MessageBox.Show(textBox2.Text);
    //MessageBox.Show(textBox3.Text);
    //MessageBox.Show(textBox4.Text);
    //MessageBox.Show(comboBox1.SelectedValue.ToString())
    ;
    //string sql = "update products set description='" +
textBox2.Text + "',unitprice='" + textBox3.Text +
    "',OnHandQuantity='" + textBox4.Text + "',categoryid='" + c
omboBox1.SelectedValue.ToString() + "' where
productcode='" + textBox1.Text + "'";
    //MessageBox.Show(sql);
    SqlCommand cmd = new SqlCommand("update products set
description=@description,unitprice=@unitprice,OnHandQ
uantity=@onhandquantity,categoryid=@categoryid where
productcode=@productcode", con);
    cmd.Parameters.AddWithValue("@description",
textBox2.Text);
    cmd.Parameters.AddWithValue("@unitprice",
textBox3.Text);
    cmd.Parameters.AddWithValue("@OnHandQuantity",
textBox4.Text);

```

```

cmd.Parameters.AddWithValue("@categoryid",
comboBox1.SelectedValue.ToString());
cmd.Parameters.AddWithValue("@productcode",
textBox1.Text);
con.Open();
cmd.ExecuteNonQuery();
con.Close();
load_data();
}

private void comboBox1_SelectedIndexChanged(object sender,
EventArgs e)
{
    //SqlConnection con = new
SqlConnection("server=.;database=sachdb;integrated
security=true;");
//SqlDataAdapter da = new SqlDataAdapter("select *
from products where categoryid=@categoryid", con);
//da.SelectCommand.Parameters.AddWithValue("@category
id",comboBox1.SelectedValue);
//DataSet ds = new DataSet();
//da.Fill(ds);
//dataGridView1.DataSource = ds.Tables[0];
}
}

```

Bài 2: Viết chương trình thống kê sinh viên theo Môn học bằng C# với database SQL trong đó:

- Load dữ liệu từ file SQL vào Visual Studio
- Click hiển thị thông tin từ ComboBox sẽ hiện lên Textbox và DataGridView
- Có sử dụng Class để kết nối

	Mã SV	Họ SV	Tên SV	Ngày sinh	Điểm
▶	s001	Tran Minh	Son	01/05/1985	4
	s001	Tran Minh	Son	01/05/1985	6
	s007	Phan Thi	Ha	03/07/1988	2
	s007	Phan Thi	Ha	03/07/1988	9
	s008	Tran The	Dung	21/10/1985	7

Hướng dẫn:

- Xây dựng lớp kết nối

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;

namespace _39_SQLTamGa_ThongKeTheoMonHoc
{
    public class KetNoiDuLieu
    {
        public SqlConnection cnn = new SqlConnection
            ("Data Source=(local);Initial Catalog=QLSV;Integrated Security=True");

        public void myconnect()
        { cnn.Open(); }

        public void myclose()
        { cnn.Close(); }

        // www.tamga.tk www.c10mt.tk www.c10maytinh.tk
        public DataTable taobang(string sql)
        {
            DataTable dt = new DataTable();
            SqlDataAdapter ds = new SqlDataAdapter(sql, cnn);
            ds.Fill(dt);
            return (dt);
        }
    }
}

```

Các hàm trong Form

```
public Form1()
{
    InitializeComponent();
}

KetNoiDulieu kn = new KetNoiDulieu();

private void Form1_Load_1(object sender, EventArgs e)
{
    kn.myconnect();
    string sql = "SELECT * FROM MONHOC";
    cbMaMon.DataSource = kn.taobang(sql);
    cbMaMon.DisplayMember = "MAMH";
}

private void btThoat_Click(object sender, EventArgs e)
{
    this.Close();
    kn.myclose();
}

// www.tamga.tk www.c10mt.tk www.c10maytinhtk
private void cbMaMon_SelectedIndexChanged_1(object sender, EventArgs e)
{
    // load dữ liệu từ combobox xuống text
    string s = "select * from monhoc where mamh='" + cbMaMon.Text + "'";
    DataTable d = kn.taobang(s);
    foreach (DataRow hang in d.Rows)
        KhungTenMon.Text = hang["TENMH"].ToString();
    foreach (DataRow hang in d.Rows)
        KhungSoTiet.Text = hang["SOTIET"].ToString();

    // load dữ liệu lên DataGridView
    string s2 = "select sv.masv,hosv,tensv,ngaysinh,diem "+
        "from SinhVien SV,KetQua KQ " +
        "where (SV.masv = KQ.masv) and (KQ.mamh= '" + cbMaMon.Text + "')";
    dataGridViewMH.DataSource = kn.taobang(s2);
}
}
```

Bài 3: Viết chương trình thống kê sinh viên theo Khoa bằng C# với database SQL trong đó:

- Load dữ liệu từ file SQL vào Visual Studio
- Click hiện thông tin từ ComboBox sẽ hiện lên Textbox và DataGridView
- Có sử dụng lớp để kết nối dữ liệu

Form1

THỐNG KÊ SỐ SINH VIÊN THEO KHOA

Mã Khoa: Tên Khoa:

www.tamga.tk
 www.c10mt.tk
 www.c10maytinh.tk

	Mã SV	Họ SV	Tên SV	Ngày Sinh
▶	s001	Tran Minh	Son	01/05/1985
	s002	Nguyen Quoc	Bao	16/05/1986
	s006	Nguyen Thi	Lam	11/11/1984
	s007	Phan Thi	Ha	03/07/1988
	s008	Tran The	Dung	21/10/1985
	s001	Tam	Ca	24/10/2012

Tổng số Sinh Viên:

```

/*
TamGa
www.tamga.tk www.c10mt.tk www.c10maytinh.tk
www.tamga85.multiply.com
Phone: 01283.98.69.98 Email : tamgaalbum@yahoo.com
*/
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data.SqlClient;
using System.Data;

namespace _40_SQLTamGa_ThongKeTheoKhoa
{
    public class KetNoiDuLieu
    {
        public SqlConnection cnn = new SqlConnection
            ("Data Source=(local);Initial Catalog=QLSV;Integrated Security=True");

        public void myconnect()
        { cnn.Open(); }
    }
}

```

```
Form1.cs  KetNoiDuLieu.cs  Form1.cs [Design]
_40_SQLTamGa_ThongKeTheoKhoa.KetNoiDuLieu  taobang(string sql)

public void myclose()
{ cnn.Close(); }

// www.tamga.tk www.c10mt.tk www.c10maytinhtk
public DataTable taobang(string sql)
{
    DataTable dt = new DataTable();
    SqlDataAdapter ds = new SqlDataAdapter(sql, cnn);
    ds.Fill(dt);
    return (dt);
}
}
```

```
namespace _40_SQLTamGa_ThongKeTheoKhoa
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            KetNoiDuLieu kn = new KetNoiDuLieu();

            private void btThoat_Click(object sender, EventArgs e)
            {
                this.Close();
                kn.myclose();
            }

            // www.tamga.tk www.c10mt.tk www.c10maytinhtk
            private void Form1_Load(object sender, EventArgs e)
            {
                kn.myconnect();
                string sql = "select * from Khoa";
                cbMaKhoa.DataSource = kn.taobang(sql);
                cbMaKhoa.DisplayMember = "MAKHOA";
            }

            private void cbMaKhoa_SelectedIndexChanged(object sender, EventArgs e)
            {
                // load dữ liệu từ combobox xuống text
                string s = "select * from khoa where makhoa='" + cbMaKhoa.Text + "'";
                DataTable d = kn.taobang(s);
                foreach (DataRow hang in d.Rows)
                    KhungTenKhoa.Text = hang["TENKHOA"].ToString();
            }
        }
    }
}
```

```

// load dữ liệu lên DataGridView
string s1 = "select masv, hosv, tensv, ngaysinh " +
            "from SinhVien SV, Khoa KH " +
            "where (SV.makhoa = KH.makhoa) " +
            "and (KH.makhoa= '" + cbMaKhoa.Text + "')";
dataGridViewKhoa.DataSource = kn.taobang(s1);

string s2 = "select count(*) from SinhVien " +
            "where makhoa='" + cbMaKhoa.Text + "' group by makhoa";
KhungTong.Text = (dataGridViewKhoa.Rows.Count).ToString();
}

```

Bài 4. Tạo 1 ứng dụng đơn giản để quản lý sinh viên. Giao diện được thiết kế như hình

Student Name	Birth Day	Address
Hoang Tuan Dat	03/10/1985	Quang Nam
Phuong Thi Nha	23/06/1988	Dong Anh
Ngo Xuan Phu	09/10/1987	Ha Noi
Dinh Duc Thanh	05/09/1988	Ha Tay
Manh Hoai Thu	22/08/1987	Nhon

Các bạn thiết kế 1 ComboBox để hiển thị tên lớp, 1 textBox hiển thị tên giáo viên chủ nhiệm, 1 textBox hiển thị số sinh viên có trong lớp.. 1 listBox hiển thị thông tin chi tiết về sinh viên bao gồm tên, ngày sinh, địa chỉ. Thiết kế cơ sở dữ liệu như sau:

Bảng Student

THAOTRINH.Stude...bo.StudentClass		THAOTRINH.Stude...nt - dbo.Student	
Column Name	Data Type	Allow Nulls	
stuNo	varchar(5)	<input type="checkbox"/>	
stuName	varchar(50)	<input checked="" type="checkbox"/>	
classNo	varchar(50)	<input type="checkbox"/>	
stuYear	datetime	<input checked="" type="checkbox"/>	
address	varchar(100)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Bảng StudentClass

THAOTRINH.Stude...bo.StudentClass		THAOTRINH.Stude...nt - dbo.Student	
	Column Name	Data Type	Allow Nulls
▶	classNo	varchar(50)	<input type="checkbox"/>
	className	varchar(30)	<input type="checkbox"/>
	totalStudent	int	<input checked="" type="checkbox"/>
	homeroomTeacher	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

BÀI TẬP THỰC HÀNH 8: THIẾT KẾ REPORT

1. Mục tiêu

- Cách tạo đối tượng Report trong C#.
- Cách kết nối CSDL trong Report.
- Sử dụng report để thiết kế báo cáo

2. Nội dung

Bài 1:

The screenshot shows a window titled "Dynamic Report". At the top, there is a text input field labeled "Tên sinh viên xin nghỉ:" containing the text "Nguyễn Văn Hậu", followed by an "OK" button. Below this is a toolbar with various icons. A tab labeled "Main Report" is active. The main content area displays a report with the following elements:

- Header: "Cộng hòa xã hội chủ nghĩa Việt Nam" in red, followed by "Độc lập - tự do - hạnh phúc" in black.
- Logo: A yellow shield-shaped logo with a green emblem inside, located in the top right corner.
- Title: "ĐƠN XIN NGHỈ HỌC" in bold black text, centered.
- Text: "Em tên là: Nguyễn Văn Hậu" in black text, located on the left side.

The status bar at the bottom shows "Current Page No.: 1", "Total Page No.: 1", and "Zoom Factor: 100%".

Khi người dùng nhấn btOK thì thiết kế Report động với nội dung của textbox được truyền xuống Report

Hướng dẫn thực hiện:

- Thiết kế Report có giao diện như hình bên dưới:

▼ Section1 (Report Header)

Cộng hòa xã hội chủ nghĩa Việt Nam

Độc lập - tự do - hạnh phúc



ĐƠN XIN NGHỈ HỌC

▼ Section2 (Page Header)

Em tên là:

→ txtTenSinhVien

▼ Section3 (Details)

▼ Section4 (Report Footer)

▼ Section5 (Page Footer)

Trên Report có một TextObject có tên txtTenSinhvien dùng để đón nhận kết quả truyền từ form sang.

- Thiết kế form frmDynamicReport: sử dụng điều khiển Container chia form làm 2 phần (2 panel)
 - Panel 1: Chứa Một textbox và một nútton Button (xem hình vẽ).
 - Panel 2: chứa điều khiển crystalReportViewer1 để hiển thị report thiết kế ở trên.
- Viết sự kiện cho btOK

```
using CrystalDecisions.CrystalReports;
using CrystalDecisions.CrystalReports.Engine;
using CrystalDecisions.CrystalReports.Engine.Migration;

namespace CrystalReport2008
{
    public partial class frmDynamicReport : Form
    {
        private void btOK_Click(object sender, EventArgs e)
        {
            crDynamicReport rpt = new crDynamicReport();

            //Truyền dữ liệu

            ((TextObject)rpt.ReportDefinition.ReportObjects["txtTenSinhVien"]).Text = txtTenSinhVien.Text;
        }
    }
}
```

```

        //Refresh data
        crystalReportViewer1.ReportSource = rpt;
        crystalReportViewer1.RefreshReport();
    }
}
}

```

Bài 2: Trường Đại học Sư phạm Kỹ thuật Hưng yên sử dụng SQL Server để lưu trữ danh sách sinh viên và bảng điểm của học sinh Dữ liệu được thiết kế như sau:

+MonHoc(Ma,ten)

+LopHoc(Ma,ten)

+SV(Ma,Ten,Malop,SOBienLai)

+Điểm (masv,mamon,diem)

Viết chương trình hiển thị danh sách sinh viên theo lớp đang chọn (từ combo).

+ Chú ý: Danh sách sinh viên được sắp xếp theo trình tự (tên, và họ).

Lưu cập nhật điểm của toàn bộ lớp vào database UTEHY

Mã	Họ	Tên	Điểm
12	Nguyễn thị	Đào	
323	Lê	Đào	
354	(null)	Hải	
236	Đào thị	Hoa	
66	Nguyễn	Quán	
5	Lý văn	Sơn	
13	Lê trần	Sơn	
135	Trưởng	Thủy	
67	Lê	Trúc	
*			

+ Tạo form report(trên word hoặc excel) dùng để tạo report cho danh sách Sv và bảng điểm của lớp.

DANH SÁCH HỌC VIÊN LỚP CHỨNG CHỈ A

Lớp: A104B

Khai Giảng: 07/09/2009

Giờ học: từ 18h đến 21h

Ngày học: Thứ 2,4,6

Phòng học: H6.3

Giáo Viên: Trần Thụy Thiên Kim

STT	Họ	Tên	Số biên lai	Ghi chú
1	Hồ Thị Trâm	Anh	013568	
2	Võ Thị Kim	Chuyên	013550	
3	Ngô Thị Thanh	Cúc	013567	
4	Nguyễn Thị	Đào	013571	
5	Vân Thị Hồng	Diễm	013537	
6	Đỗ Thị Ngọc	Duyên	013548	
7	Võ Thị Hồng	Gấm	013538	
8	Phạm Ngọc	Hoa	013532	
9	Nguyễn Thị Xuân	Hương	013527	
10	Nguyễn Thị Thu	Hường	013534	
11	Phan Thị Diệu	Huyền	013546	
12	Nguyễn Phi	Khải	013549	
13	Trần Thị Hồng	Linh	013531	
14	Phin Quốc	Long	013551	
15	Cao Cát	Lương	013584	
16	Đinh Thị Ngọc	Mỹ	013566	
17	Hoàng Thị	Nga	013535	
18	Nguyễn Thị Lê	Nga	013528	
19	Đặng Thị Bích	Ngọc	013565	
20	Lý Thành	Nhân	013533	
21	Lê Thị Tuyết	Nhung	013517	

---- HẾT ----