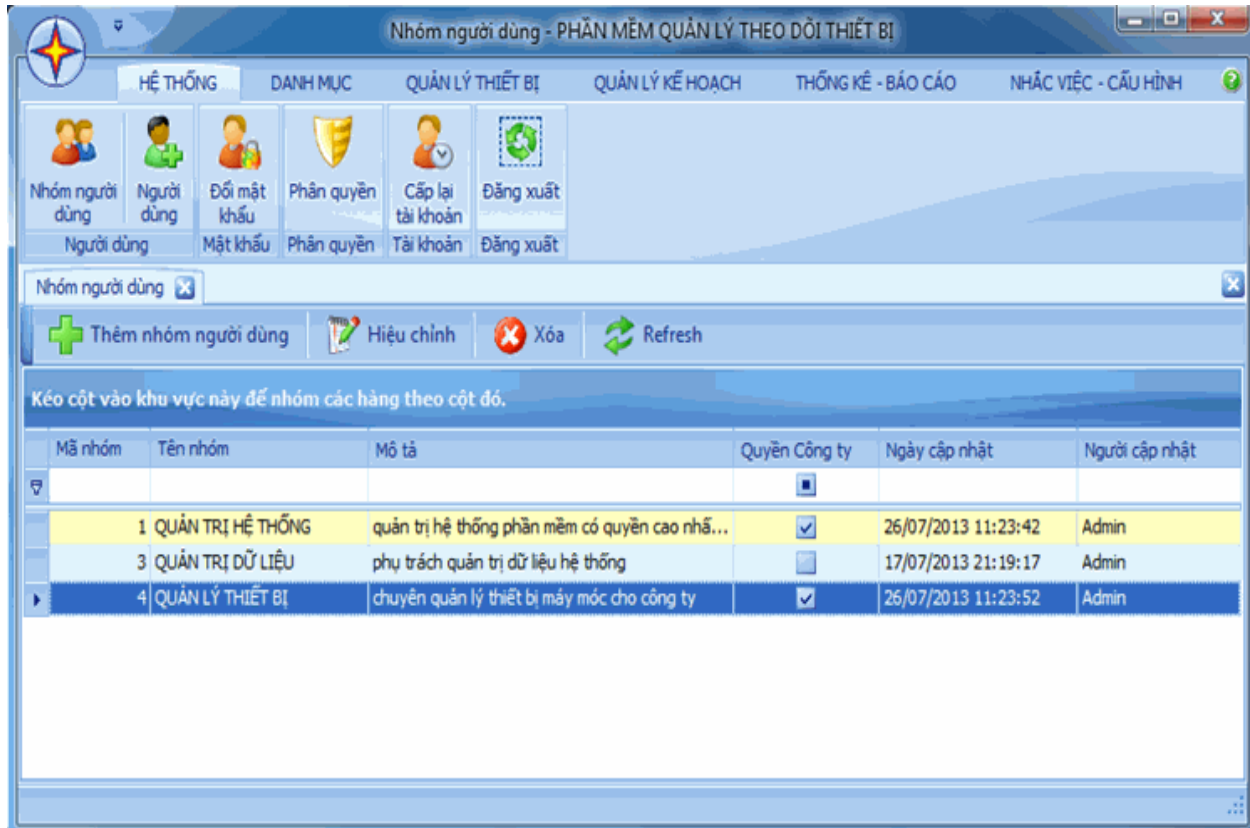
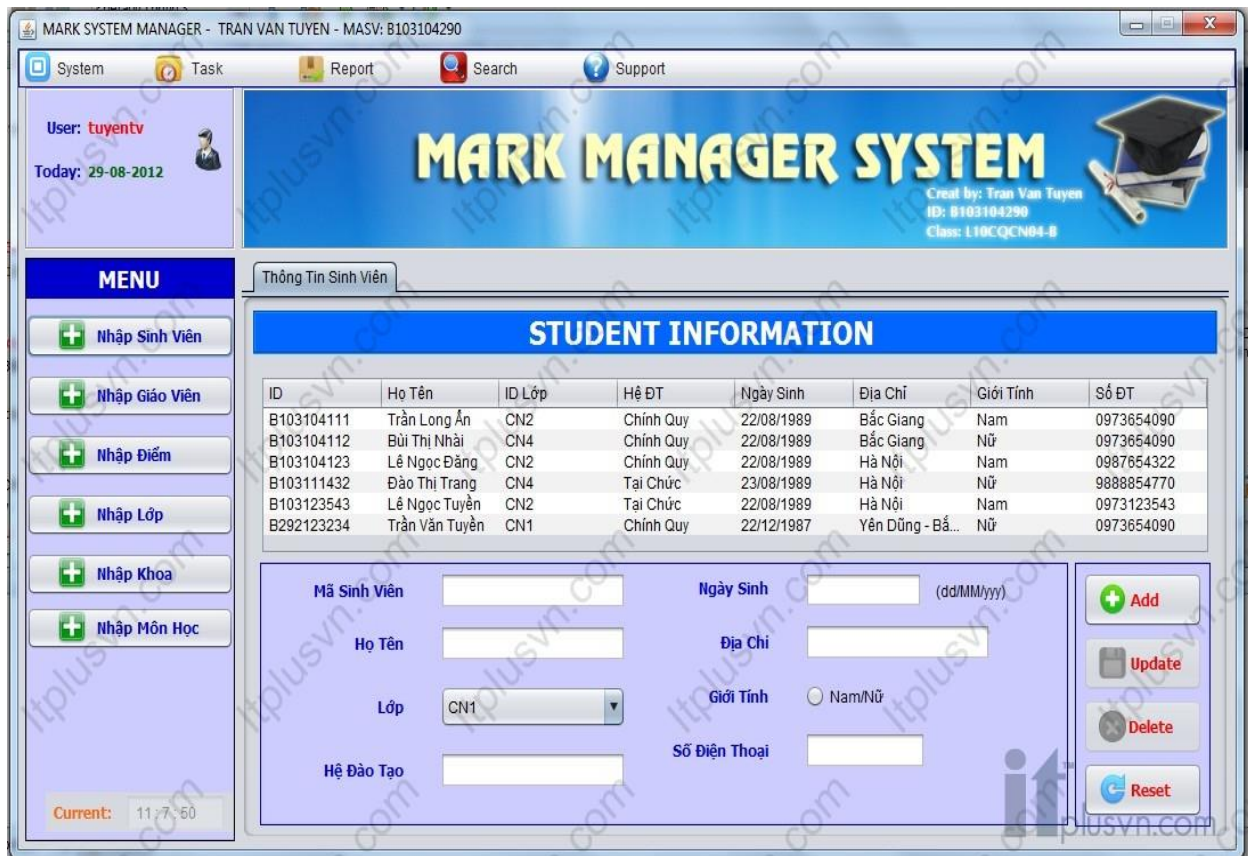


LẬP TRÌNH WINDOWS FORM C#

**Mục tiêu cuối cùng của khóa học OOP:
Viết phần mềm ứng dụng quản lý (C# hoặc Java)**



C#.NET



Java

Hướng đến mục tiêu khóa học
Android/iOS/Windows Phone

Các bước tạo Project:

File -> New -> Project

Bên cột trái chọn Visual C#, bên cột phải chọn Windows Form Application

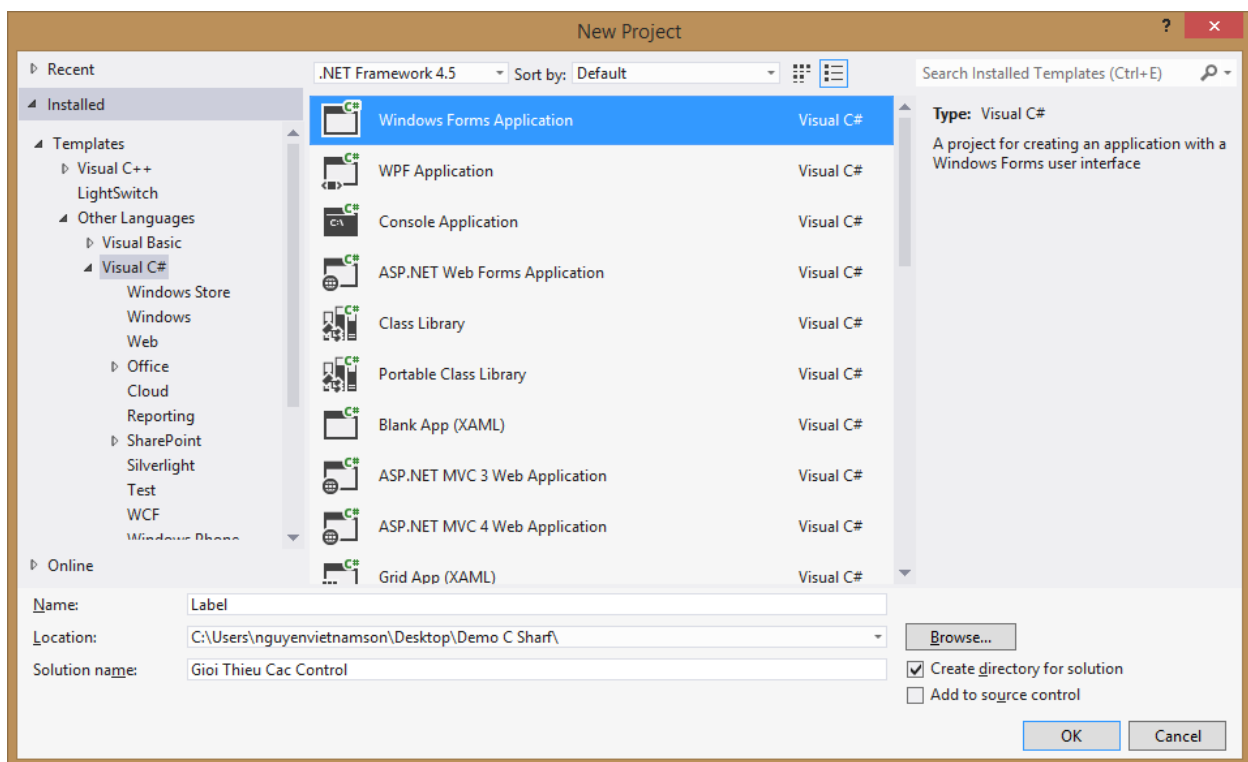
Ở dưới tùy chỉnh 3 mục:

Name: Mặc định sẽ là tên Project đầu tiên (có thể để có dấu được)

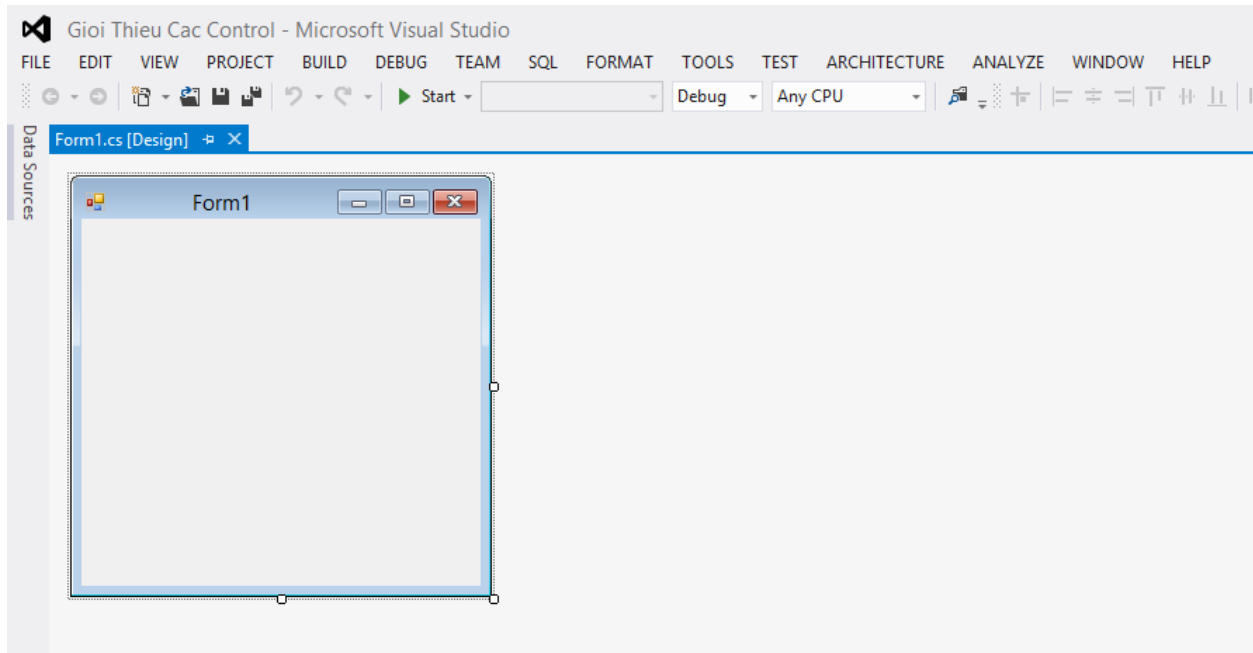
Location: Đường dẫn đến thư mục chương trình sẽ tạo ra

Solution Name: Tên Solution (1 Solution quản lý nhiều Project bên trong, mỗi Project là 1 bài, tên của Solution không được để có dấu)

Sau đó bấm OK để tạo.



Giao diện Form tạo ra sẽ như sau:

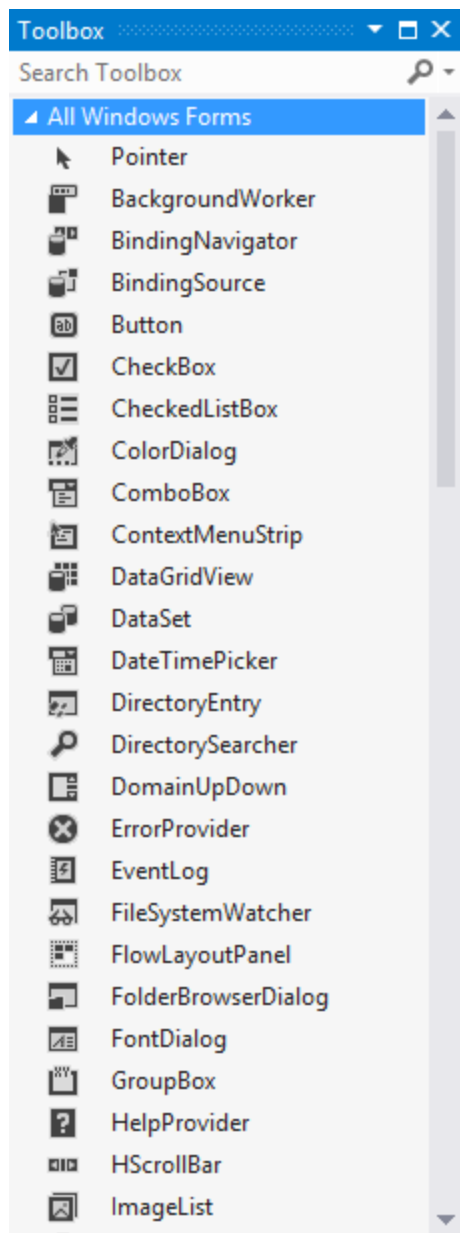


Nếu muốn thay đổi lại thuộc tính của Form, ta Click phải chuột lên Form chọn Properties và tùy chỉnh trong bảng Properties đó.

Nếu cần tùy chỉnh tạo thêm Project khác hoặc 1 số tác vụ khác thì chọn: View → Solution Explorer. Click chuột phải vào Solution chọn Add → New Project, điền đầy đủ thông tin rồi nhấn OK. Project mới được tạo ra muốn được cấp phát chạy thì Click chuột phải vào Project đó chọn Set As StartUp Project

Mở danh sách các công cụ Control đã được hỗ trợ sẵn trong Visual C#: View → Toolbox

Sau khi bảng danh sách các Control hiện ra, chọn All Windows Forms



Muốn đưa Control vào trong Form, ta Click chuột trái (click 1 lần) lên Control và di chuyển con trỏ chuột vào trong Form và click thêm lần nữa.

2 dạng tương tác với Control:

- + **Action:** Hành động thực hiện
- + **Outlet:** Hiển thị dữ liệu hoặc cho phép người dùng nhập dữ liệu vào

Nguyên tắc khi thao tác trên Control trong Visual C#

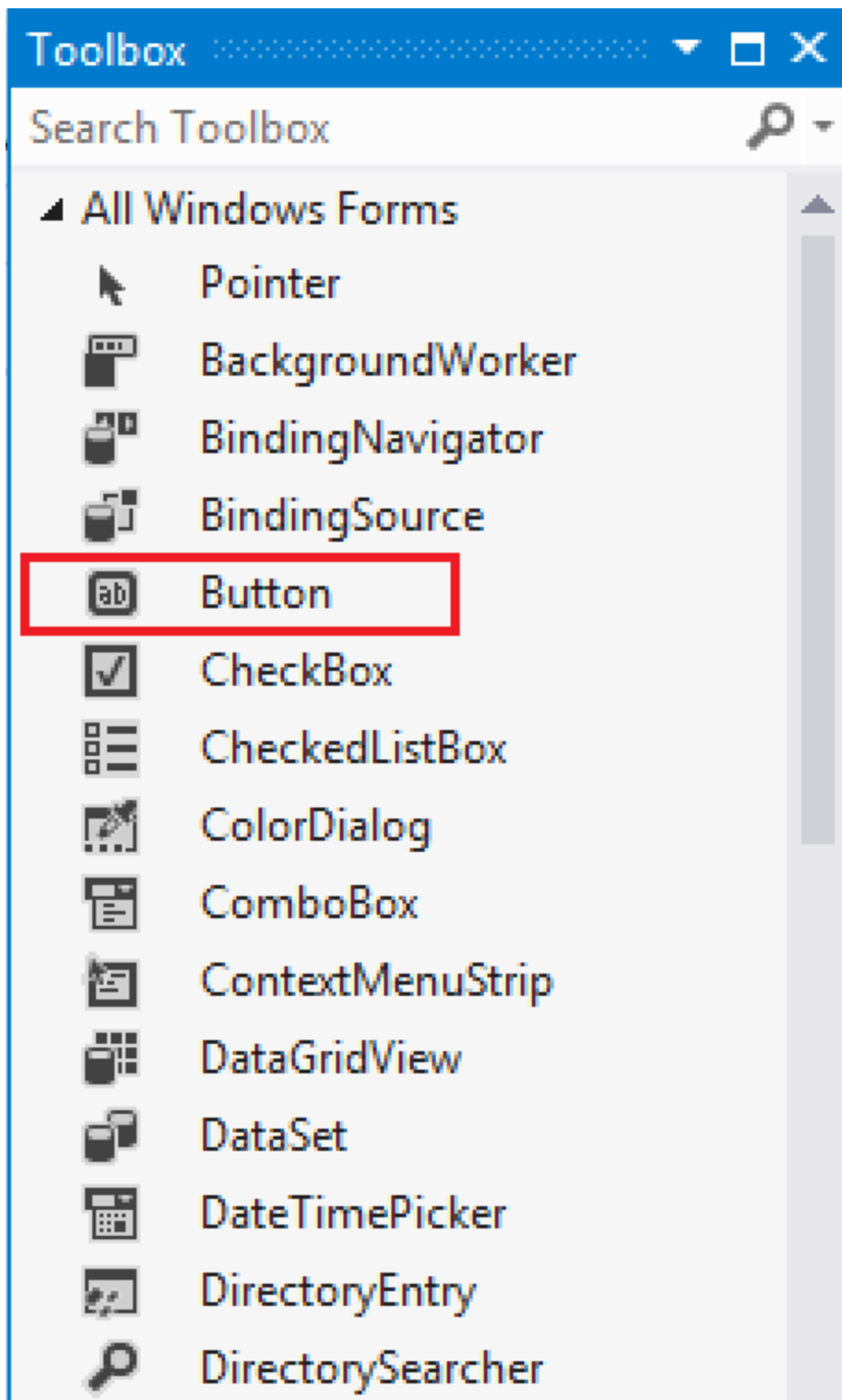
Khi đã kéo thả Control qua Form thì ta phải tùy chỉnh Properties cho Control đó, Click phải vào Control chọn Properties và đi đặt tên cho Control đó (việc đặt tên cho Control cũng giống như việc ta khai báo các biến, mục đích để sau này cần gọi tới Control thì ta chỉ cần gọi đúng tên Control đã đặt lúc ban đầu)

Các Outlet của Control sẽ nằm trong mục Properties (Mặc định)

Các Action của Control sẽ nằm trong mục bên cạnh (Có hình tia sét)

Giới thiệu 1 số Control trong Visual C#

1. Button



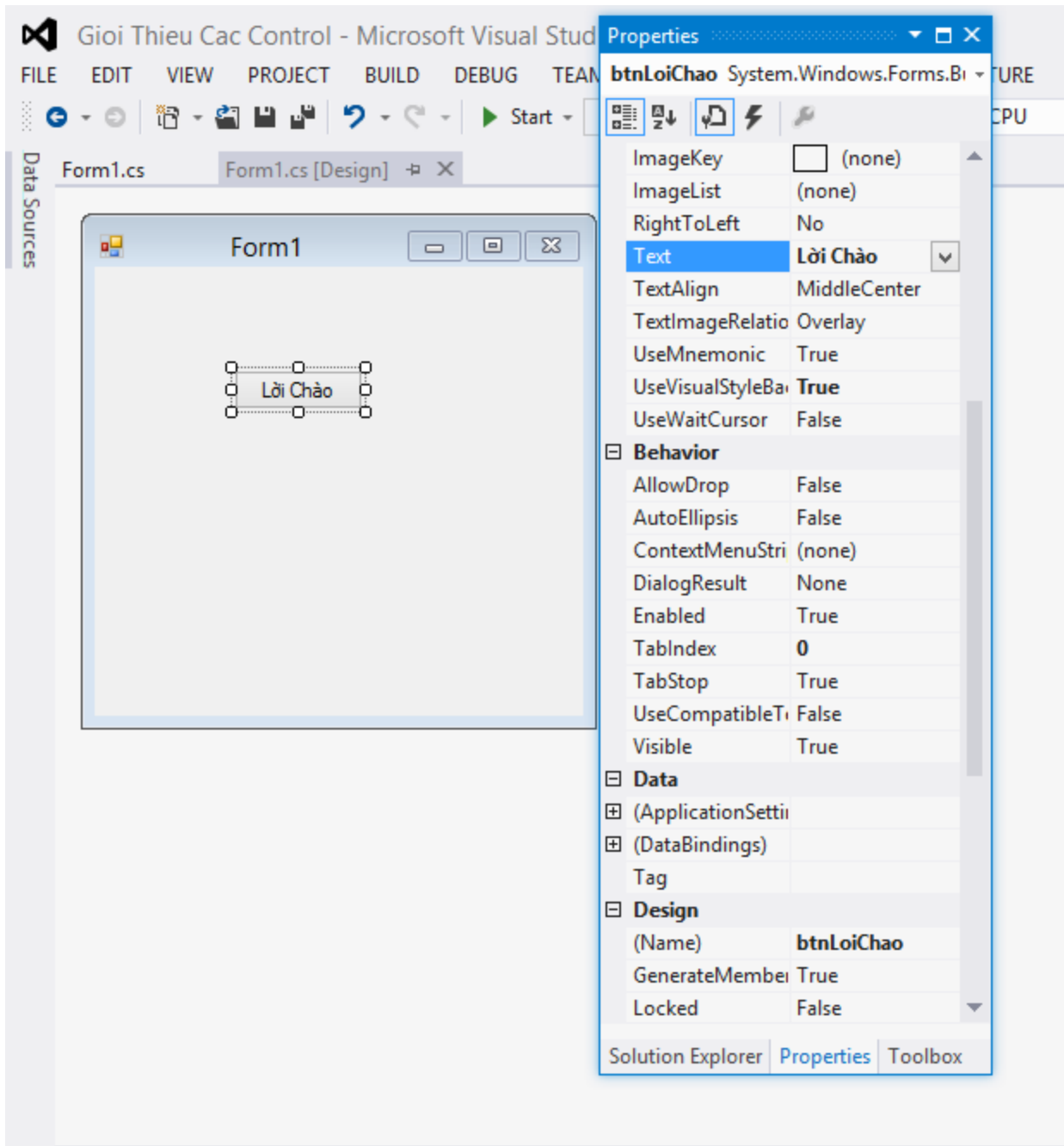
Nguyên tắc đặt tên: btn<Tên Tự Nghĩ ra>

Vd: btnCong, btnTru, btnNhan, btnChia, btnXuLy ...

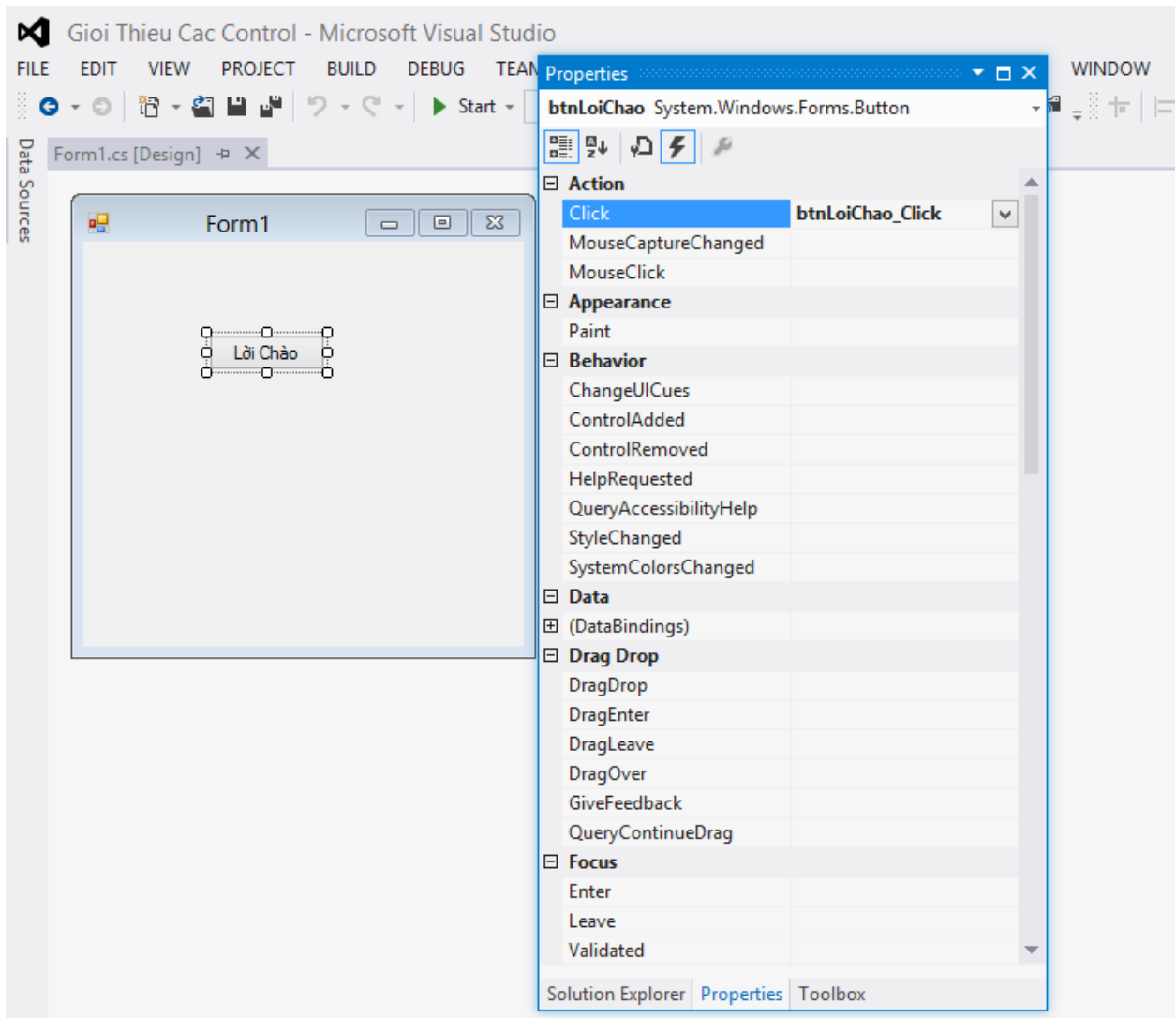
Sử dụng khi cần tạo ra sự tương tác giữa người dùng với chức năng xử lý

Vd: Khi người dùng nhấn vào nút sẽ xảy ra 1 hành động gì đó.

Mục Outlet của Button



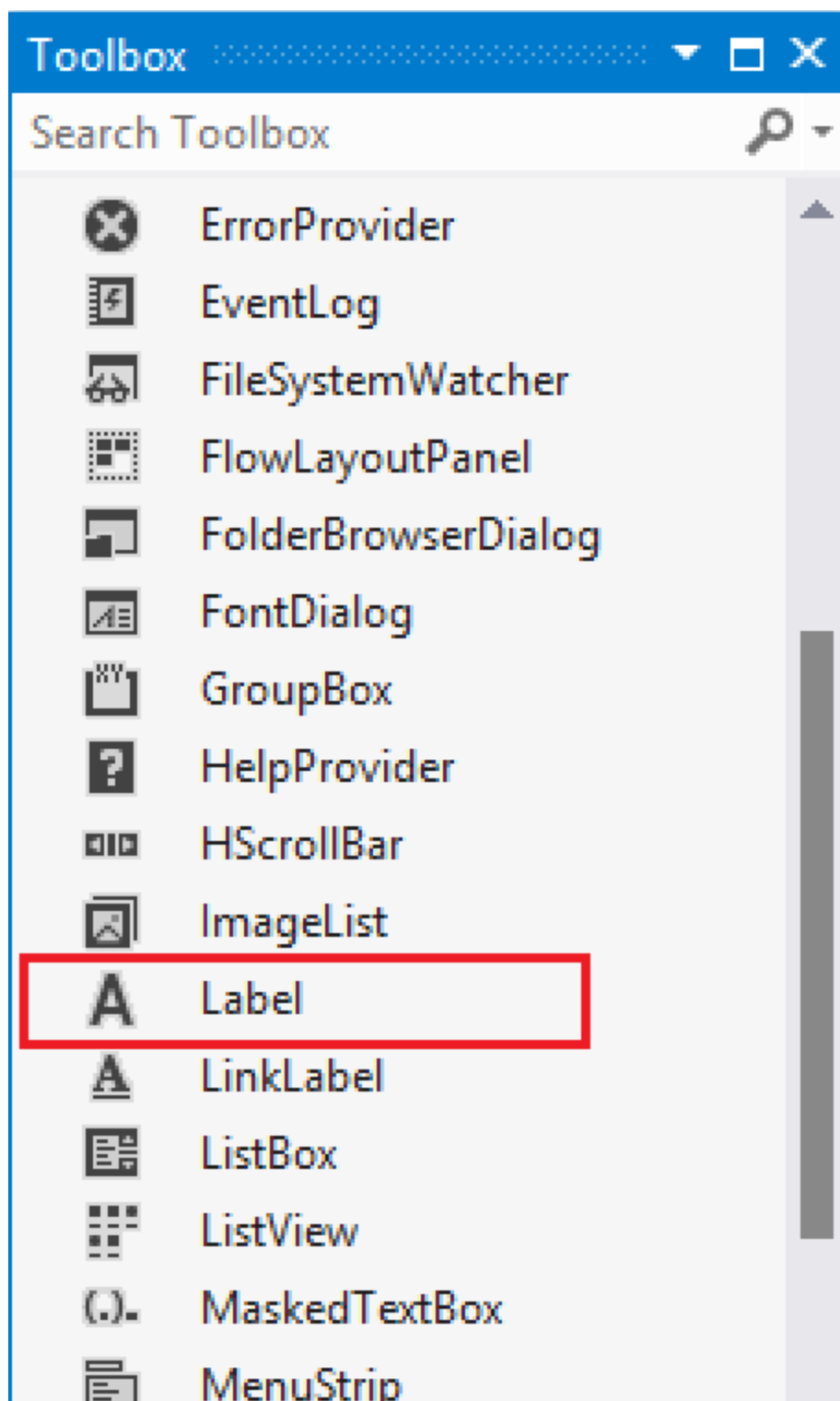
Mục Action của Button:



Sau khi đặt tên xong thì công việc còn lại là đi viết code cho Button bằng cách Click trái 2 lần vào Button thì tự động sẽ có 1 hàm được sinh ra, ta chỉ việc viết code trong hàm đó.

```
Form1.cs [Design]
Button.Form1 Form
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Button
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private void btnLoiChao_Click(object sender, EventArgs e)
21         {
22             MessageBox.Show("Chúc bạn một ngày vui vẻ", "Lời Chào");
23         }
24     }
25 }
```

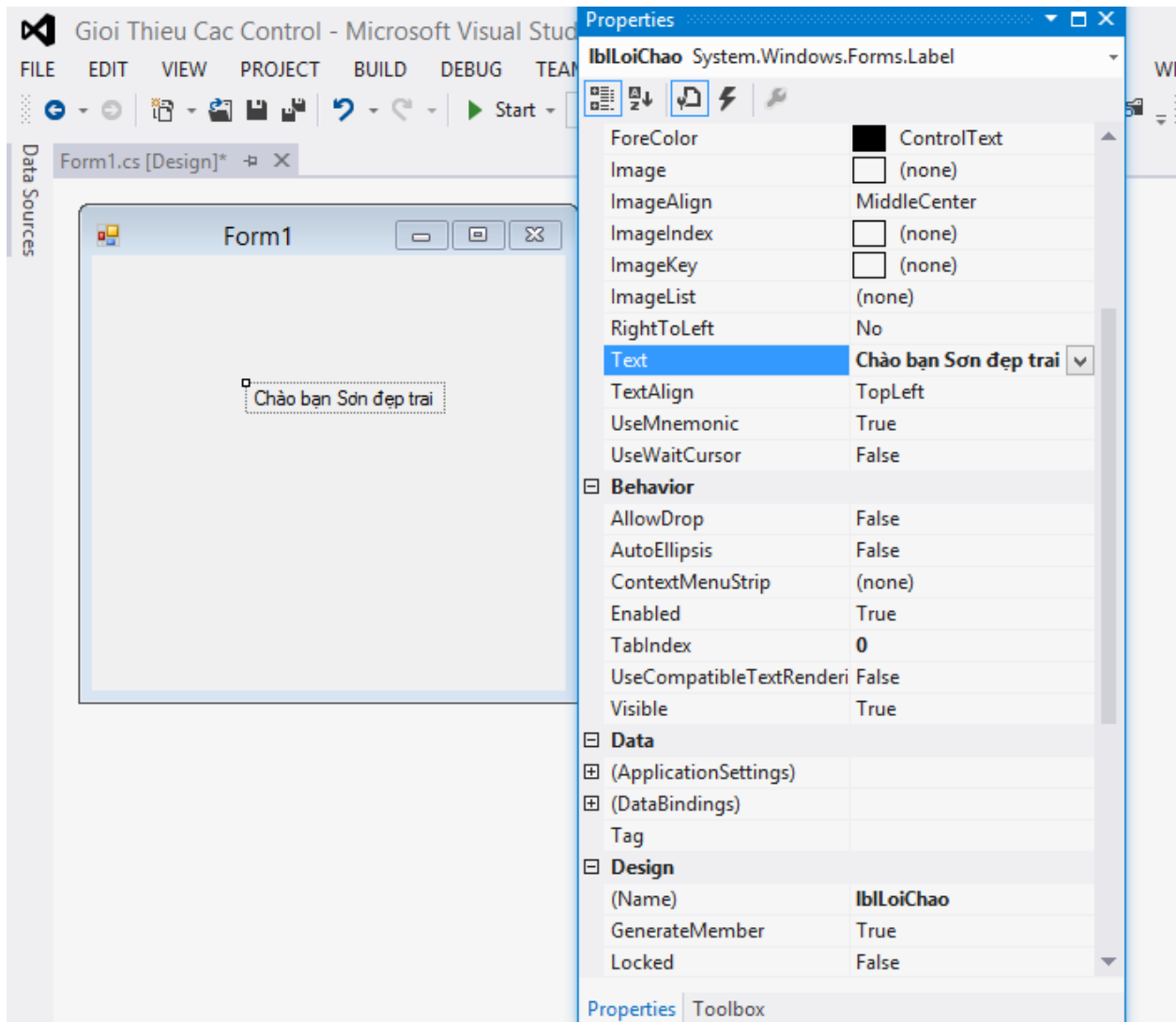
2. Label



Nguyên tắc đặt tên: lbl<Tên Tự Nghĩa ra>

Vd: lblGioiThieu, lblThongBao ...

Sử dụng để hiển thị ra màn hình 1 câu thông báo gì đó.

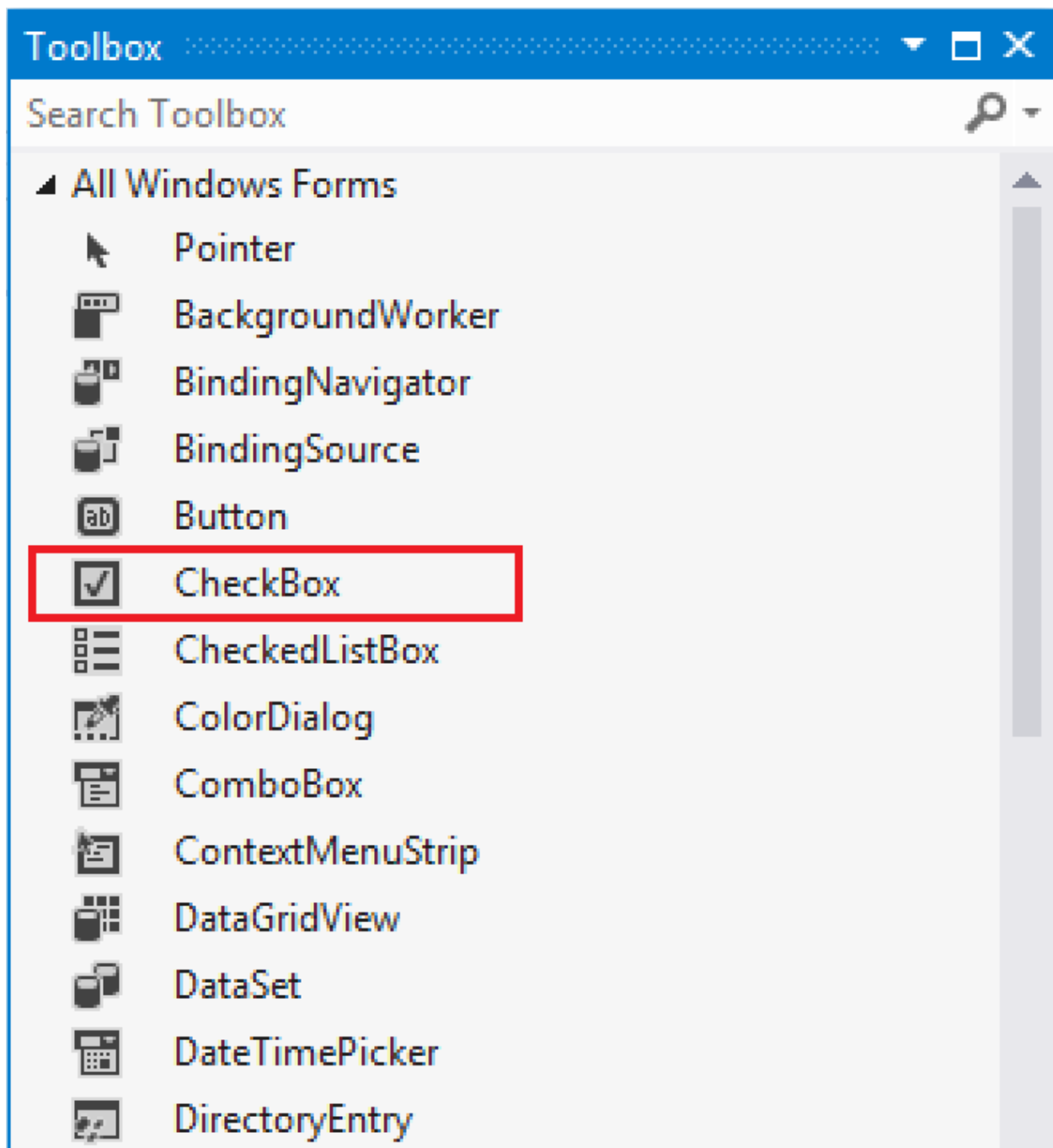


Nếu ta không muốn gán mặc định chuỗi text cho Label thì có thể gán cho nó bằng code như sau:

```
Form1.cs [Design]
Label.Form1
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Label
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private void Form1_Load(object sender, EventArgs e)
21         {
22             lblLoiChao.Text = "Bạn rất đẹp trai";
23         }
24     }
25 }
```

Form1_Load là sự kiện sẽ chạy đầu tiên khi Project được Run, Form1_Load chạy xong rồi Form mới hiển thị lên.

3. CheckBox



Nguyên tắc đặt tên: chk<Tên Tự Nghĩa ra>

Vd: chkGioiTinh, chkLuaChon ...

Sử dụng để cho người dùng chọn lựa điều gì đó.

Ví dụ: **Viết ứng dụng cho phép người dùng đánh giá giáo viên**

Giao diện như sau:

Đánh Giá Giáo Viên

Hãy nhận xét về Sơn đẹp trai

☒ Hài Hước ☐ Dễ Thương ☒ Hoạt Bát

Nhận Xét Nào

Kết Quả: Hài Hước Hoạt Bát

chkHaiHuoc, chkDeThuong, chkHoatBat

btnNhanXet

lblKetQua

Viết code xử lý cho nút Button

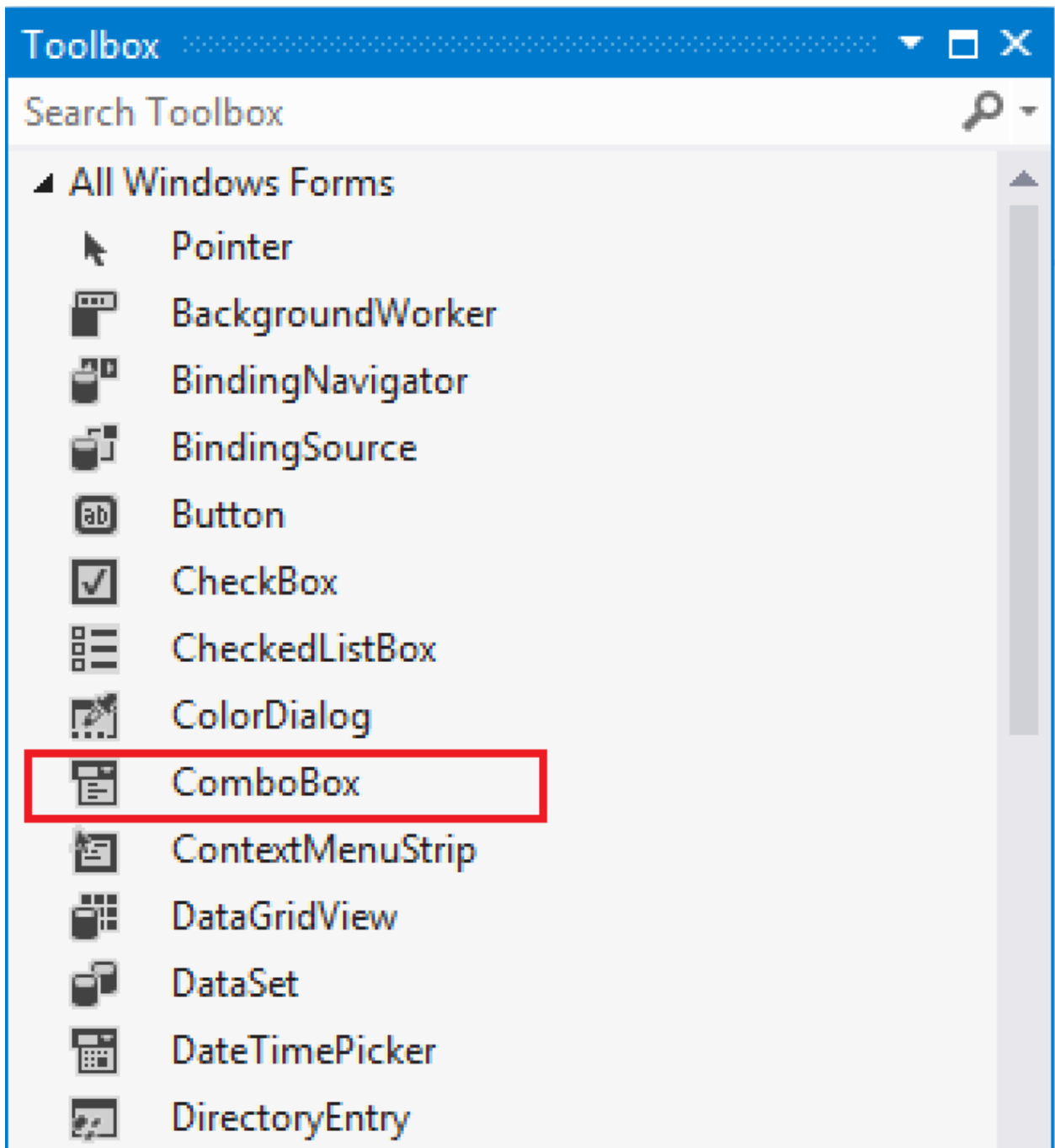
```
private void btnNhanXet_Click(object sender, EventArgs e)
{
    lblKetQua.Text = "Kết Quả: ";

    if (chkDeThuong.Checked)
    {
        lblKetQua.Text += chkDeThuong.Text + " ";
    }

    if (chkHaiHuoc.Checked)
    {
        lblKetQua.Text += chkHaiHuoc.Text + " ";
    }

    if (chkHoatBat.Checked)
    {
        lblKetQua.Text += chkHoatBat.Text + " ";
    }
}
```

4. ComboBox



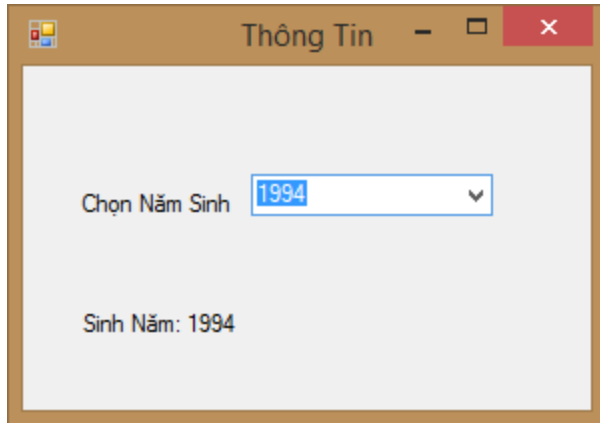
Nguyên tắc đặt tên: cmb<Tên Tự Nghĩ ra>

Vd: cmbMonHoc, cmbMonAn ...

Sử dụng để hiển thị bảng danh sách xổ ra từ trên xuống cho người dùng chọn lựa

Ví dụ: **Viết ứng dụng lựa chọn năm sinh**

Giao diện:



cmbNamSinh

lblNamSinh

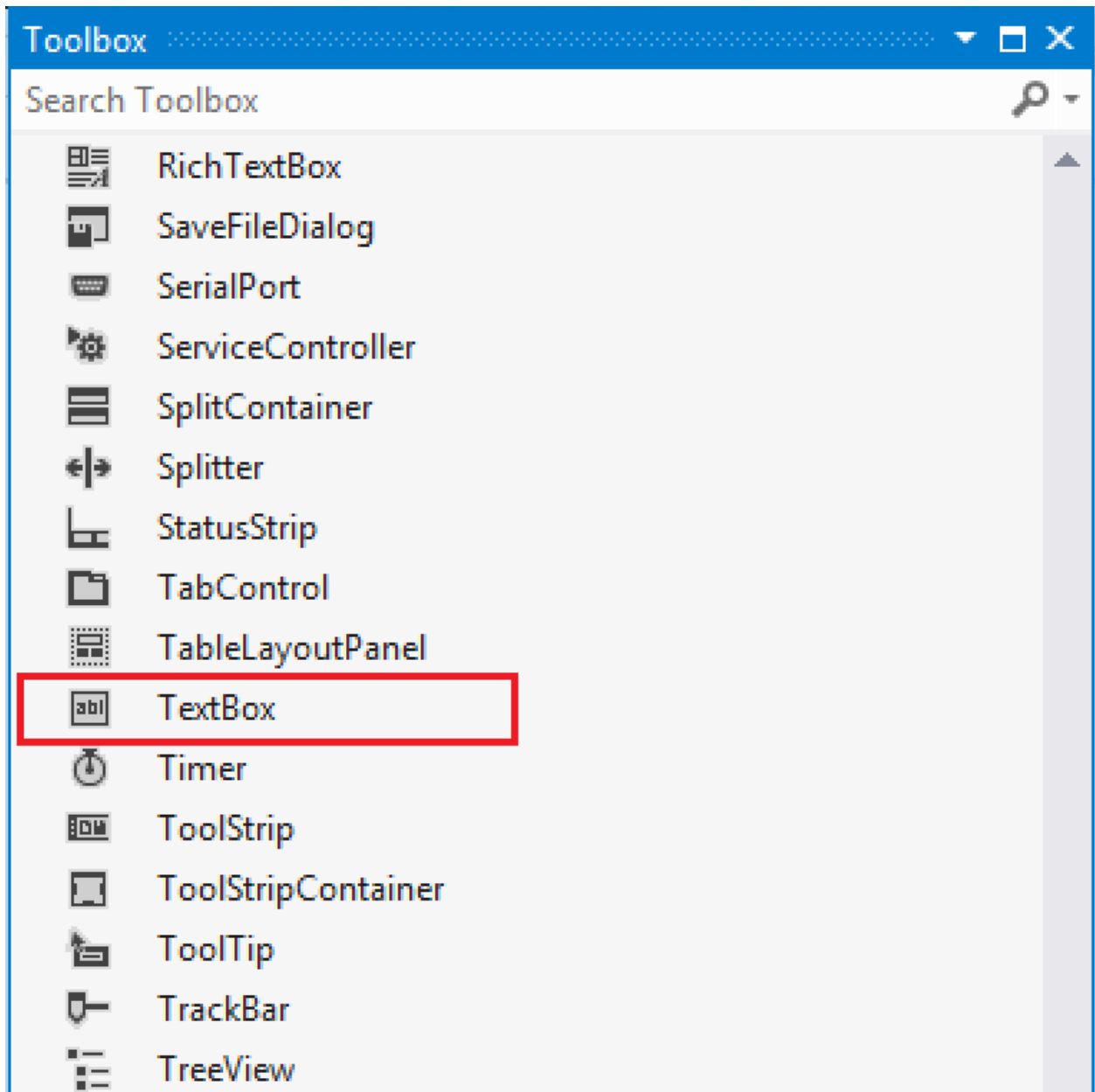
Viết code xử lý

```
private void Form1_Load(object sender, EventArgs e)
{
    // Đưa dữ liệu vào ComboBox
    for (int i = 1950; i <= 2010; ++i)
    {
        cmbNamSinh.Items.Add(i);
    }

    // Mặc định ComboBox sẽ hiển thị dữ liệu đầu tiên (Năm 1950)
    cmbNamSinh.SelectedIndex = 0;
}

private void cmbNamSinh_SelectionChangeCommitted(object sender, EventArgs e)
{
    lblNamSinh.Text = "Sinh Năm: " + cmbNamSinh.SelectedItem.ToString();
}
```

5. TextBox



Nguyên tắc đặt tên: txt<Tên Tự Nghĩa ra>

Vd: txtHoTen, txtNamSinh ...

Sử dụng để cho người dùng điền vào thông tin

Ví dụ: **Viết ứng dụng nhập vào họ tên và xuất ra màn hình lời chào tương ứng**

Giao diện:



Thông Tin

Nhập vào tên:

Chào Bạn: Sơn đẹp trai

txtTen

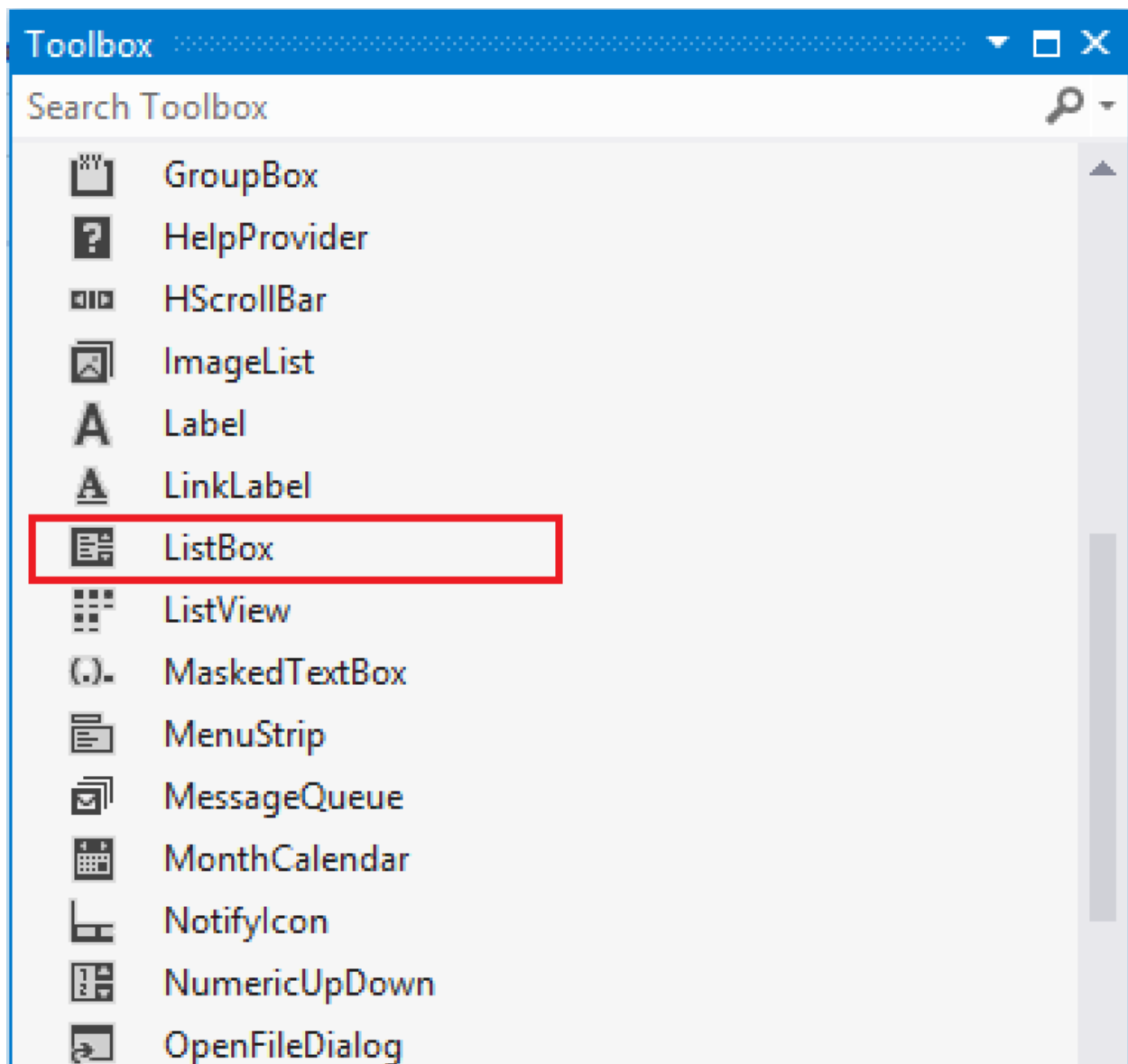
btnXacNhan

lblLoiChao

Viết code xử lý cho Button Xác Nhận

```
private void btnXacNhan_Click(object sender, EventArgs e)
{
    lblLoiChao.Text = "Chào Bạn: " + txtTen.Text;
}
```

6. ListBox



Nguyên tắc đặt tên: lsb<Tên Tự Nghĩa ra>

Vd: lsbMonHoc, lsbMonAn ...

Sử dụng để cho người dùng chọn lựa

Lưu ý: Có thể chọn nhiều lựa chọn 1 lúc (Mặc định ban đầu chỉ được chọn 1)

⇒ Ta tùy chỉnh trong Properties mục: SelectionMode

Trong mục có 4 lựa chọn

None: Không được chọn gì cả

One: Chỉ chọn một (Mặc định là chế độ này)

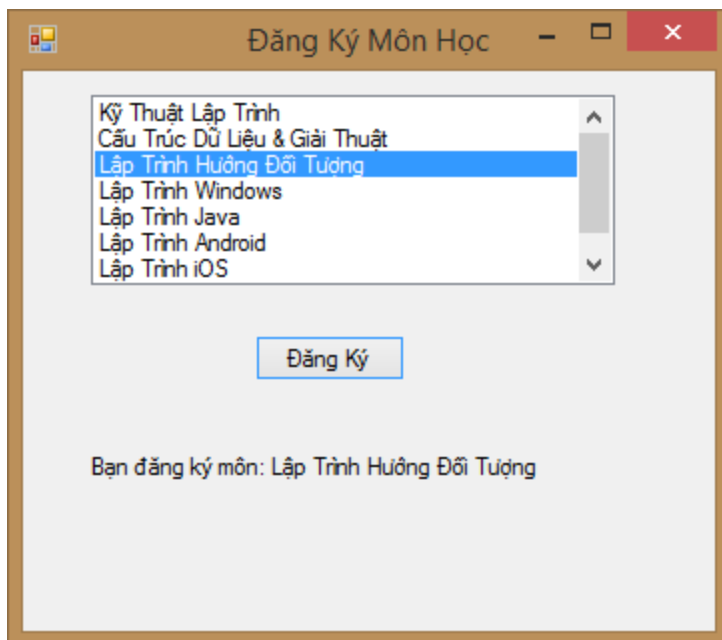
MultiSimple: Chọn được nhiều bằng cách Click nhiều lần, muốn bỏ chọn thì Click lại lần nữa.

MultiExtended: Chọn được nhiều bằng cách nhấn phím Ctrl để chọn từng cái hoặc nhấn Alt để chọn 1 khoảng hoặc kéo rê chuột

Ví dụ: Viết ứng dụng đăng ký môn học

Đề chế độ One

Giao diện:



lsbMonHoc

btnDangKy

lblKetQua

Viết code xử lý cho nút Button

```

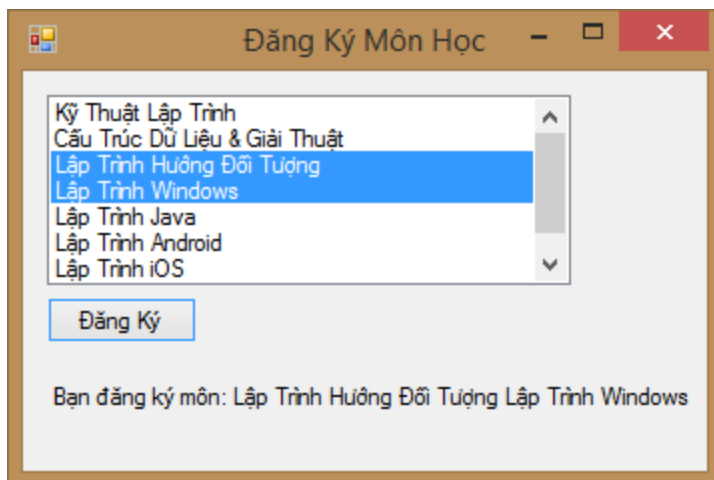
private void Form1_Load(object sender, EventArgs e)
{
    lsbMonHoc.Items.Add("Kỹ Thuật Lập Trình");
    lsbMonHoc.Items.Add("Cấu Trúc Dữ Liệu & Giải Thuật");
    lsbMonHoc.Items.Add("Lập Trình Hướng Đối Tượng");
    lsbMonHoc.Items.Add("Lập Trình Windows");
    lsbMonHoc.Items.Add("Lập Trình Java");
    lsbMonHoc.Items.Add("Lập Trình Android");
    lsbMonHoc.Items.Add("Lập Trình iOS");
    lsbMonHoc.Items.Add("Lập Trình Windows Phone");
}

private void btnDangKy_Click(object sender, EventArgs e)
{
    lblKetQua.Text = "Bạn đăng ký môn: " + lsbMonHoc.SelectedItem.ToString();
}

```

Đề chế độ MultiSimple

Giao diện:



Viết code xử lý cho nút Button

```

private void Form1_Load(object sender, EventArgs e)
{
    lsbMonHoc.Items.Add("Kỹ Thuật Lập Trình");
    lsbMonHoc.Items.Add("Cấu Trúc Dữ Liệu & Giải Thuật");
    lsbMonHoc.Items.Add("Lập Trình Hướng Đối Tượng");
    lsbMonHoc.Items.Add("Lập Trình Windows");
    lsbMonHoc.Items.Add("Lập Trình Java");
    lsbMonHoc.Items.Add("Lập Trình Android");
    lsbMonHoc.Items.Add("Lập Trình iOS");
    lsbMonHoc.Items.Add("Lập Trình Windows Phone");
}

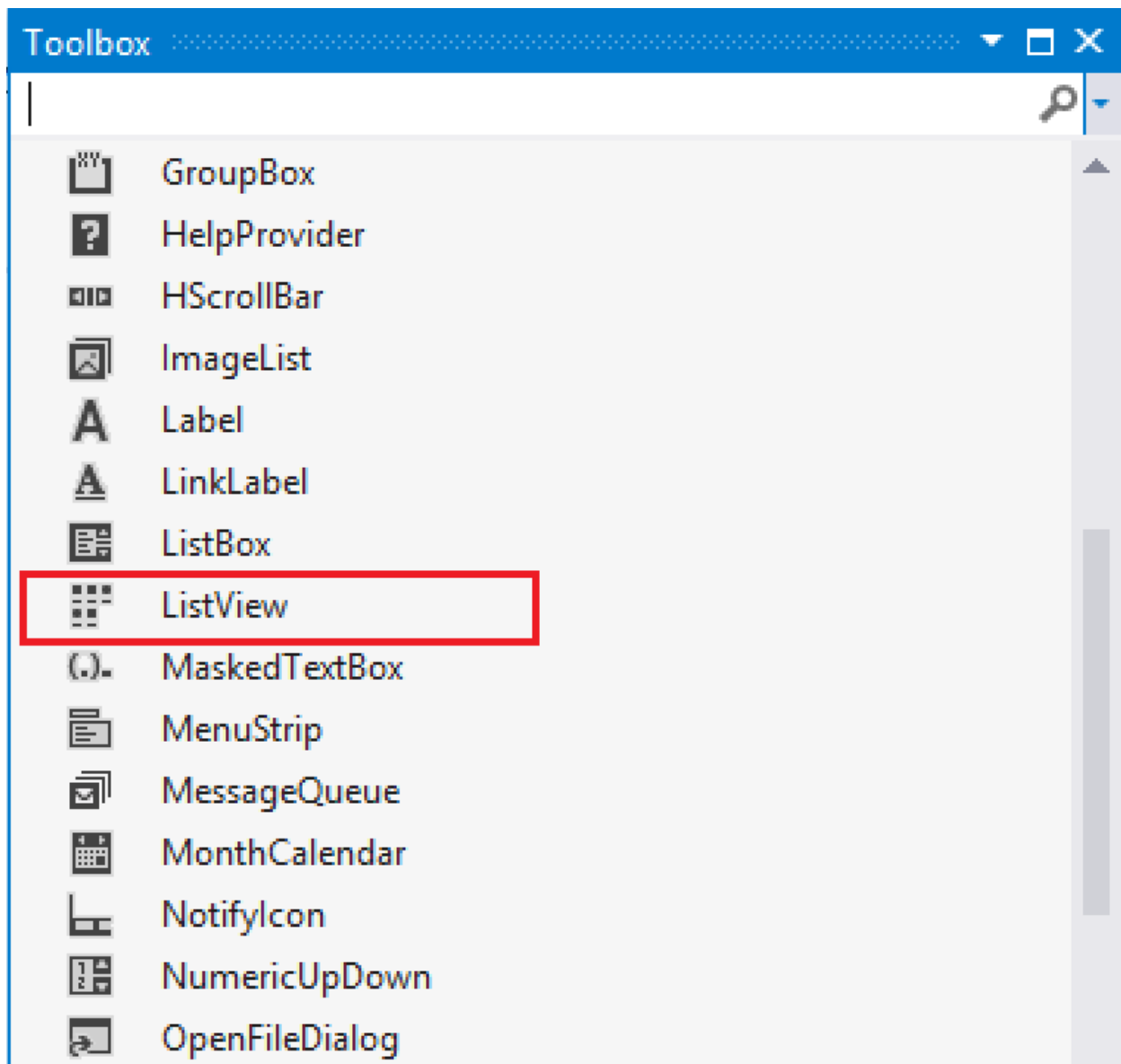
private void btnDangKy_Click(object sender, EventArgs e)
{
    lblKetQua.Text = "Bạn đăng ký môn: ";
    int count = lsbMonHoc.Items.Count;

    for (int i = 0; i < count; ++i)
    {
        if (lsbMonHoc.GetSelected(i) == true)
        {
            lblKetQua.Text += lsbMonHoc.Items[i].ToString() + " ";
        }
    }
}

```

7. ListView

ĐÂY LÀ CONTROL CỰC KỲ QUAN TRỌNG, ỨNG DỤNG RẤT NHIỀU



Nguyên tắc đặt tên: lsv<Tên Tự Nghĩa ra>

Vd: lsvThôngTin ...

Sử dụng để hiển thị dữ liệu cho người dùng thấy

Lưu ý: Có rất nhiều chế độ hiển thị, tuy nhiên ta thường chọn Details (dạng chi tiết)

Tùy chỉnh trong Properties phần View → Details

Muốn add các cột vào trong ListView ta có 2 cách

Cách 1: Vào Properties → Columns → Add và cài đặt các thông tin cho các cột, cần cài đặt thông tin độ rộng tương ứng (width)

Cách 2: Cài đặt trực tiếp bằng code

Muốn có lần kẻ ngang – dọc cho dễ nhìn thấy thì chỉnh GridLines → true

Muốn chọn hết nguyên 1 dòng – chỉnh FullRowSelect → true

Ví dụ: Viết ứng dụng nhập vào và hiển thị thông tin các sinh viên có trong lớp học

Lưu ý:

- Khi dữ liệu đã nhập xong thì nhấn nút Button => lập tức đưa dữ liệu đó vào ListView
- Khi danh sách đã có phần tử thì ta Click chuột trái 2 lần vào 1 dòng thì lập tức thông tin của sinh viên đó sẽ hiển thị trên Form.

Giao diện:

Viết code xử lý:

```

private void Form1_Load(object sender, EventArgs e)
{
    // Tạo ra các cột
    lsvDanhSachHocSinh.Columns.Add("Họ Tên", 140);
    lsvDanhSachHocSinh.Columns.Add("Quê Quán", 140);
    lsvDanhSachHocSinh.Columns.Add("Trường", 140);
}

private void btnThem_Click(object sender, EventArgs e)
{
    string[] arr = new string[3];
    ListViewItem item; // Add item vào ListView

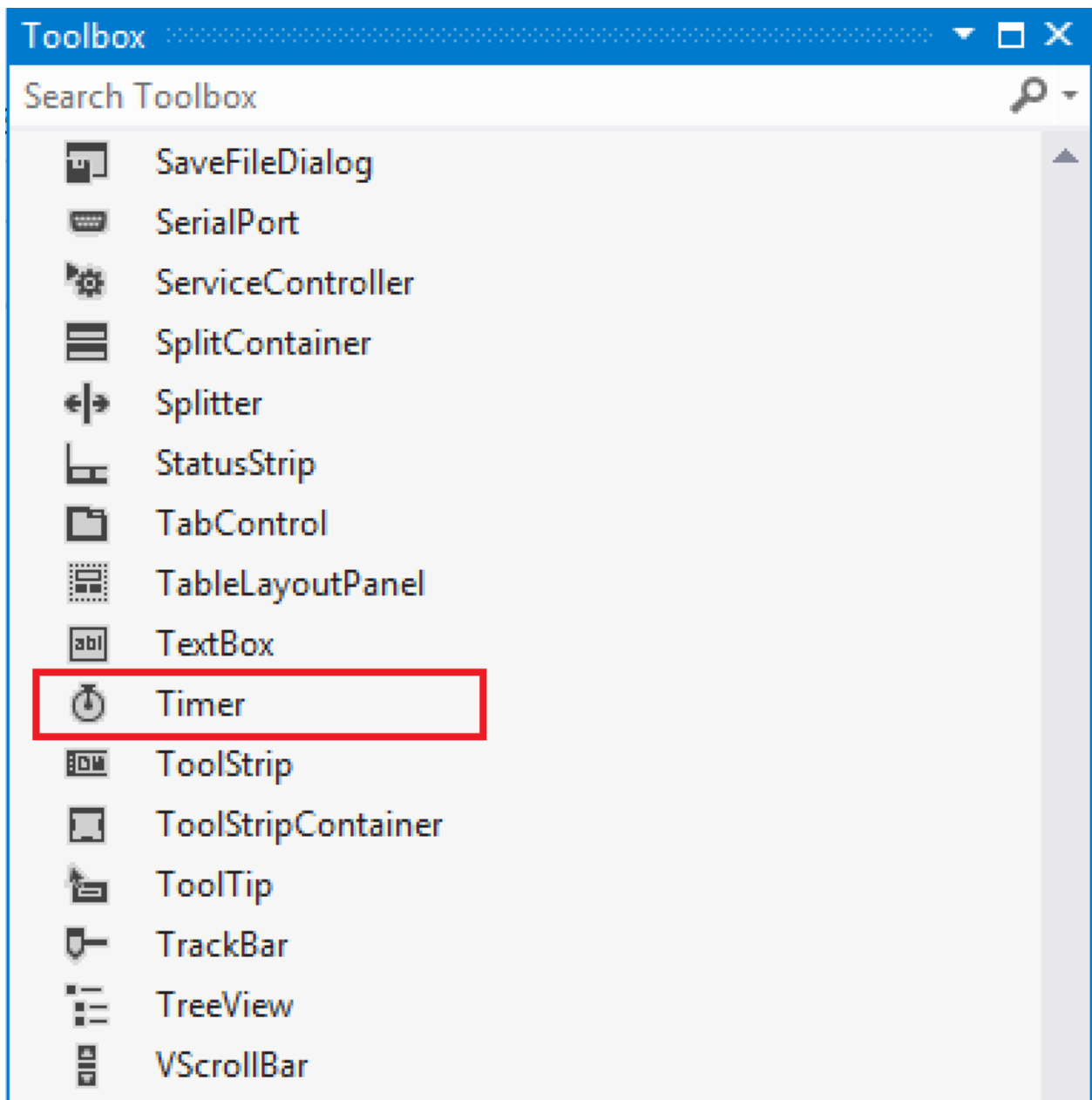
    // Đổ dữ liệu qua mảng
    arr[0] = txtHoTen.Text;
    arr[1] = txtQueQuan.Text;
    arr[2] = txtTruongHoc.Text;

    item = new ListViewItem(arr); // Đổ từ mảng vào item
    lsvDanhSachHocSinh.Items.Add(item); // Đổ từ item vào ListView
}

private void lsvDanhSachHocSinh_DoubleClick(object sender, EventArgs e)
{
    // Để SelectedItems[0] là nó tự động tăng lên khi ta Click qua dòng mới
    // SubItems là các cột trên dòng đó.
    txtHoTen.Text = lsvDanhSachHocSinh.SelectedItems[0].SubItems[0].Text;
    txtQueQuan.Text = lsvDanhSachHocSinh.SelectedItems[0].SubItems[1].Text;
    txtTruongHoc.Text = lsvDanhSachHocSinh.SelectedItems[0].SubItems[2].Text;
}

```

8. Timer



Nguyên tắc đặt tên: tm<Tên Tự Nghĩ ra>

Vd: tmThoiGian

Sử dụng để quy tắc cứ sau 1 khoảng thời gian nào đó thì sẽ lặp lại hành động nào đó

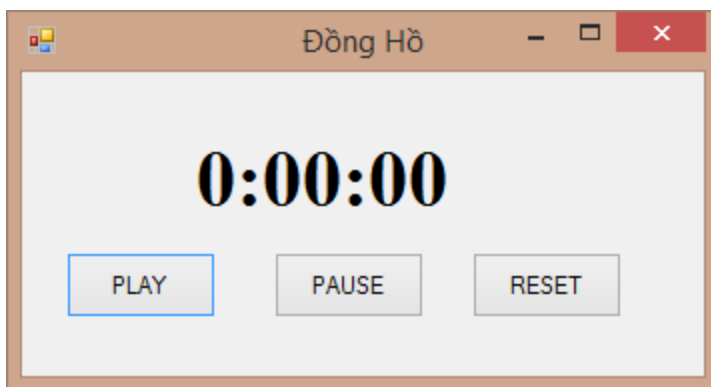
- ⇒ Sử dụng để xét tính liên tục của 1 ứng dụng (cập nhật liên tục)
- ⇒ Tạo hiệu ứng (Animation)

Tùy chỉnh Properties cho Timer:

Interval: Tính theo Miliseconds, cứ sau 1 khoảng Miliseconds thì sẽ lặp lại hành động

Ví dụ: Viết ứng dụng tạo đồng hồ điện tử (có giờ, phút, giây). Bấm nút bắt đầu là lập tức chạy thời gian, bấm Nút pause để tạm dừng lại, bấm thêm lần nữa thì tiếp tục chạy tiếp. Bấm Nút reset để cho thời gian về lại ban đầu.

Giao diện:



lblThoiGian

btnPlay, btnPause, btnReset

tmThoiGian

Viết code:

```
int Gio = 0, Phut = 0, Giay = 0; // Khai báo biến toàn cục.
private void Form1_Load(object sender, EventArgs e)
{
    lblThoiGian.Text = "0:00:00";
    tmThoiGian.Interval = 1000; // 1 second
}

private void btnPlay_Click(object sender, EventArgs e)
{
    tmThoiGian.Start(); // Khởi động Timer
}
int dem = 0;
private void btnPause_Click(object sender, EventArgs e)
{
    dem++;

    // Bấm vào 1 lần
    if (dem % 2 != 0)
    {
        tmThoiGian.Stop(); // Dừng Timer
    }

    // Bấm lần thứ 2
    else
    {
        tmThoiGian.Start(); // Khởi động Timer
    }
}
```

```
private void btnReset_Click(object sender, EventArgs e)
{
    Gio = Phut = Giay = 0; // Reset
    lblThoiGian.Text = "0:00:00"; // Reset
    tmThoiGian.Stop(); // Dừng Timer
}

private void tmThoiGian_Tick(object sender, EventArgs e)
{
    /* Quy Tắc Tính */
    Giay++;
    if (Giay == 60)
    {
        Giay = 0;
        Phut++;
    }
    if (Phut == 60)
    {
        Phut = 0;
        Gio++;
    }

    // Chuẩn hóa để hiển thị ra đủ 2 số
    string hienthi_gio = Gio > 9 ? Gio.ToString() : "0" + Gio.ToString();
    string hienthi_phut = Phut > 9 ? Phut.ToString() : "0" + Phut.ToString();
    string hienthi_giay = Giay > 9 ? Giay.ToString() : "0" + Giay.ToString();

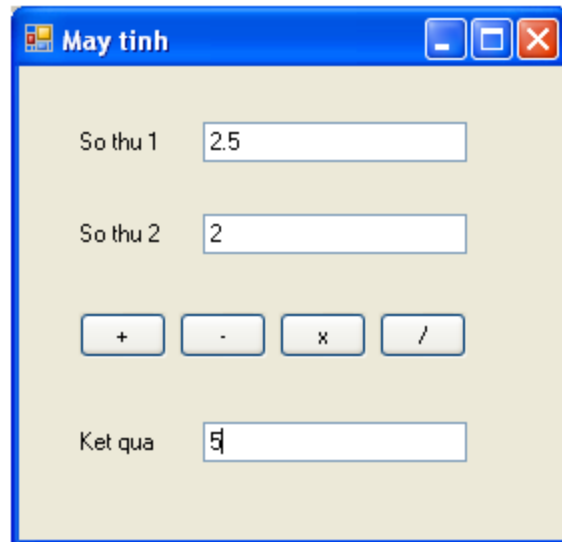
    // Hiển thị ra trên Form.
    lblThoiGian.Text = hienthi_gio + ":" + hienthi_phut + ":" + hienthi_giay;
}
}
```

HƯỚNG DẪN MỘT SỐ BÀI TẬP

Bài tập 1

Yêu cầu :

Viết chương trình Máy tính với 4 phép toán Cộng, Trừ, Nhân, Chia



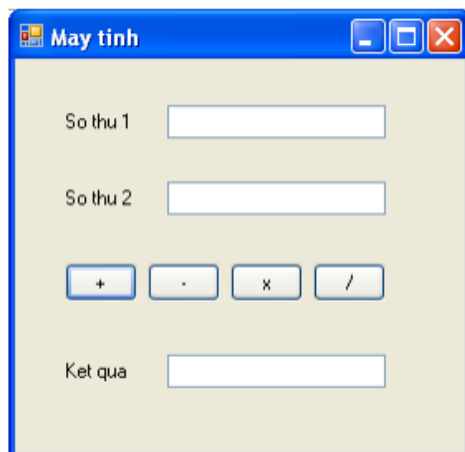
The screenshot shows a Windows-style application window titled "Máy tính" (Calculator). The window has a blue title bar with standard minimize, maximize, and close buttons. The main area has a light beige background. It contains two input fields for numbers: "Số thu 1" (Number 1) with the value "2.5" and "Số thu 2" (Number 2) with the value "2". Below these are four buttons for arithmetic operations: "+", "-", "x", and "/". At the bottom, there is an output field labeled "Kết quả" (Result) which displays "5".

Hướng dẫn :

Bước 1 : Tạo project mới

- Tạo project mới **File** → **New** → **Project...**
- Tạo ứng dụng dạng **Window Application**, ngôn ngữ **C#**

Bước 2 : Tạo các control cơ bản (TextBox, Label, Button)



Bước 3 : Đặt thuộc tính cho các control

Diễn giải	Thuộc tính	Giá trị
Textbox Số thứ 1	Name	txtSoThu1
Textbox Số thứ 2	Name	txtSoThu2
Button Cộng	Name	btnCong
	Text	+
Button Trừ	Name	btnTru
	Text	-
Button Nhân	Name	btnNhan
	Text	x
Button Chia	Name	btnChia
	Text	/

Bước 4 : Viết sự kiện Click cho btnCong bằng cách double click vào button

```
private void btnCong_Click(object sender, EventArgs e)
{
    double so1 = Convert.ToDouble(txtSoThu1.Text);
    double so2 = Convert.ToDouble(txtSoThu2.Text);
    double kq = so1 + so2;

    txtKetQua.Text = kq.ToString();
}
```

Tương tự cho các button khác

Bước 5 : bấm F5 để chạy chương trình

Bài tập 2

Yêu cầu :

Viết chương trình Giải phương trình bậc 1



Hướng dẫn :

Bước 1 : Thiết kế giao diện



Bước 2 : Đặt thuộc tính cho các control

Diễn giải	Thuộc tính	Giá trị
Textbox Tham số a	Name	txtThamSoA
Textbox Tham số b	Name	txtThamSoB
Textbox Nghiệm	Name	txtNghiệm

Button Tính nghiệm	Name	btnTinhNghiem
	Text	Tinh nghiệm

Bước 3 : Viết sự kiện Click cho btnTinhNghiem

```
private void btnTinhNghiem_Click(object sender, EventArgs e)
{
    double a = Convert.ToDouble(txtThamSoA.Text);
    double b = Convert.ToDouble(txtThamSoB.Text);

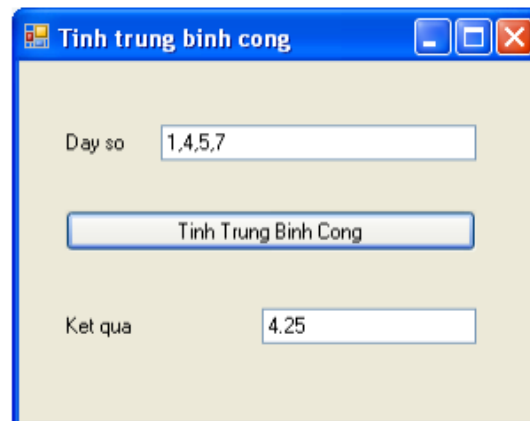
    if (a == 0)
    {
        if (b == 0) txtNghiem.Text = "Vo so nghiem";
        else txtNghiem.Text = "Vo nghiem";
    }
    else
    {
        double x = -b / a;
        txtNghiem.Text = "x=" + x.ToString();
    }
}
```

Bước 4 : bấm F5 để chạy chương trình

Bài tập 3

Yêu cầu :

Viết chương trình Tính trung bình cộng của 1 dãy số phân cách nhau bởi dấu phẩy “,”



Hướng dẫn :

Bước 1 : Thiết kế giao diện và đặt thuộc tính cho các control

Diễn giải	Thuộc tính	Giá trị
Textbox Dãy số	Name	txtDaySo
Textbox Kết quả	Name	txtKếtQuả
Button Tính TBC	Name	btnTínhTBC
	Text	Tính Trung Bình Cong

Bước 2 : Viết sự kiện Click cho btnTínhTBC

```
private void btnTínhTBC_Click(object sender, EventArgs e)
{
    string strDaySo = txtDaySo.Text;
    string[] arrDaySo = strDaySo.Split(',');
    double tong = 0;
    int n = arrDaySo.Length;
    for (int i = 0; i < n; i++)
    {
        double so = Convert.ToDouble(arrDaySo[i]);
        tong = tong + so;
    }
    double tbc = tong / n;

    txtKetQua.Text = tbc.ToString();
}
```

3 Bài tập về nhà

3.1 Bài tập 1 :

Nội dung :

Viết chương trình Máy tính với 4 phép toán Cộng, Trừ, Nhân, Chia cho 2 **Phân số**

3.2 Bài tập 2 :

Nội dung :

Viết chương trình Giải phương trình bậc 2

3.3 Bài tập 3 :

Nội dung :

Viết chương trình Tính trung bình cộng của 1 dãy **Phân số** (có dạng **a/b**) phân cách nhau bởi dấu phẩy “,”

Ví dụ : nhập dãy “1/2,3/4,5/8” có giá trị trung bình là “5/8”

2.1 Bài tập 1

Yêu cầu :

Viết chương trình nhập vào một mảng số nguyên. Tính trung bình cộng các số nguyên dương. Dùng listbox để biểu diễn mảng số nguyên. Phải có các button cho phép thêm, xóa, sửa các phần tử của mảng.xóa, sửa các phần tử của mảng.

The screenshot shows a Windows application window titled "Form1". The main title of the application is "Tính trung bình cộng" (Calculate the average). Below this, there is a section titled "Danh sách các số nguyên" (List of integers) which contains a listbox with the numbers 1, 2, 3, and 4. Below the listbox is a section titled "Chi tiết phần tử của danh sách" (Details of the element in the list). This section includes a label "Giá trị" (Value) followed by a text box containing the number 4. Below the text box are three buttons: "Thêm" (Add), "Xóa" (Delete), and "Sửa" (Edit). Below these buttons is a "Tính" (Calculate) button. At the bottom of the form, there is a label "Kết quả" (Result) followed by a text box containing the value 2.5.

Hướng dẫn :

Bước 1 : Thiết kế giao diện và đặt thuộc tính cho các control

Diễn giải	Thuộc tính	Giá trị
ListBox danh sách các số Nguyên	Name	lbxDanhSachSoNguyen
Textbox giá trị	Name	txtGiaTri
Textbox kết quả	Name	txtKetQua
Button Thêm	Name	btnThem
	Text	Them
Button Xóa	Name	btnXoa
	Text	Xoa
Button Sửa	Name	btnSua
	Text	Sua
Button Tính	Name	btnTinh
	Text	Tinh

Bước 2 : Viết code cho sự kiện Click của btnThem

```
private void btnThem_Click(object sender, EventArgs e)
{
    try
    {
        int number = Convert.ToInt32(txtGiaTri.Text);
        lbxDanhSachSoNguyen.Items.Add(number);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Nhập số không hợp lệ");
    }
}
```

Bước 3 : Viết code cho sự kiện Click của btnXoa

```
private void btnXoa_Click(object sender, EventArgs e)
{
    int index = lbxDanhSachSoNguyen.SelectedIndex;
    if (index >= 0)
    {
        lbxDanhSachSoNguyen.Items.RemoveAt(index);
    }
    else
    {
        MessageBox.Show("Chưa chọn phần tử cần xóa");
    }
}
```

```
}
```

Bước 4 : Viết code cho sự kiện Click của btnSua

```
private void btnSua_Click(object sender, EventArgs e)
{
    int index = lbxDanhSachSoNguyen.SelectedIndex;
    if (index >= 0)
    {
        try
        {
            int number = Convert.ToInt32(txtGiaTri.Text);
            lbxDanhSachSoNguyen.Items[index] = number;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Nhập số không hợp lệ");
        }
    }
    else
    {
        MessageBox.Show("Chưa chọn phần tử cần sửa");
    }
}
```

Bước 5 : Viết code cho sự kiện Click của btnTinh

```
private void btnTinh_Click(object sender, EventArgs e)
{
    int tong = 0;
    int n = lbxDanhSachSoNguyen.Items.Count;

    for (int i = 0; i < n; i++)
    {
        int number = Convert.ToInt32(lbxDanhSachSoNguyen.Items[i]);
        tong += number;
    }

    double trungBinhCong = (double)tong / n;
    txtKetQua.Text = trungBinhCong.ToString();
}
```

Bước 6 : Viết code cho sự kiện SelectedIndexChanged của lbxDanhSachSoNguyen

```
private void lbxDanhSachSoNguyen_SelectedIndexChanged(object sender, EventArgs e)
{
    int index = lbxDanhSachSoNguyen.SelectedIndex;
    if (index >= 0)
    {
        txtGiaTri.Text = lbxDanhSachSoNguyen.Items[index].ToString();
    }
}
```


3 Bài tập về nhà

3.1 Bài tập 1 :

Nội dung :

Viết chương trình nhập vào một mảng số nguyên. Tính trung bình cộng các số nguyên dương. Dùng **ComboBox** để biểu diễn mảng số nguyên. Phải có các button cho phép thêm, xóa, sửa các phần tử của mảng.xóa, sửa các phần tử của mảng.

Muốn dữ liệu của TextBox khi nhập vào hiển thị ra dạng mật khẩu ta có 2 cách như sau:

Cách 1: Trong Properties ta tìm mục UseSystemPasswordChar và bật nó lên là true

⇒ Hiển thị dấu chấm tròn đen

Cách 2: Trong Properties ta tìm mục PasswordChar và ta muốn nó hiển thị ký tự là gì thì ta điền ký tự mà ta muốn vào đó (Lưu ý chỉ được có 1 ký tự)

```
private void Form1_Load(object sender, EventArgs e)
{
    // Cách 1
    txtMatKhau.UseSystemPasswordChar = true;

    // Cách 2
    txtMatKhau.PasswordChar = '*';
}
```

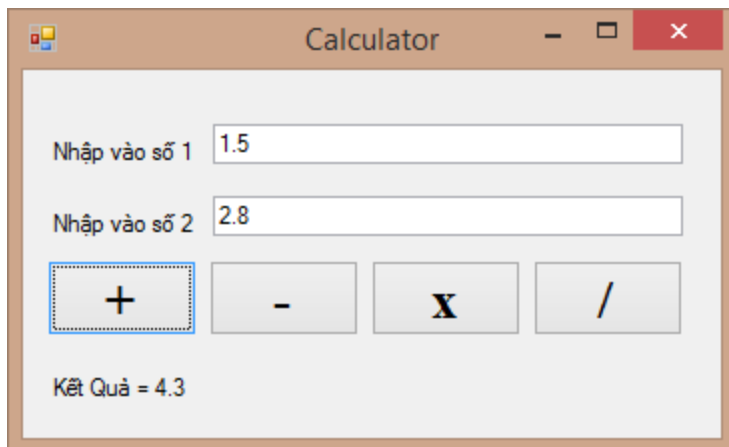
Chạy thử Form lên và trải nghiệm ☺

Cơ chế try catch trong C#.NET

Tác dụng: Để bắt ra được lỗi xảy ra trong quá trình biên dịch (ví dụ sai kiểu dữ liệu)

Ví dụ làm ứng dụng máy tính bỏ túi, thực hiện nhập vào 2 số và đi tính tổng/hiệu/tích/thương. Cần lưu ý dữ liệu nhập vào phải là số (số nguyên/số thực) thì chương trình mới đi tính toán, ngược lại sẽ báo lỗi.

Giao diện:



txtSo1, txtSo2

btnCong, btnTru, btnNhan, btnChia

lblKetQua

Viết code cho nút Button cộng như sau:

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

double So1, So2, KetQua;
bool Check;
private void btnCong_Click(object sender, EventArgs e)
{
    Check = true;
    try{
        So1 = double.Parse(txtSo1.Text);
    }catch{
        MessageBox.Show("Kiểu dữ liệu của Số 1 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo1.Clear(); // Xóa hết TextBox Số 1
        txtSo1.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 1
        Check = false;
    }

    try{
        So2 = double.Parse(txtSo2.Text);
    }catch{
        MessageBox.Show("Kiểu dữ liệu của Số 2 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo2.Clear(); // Xóa hết TextBox Số 2
        txtSo2.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 2
        Check = false;
    }

    if (Check == true){
        KetQua = So1 + So2;
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
    }
}
```

Tương tự như vậy, ta chỉ cần copy đoạn code vừa làm bên Button cộng, qua bên các Button còn lại sửa lại dấu tương ứng là xong

```
private void btnTru_Click(object sender, EventArgs e)
{
    Check = true;
    try
    {
        So1 = double.Parse(txtSo1.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 1 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo1.Clear(); // Xóa hết TextBox Số 1
        txtSo1.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 1
        Check = false;
    }

    try
    {
        So2 = double.Parse(txtSo2.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 2 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo2.Clear(); // Xóa hết TextBox Số 2
        txtSo2.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 2
        Check = false;
    }

    if (Check == true)
    {
        KetQua = So1 - So2;
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
    }
}
```

```
private void btnNhan_Click(object sender, EventArgs e)
{
    Check = true;
    try
    {
        So1 = double.Parse(txtSo1.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 1 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo1.Clear(); // Xóa hết TextBox Số 1
        txtSo1.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 1
        Check = false;
    }

    try
    {
        So2 = double.Parse(txtSo2.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 2 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo2.Clear(); // Xóa hết TextBox Số 2
        txtSo2.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 2
        Check = false;
    }

    if (Check == true)
    {
        KetQua = So1 * So2;
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
    }
}
```

```

private void btnChia_Click(object sender, EventArgs e)
{
    Check = true;
    try
    {
        So1 = double.Parse(txtSo1.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 1 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo1.Clear(); // Xóa hết TextBox Số 1
        txtSo1.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 1
        Check = false;
    }

    try
    {
        So2 = double.Parse(txtSo2.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 2 không hợp lệ. Xin kiểm tra lại !", "Lỗi");
        txtSo2.Clear(); // Xóa hết TextBox Số 2
        txtSo2.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 2
        Check = false;
    }

    if (Check == true)
    {
        KetQua = So1 / So2;
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
    }
}

```

Tuy nhiên ta thấy rằng ở 4 nút này nó đều giống nhau ở việc xét try catch cho 2 TextBox cho nên ta có thể gom chúng lại tạo thành 1 hàm và chỉ việc gọi tên hàm vào lại trong sự kiện Click của từng Button tương ứng

Có 2 cách để gom chúng thành hàm

Cách 1: Tự viết hàm ra

⇒ Xác định hàm sẽ có kiểu trả về là kiểu gì ? có cần tham số hay không ? Và đi cài đặt hàm

Cách 2: Dùng cơ chế Visual C# hỗ trợ

Tô đen đoạn Code cần gom chúng thành hàm. Click phải chọn Refactor → Extract Method và tiến hành đặt tên cho nó, xong rồi bấm OK để hoàn thành.

```
bool Check;
private void btnCong_Click(object sender, EventArgs e)
{
```

```
    Check = true;
```

```
    try{
```

```
        So1 = double.Parse(txtSo1.Text);
```

```
    }catch{
```

```
        MessageBox.Show("Kiểu
```

```
        txtSo1.Clear(); // Xóa
```

```
        txtSo1.Focus(); // Ch
```

```
        Check = false;
```

```
    }
```

```
    try{
```

```
        So2 = double.Parse(tx
```

```
    }catch{
```

```
        MessageBox.Show("Kiểu
```

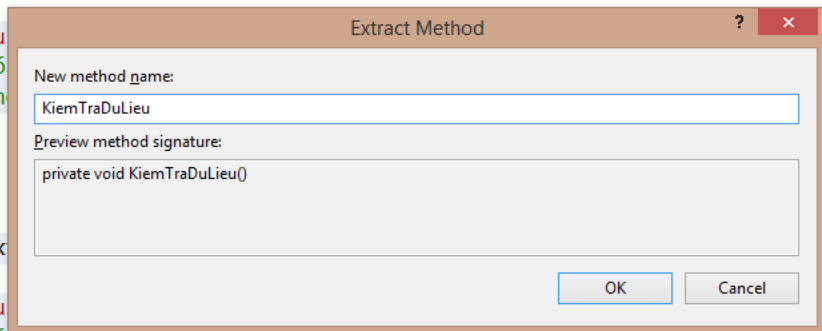
```
        txtSo2.Clear(); // Xóa hết TextBox Số 2
```

```
        txtSo2.Focus(); // Cho con trỏ dừng ngay tại TextBox Số 2
```

```
        Check = false;
```

```
    }
```

```
    if (Check == true){
```



Kết quả sau khi nhấn OK

```
double So1, So2, KetQua;
```

```
bool Check;
```

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

```
{
```

```
    KiemTraDuLieu();
```

```
    if (Check == true){
```

```
        KetQua = So1 + So2;
```

```
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
```

```
    }
```

```
}
```

```

private void KiemTraDuLieu()
{
    Check = true;
    try
    {
        So1 = double.Parse(txtSo1.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 1 không hợp lệ. Xin kiểm tra lại!", "Lỗi");
        txtSo1.Clear(); // Xóa hết TextBox Số 1
        txtSo1.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 1
        Check = false;
    }

    try
    {
        So2 = double.Parse(txtSo2.Text);
    }
    catch
    {
        MessageBox.Show("Kiểu dữ liệu của Số 2 không hợp lệ. Xin kiểm tra lại!", "Lỗi");
        txtSo2.Clear(); // Xóa hết TextBox Số 2
        txtSo2.Focus(); // Cho con trỏ đứng ngay tại TextBox Số 2
        Check = false;
    }
}

```

Tương tự ta viết code cho các nút Button còn lại


```

private void btnTru_Click(object sender, EventArgs e)
{
    KiemTraDuLieu();

    if (Check == true)
    {
        KetQua = So1 - So2;
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
    }
}

private void btnNhan_Click(object sender, EventArgs e)
{
    KiemTraDuLieu();

    if (Check == true)
    {
        KetQua = So1 * So2;
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
    }
}

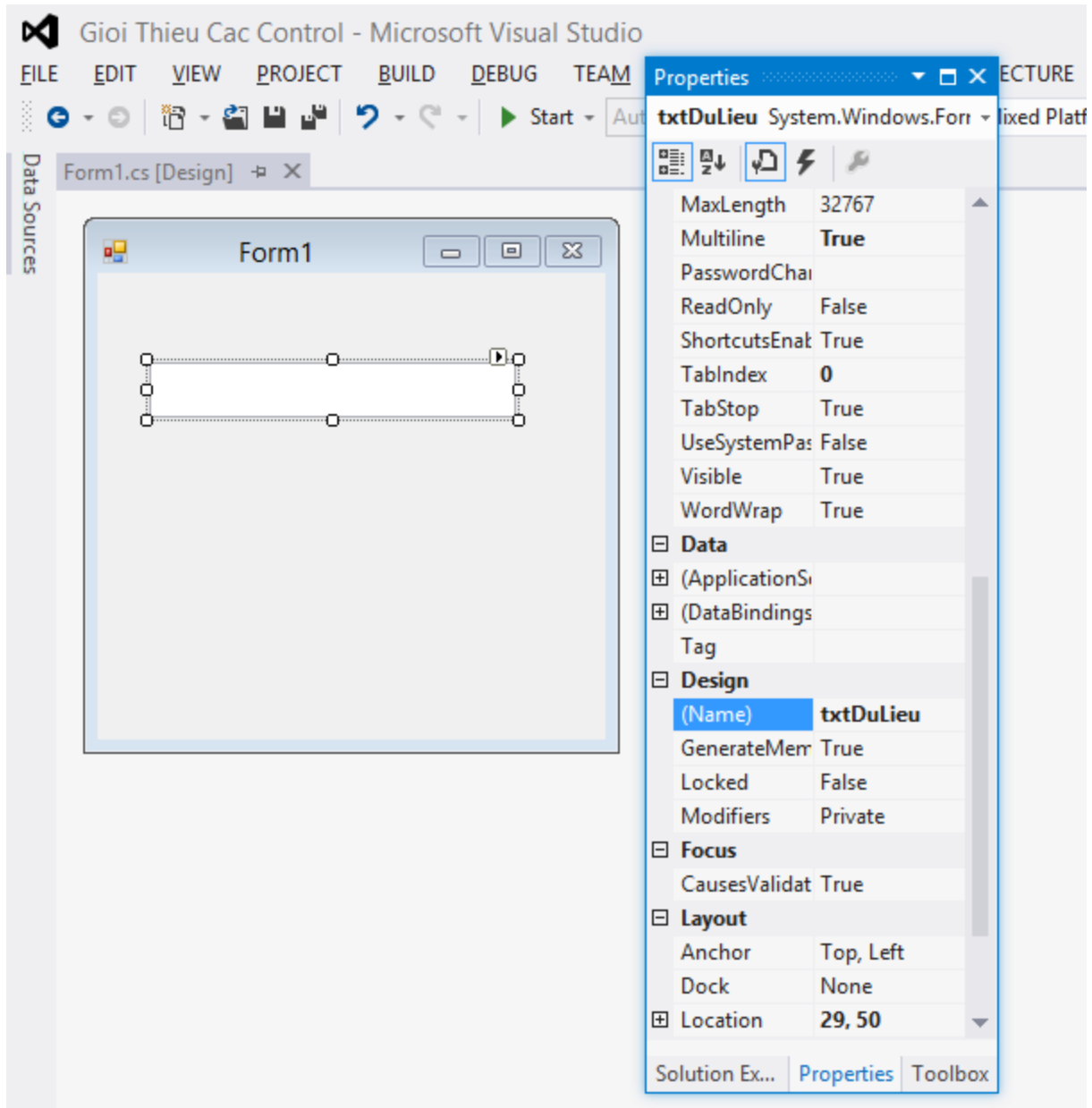
private void btnChia_Click(object sender, EventArgs e)
{
    KiemTraDuLieu();

    if (Check == true)
    {
        KetQua = So1 / So2;
        lblKetQua.Text = "Kết Quả = " + KetQua.ToString();
    }
}

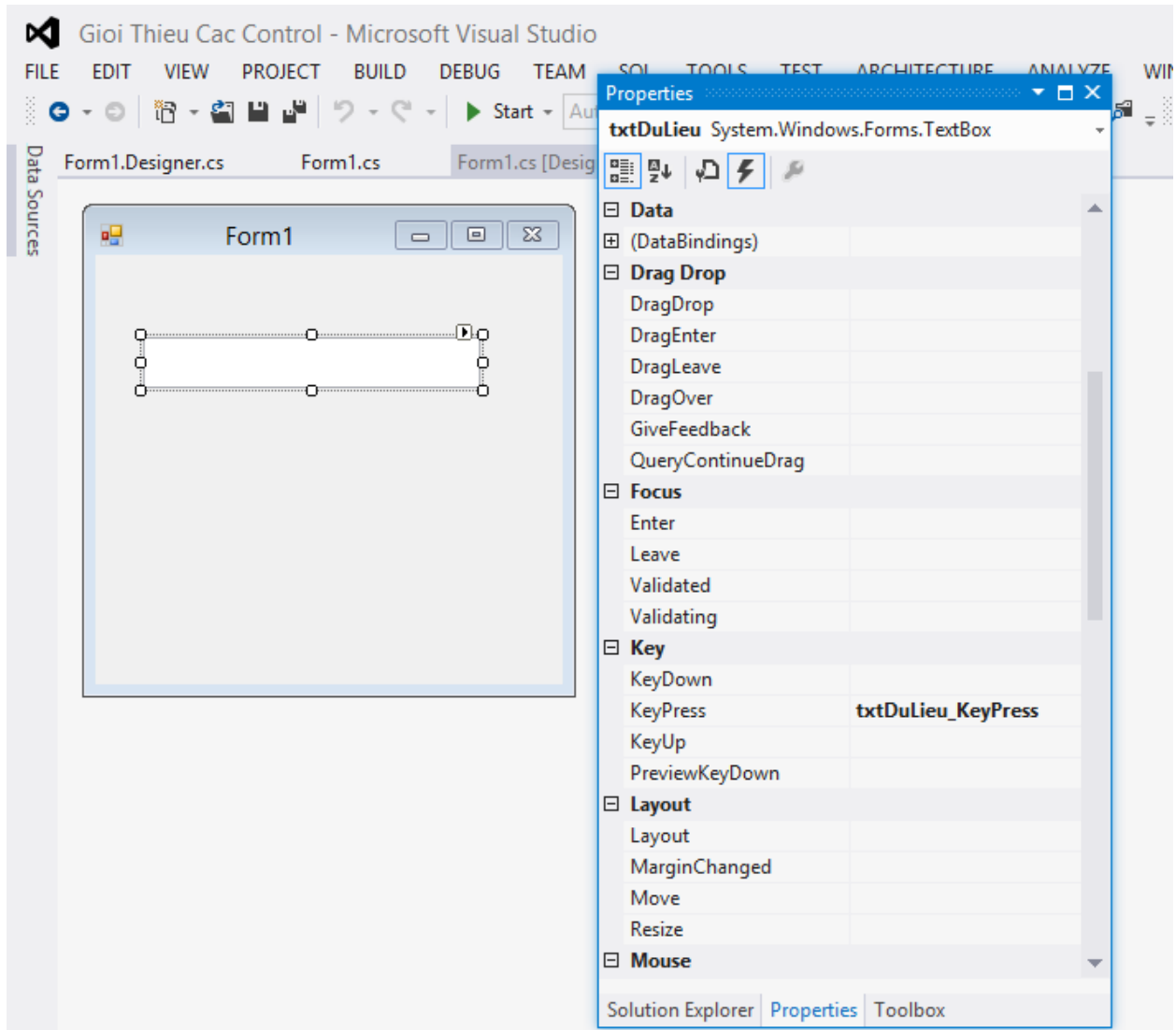
```

Tuy nhiên ngoài việc dùng try catch để đảm bảo nhập dữ liệu vào ô TextBox phải là số thì chúng ta có thể quy tắc ngay khi người dùng nhập dữ liệu vào trên ô TextBox, nếu nhập vào chữ cái thì sẽ không hiển thị ra.

Tạo 1 Project mới đặt tên là “Xử Lý Nhập Liệu Trên TextBox”



Và ở đây mình muốn là ngoài các số từ 0 tới 9 ra nó không được phép nhập thêm bất kỳ cái gì khác, chúng ta sẽ cài đặt sự kiện KeyPress như sau:



Viết code cài đặt như sau:

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

    // Bắt sự kiện tương tác phím. Nếu những giá trị người dùng nhập vào nằm trong những khoảng này thì sẽ không cho phép nhập.
    private void txtDuLieu_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (e.KeyChar < '0' || e.KeyChar > '9')
        {
            e.Handled = true; // Không cho phép nhập.
        }
    }
}
```

Tuy nhiên để ý thấy ta không thể nào xóa đi dữ liệu (Vì nút BackSpace nó nằm ngoài đoạn 0 → 9 cho nên chương trình không cho ta tương tác).

Vậy nếu như dữ liệu ta nhập sai, ta muốn dùng nút BackSpace để xóa đi thì phải làm sao ???

Khắc phục: Cài đặt chế độ KeyDown

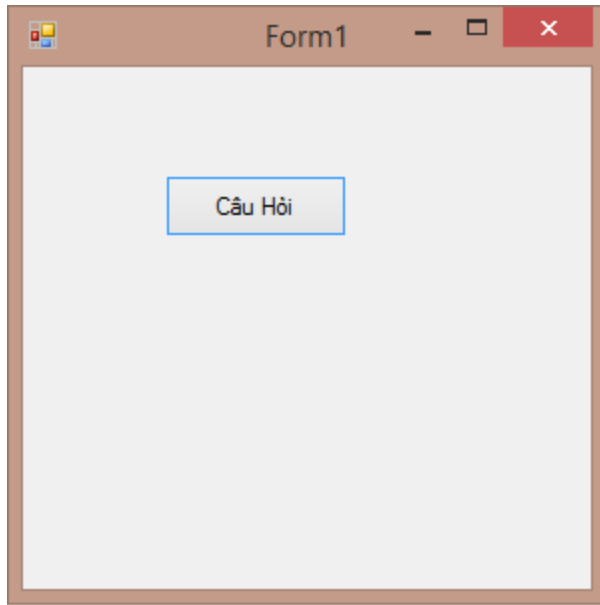
```
/* Bắt sự kiện tương tác phím. Nếu người dùng chọn phím Backspace => xóa đi ký tự cuối cùng và vị trí con trỏ nhấp nháy vẫn luôn được cài đặt ở cuối. */
private void txtDuLieu_KeyDown(object sender, KeyEventArgs e)
{
    // Tương tác khi nhấn vào nút BackSpace
    if (e.KeyCode == Keys.Back)
    {
        // Chỉ xử lý khi độ dài chuỗi khác 0.
        if (txtDuLieu.Text.Length != 0)
        {
            /* Xóa ký tự ở vị trí cuối cùng, nhưng sau khi xóa thì mặc định con trỏ nhấp nháy sẽ tự động nhảy về đầu hàng */
            txtDuLieu.Text = txtDuLieu.Text.Remove(txtDuLieu.Text.Length - 1);

            // Ta phải lùi lại vị trí con trỏ đứng ở cuối chuỗi.
            txtDuLieu.Select(txtDuLieu.Text.Length, 1);
        }
    }
}
```

Xử lý kết quả lựa chọn của hộp thoại MessageBox

Tạo 1 Project mới đặt tên là: Xu Ly Tuong Tac MessageBox

Tạo giao diện như sau:



btnCauHoi

Viết code cho nút Button như sau:

```
private void btnCauHoi_Click(object sender, EventArgs e)
{
    DialogResult TraLoi = MessageBox.Show("Nguyễn Việt Nam Sơn có đẹp trai hay không ?", "Có Thể Bạn Thừa Biết",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question);

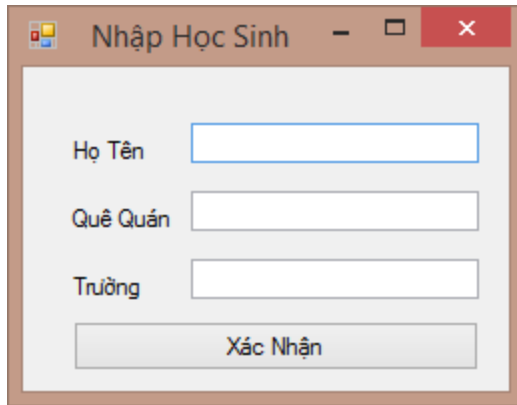
    if (TraLoi == DialogResult.Yes)
    {
        MessageBox.Show("Bạn đã lựa chọn đúng, xin chúc mừng bạn");
    }
    else
    {
        MessageBox.Show("Hừm >.< Bạn đã trả lời sai rồi đấy");
    }
}
```

Từ 1 Form ta mở ra thêm 1 Form mới, truyền dữ liệu qua lại giữa 2 Form

Tạo Project đặt tên là: Truyền Dữ Liệu Giữa 2 Form

Mặc định Form đầu tiên được tạo sẽ có tên là Form1

Tạo giao diện như sau



Nhập Học Sinh

Họ Tên

Quê Quán

Trường

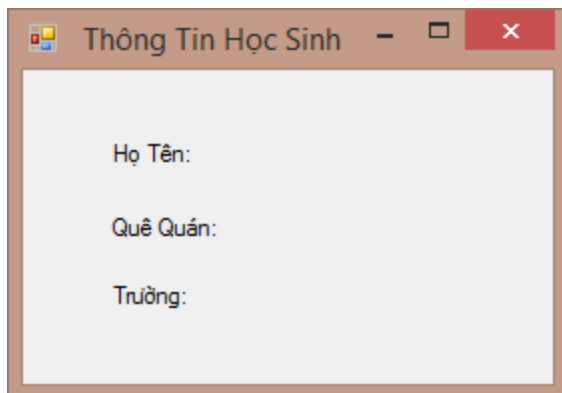
Xác Nhận

txtHoTen, txtQueQuan, txtTruong

btnXacNhan

Tiếp tục tạo ra thêm 1 Form thứ 2 bằng cách Click phải chuột lên Project hiện tại chọn Add → Windows Form

Đặt tên Là Form2 và tạo giao diện như sau:



Thông Tin Học Sinh

Họ Tên:

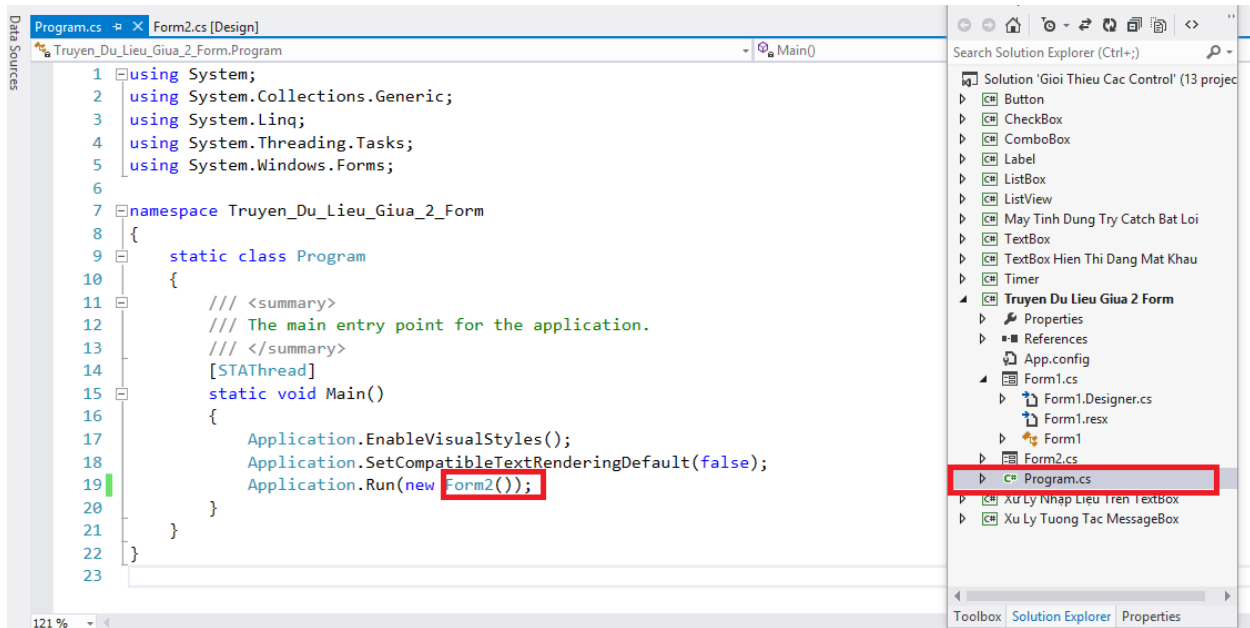
Quê Quán:

Trường:

lblHoTen, lblQueQuan, lblTruong

Mặc định khi Form2 được tạo xong, ta thử bấm F5 thì nó vẫn chỉ chạy Form1, muốn chạy thử nghiệm lên Form2 ta phải làm như sau:

Trong Project tương ứng mở file Program.cs (1 Project sẽ luôn có 1 file này, file này sẽ quyết định Form nào được chạy lần đầu tiên)



Để Test vậy thôi nhưng ta vẫn để mặc định là Form1 sẽ được chạy đầu tiên, khi ta bấm nút OK thì nó sẽ mở Form2 lên và truyền dữ liệu từ Form1 qua Form2, Form2 sẽ hiển thị thông tin học sinh lên

Ta vào Form2 tạo ra 3 biến toàn cục để tầm vực là public và cài đặt sự kiện FormLoad như sau:

```

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

    public string hoten, quequan, truong;

    private void Form2_Load(object sender, EventArgs e)
    {
        lblHoTen.Text = "Họ tên: " + hoten;
        lblQueQuan.Text = "Quê Quán: " + quequan;
        lblTruong.Text = "Trường: " + truong;
    }
}

```

Sau đó ta về Form1 và đi viết code cho nút Button Xác Nhận:

```

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

    private void btnXacNhan_Click(object sender, EventArgs e)
    {
        Form2 frm2 = new Form2();
        frm2.hoten = txtHoTen.Text;
        frm2.quequan = txtQueQuan.Text;
        frm2.truong = txtTruong.Text;
        frm2.ShowDialog();
        // Ngoài ra còn có chế độ thứ 2 là: frm2.Show();
        // Hãy thử 2 chế độ và tự hiểu nó khác nhau như thế nào nhé các bạn :)
    }
}

```


ÔN TẬP TỔNG QUÁT LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRÊN CÁC NGÔN NGỮ KHÁC NHAU

Bài toán 1: Viết chương trình nhập vào thông tin của học sinh gồm: Họ tên + Giới Tính (Nam/Nữ/Gay/Less) + Điểm toán + Điểm lý + Điểm hóa. Tính Điểm Trung Bình và xếp loại học sinh, biết công thức xếp loại như sau:

$9 \leq \text{Đtb} \leq 10$: Xuất Sắc

$8 \leq \text{Đtb} < 9$: Giỏi

$7 \leq \text{Đtb} < 8$: Khá

$6 \leq \text{Đtb} < 7$: Trung Bình Khá

$5 \leq \text{Đtb} < 6$: Trung Bình

$2 \leq \text{Đtb} < 5$: Yếu

$\text{Đtb} < 2$: Kém

Công thức tính Điểm Trung Bình = (Điểm toán + Điểm lý + Điểm hóa) / 3

Làm Bằng C++ không chia ra file (Viết hết code trong 1 file cpp)

```
#include <iostream>
#include <string>
using namespace std;

class HocSinh
{
private:
    string HoTen;
    int GioiTinh;
    double DiemToan, DiemLy, DiemHoa;
public:
    void NhapHocSinh()
    {
        fflush(stdin); // Xóa bộ nhớ đệm.
        cout << "\nNhap vao ho ten: ";
        getline(cin, HoTen);

        cout << "\n----- Nhap Vao Gioi Tinh -----";
        cout << "\n1. Nam";
        cout << "\n2. Nu";
        cout << "\n3. Gay";
        cout << "\n4. Less";

        do{
            cout << "\nNhap vao gioi tinh: ";
```

```
    cin >> GioiTinh;

    if(GioiTinh < 1 || GioiTinh > 4)
    {
        cout << "\nGioi tinh khong hop le. Xin kiem tra lai !";
    }
}while(GioiTinh < 1 || GioiTinh > 4);

do{
    cout << "\nNhap vao diem toan: ";
    cin >> DiemToan;

    if(DiemToan < 0 || DiemToan > 10)
    {
        cout << "\nDiem toan khong hop le. Xin kiem tra lai !";
    }
}while(DiemToan < 0 || DiemToan > 10);

do{
    cout << "\nNhap vao diem ly: ";
    cin >> DiemLy;

    if(DiemLy < 0 || DiemLy > 10)
    {
        cout << "\nDiem ly khong hop le. Xin kiem tra lai !";
    }
}
```

```

    }
}while(DiemLy < 0 || DiemLy > 10);

do{
    cout << "\nNhap vao diem hoa: ";
    cin >> DiemHoa;

    if(DiemHoa < 0 || DiemHoa > 10)
    {
        cout << "\nDiem hoa khong hop le. Xin kiem tra lai !";
    }
}while(DiemHoa < 0 || DiemHoa > 10);
}

void XuatHocSinh()
{
    cout << "\nHo ten: " << HoTen;
    cout << "\nGioi tinh: ";
    if(GioiTinh == 1)
    {
        cout << "Nam";
    }
    else if(GioiTinh == 2)
    {
        cout << "Nu";
    }
}

```

```

    }
    else if(GioiTinh == 3)
    {
        cout << "Gay";
    }
    else
    {
        cout << "Less";
    }
    cout << "\nDiem toan: " << DiemToan;
    cout << "\nDiem ly: " << DiemLy;
    cout << "\nDiem hoa: " << DiemHoa;
    cout << "\n=> Diem trung binh: " << TinhDiemTrungBinh();
}

double TinhDiemTrungBinh()
{
    return (DiemToan + DiemLy + DiemHoa) / 3;
}

string XepLoaiHocSinh()
{
    double Dtb = TinhDiemTrungBinh();

    if(Dtb < 2)

```

```

        {
            return "Kem";
        }
        if(Dtb < 5)
        {
            return "Yeu";
        }
        if(Dtb < 6)
        {
            return "Trung Binh";
        }
        if(Dtb < 7)
        {
            return "Trung Binh Kha";
        }
        if(Dtb < 8)
        {
            return "Kha";
        }
        if(Dtb < 9)
        {
            return "Gioi";
        }
        return "Xuat Sac";
    }
};
// <Kiểu Dữ Liệu> <Tên Lớp>::<Tên Phương Thức>
int main()
{
    HocSinh hs;
    hs.NhapHocSinh();
    hs.XuatHocSinh();
    cout << "\n\nXep loai: " << hs.XepLoaiHocSinh();

    system("pause");
    return 0;
}

```

Tuy nhiên chúng ta nên chia ra file để dễ dàng quản trị hơn (trong trường hợp chúng ta có nhiều class), lúc này ta sẽ chia ra thành 2 file (*.h và *.cpp)

***.h: File dùng để khai báo**

***.cpp: File dùng để cài đặt cho những gì đã khai báo ở *.h**

Cách Tạo: Click Phải vào Project → Add → Class → Add → Đặt tên cho Class (Class Name thì tự động *.h và *.cpp sẽ tự động sinh ra theo tên ta đặt trong Class Name) → Finish

Hoặc trong Project ta có thể tự tạo thủ công bằng cách Click phải vào Project chọn Add → New item → Header File(.h) và Source File(.cpp)

Sau khi chia File xong thì chúng ta cài đặt viết code như sau:

// File HocSinh.h

```
#pragma once
#include <iostream>
#include <string>
using namespace std;

class HocSinh
{
private:
    string HoTen;
    int GioiTinh;
    double DiemToan, DiemLy, DiemHoa;
public:
    HocSinh(void);
    ~HocSinh(void);
    void NhapHocSinh();
    void XuatHocSinh();
    double TinhDiemTrungBinh();
    string XepLoaiHocSinh();
};
```

File HocSinh.cpp


```
#include "HocSinh.h"
```

```
HocSinh::HocSinh(void)
{
}
```

```
HocSinh::~~HocSinh(void)
{
}
```

```
// <Kiểu Dữ Liệu> <Tên Lớp>::<Tên Phương Thức>
```

```
void HocSinh::NhapHocSinh()
```

```
{
    fflush(stdin); // Xóa bộ nhớ đệm.
    cout << "\nNhap vao ho ten: ";
    getline(cin, HoTen);

    cout << "\n----- Nhap Vao Gioi Tinh -----";
    cout << "\n1. Nam";
    cout << "\n2. Nu";
    cout << "\n3. Gay";
    cout << "\n4. Less";
```

```
do{
    cout << "\nNhap vao gioi tinh: ";
    cin >> GioiTinh;

    if(GioiTinh < 1 || GioiTinh > 4)
    {
        cout << "\nGioi tinh khong hop le. Xin kiem tra lai !";
    }
}while(GioiTinh < 1 || GioiTinh > 4);

do{
    cout << "\nNhap vao diem toan: ";
    cin >> DiemToan;

    if(DiemToan < 0 || DiemToan > 10)
    {
        cout << "\nDiem toan khong hop le. Xin kiem tra lai !";
    }
}while(DiemToan < 0 || DiemToan > 10);

do{
    cout << "\nNhap vao diem ly: ";
    cin >> DiemLy;

    if(DiemLy < 0 || DiemLy > 10)
```

```
        {
            cout << "\nDiem ly khong hop le. Xin kiem tra lai !";
        }
    }while(DiemLy < 0 || DiemLy > 10);

    do{
        cout << "\nNhap vao diem hoa: ";
        cin >> DiemHoa;

        if(DiemHoa < 0 || DiemHoa > 10)
        {
            cout << "\nDiem hoa khong hop le. Xin kiem tra lai !";
        }
    }while(DiemHoa < 0 || DiemHoa > 10);
}

void HocSinh::XuatHocSinh()
{
    cout << "\nHo ten: " << HoTen;
    cout << "\nGioi tinh: ";
    if(GioiTinh == 1)
    {
        cout << "Nam";
    }
    else if(GioiTinh == 2)
```

```

    {
        cout << "Nu";
    }
    else if(GioiTinh == 3)
    {
        cout << "Gay";
    }
    else
    {
        cout << "Less";
    }
    cout << "\nDiem toan: " << DiemToan;
    cout << "\nDiem ly: " << DiemLy;
    cout << "\nDiem hoa: " << DiemHoa;
    cout << "\n=> Diem trung binh: " << TinhDiemTrungBinh();
}

```

```

double HocSinh::TinhDiemTrungBinh()
{
    return (DiemToan + DiemLy + DiemHoa) / 3;
}

```

```

string HocSinh::XepLoaiHocSinh()
{
    double Dtb = TinhDiemTrungBinh();
}

```

```
        if(Dtb < 2)
        {
            return "Kem";
        }
        if(Dtb < 5)
        {
            return "Yeu";
        }
        if(Dtb < 6)
        {
            return "Trung Binh";
        }
        if(Dtb < 7)
        {
            return "Trung Binh Kha";
        }
        if(Dtb < 8)
        {
            return "Kha";
        }
        if(Dtb < 9)
        {
            return "Gioi";
        }
        return "Xuat Sac";
    }
```

Cũng ví dụ trên chúng ta làm trên C# Console

Mặc định khi ứng dụng C# Console mới tạo ra thì sẽ có File Program.cs, file này sẽ quản trị cả bài (Chứa hàm main bên trong). Chúng ta có thể cùng ở trong file này khai báo ra class HocSinh và cài đặt trực tiếp ở bên trong hoặc tốt hơn hết là chúng ta tạo riêng File HocSinh.cs và cài đặt class HocSinh ở bên trong đó (Click phải Project → Add → Class → Đặt tên → Finish)

Class HocSinh cài đặt như sau:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace QuanLyHocSinh
8 {
9     class HocSinh
10    {
11        // Khai báo thuộc tính
12        private string HoTen;
13        private int GioiTinh;
14        private double DiemToan, DiemLy, DiemHoa;
15
16        // Khai báo phương thức
17        public void NhapHocSinh()
18        {
19            Console.Write("\nNhập vào họ tên: ");
20            HoTen = Console.ReadLine();
21
22            Console.Write("\n----- Nhập Vào Giới Tính -----");
23            Console.Write("\n1. Nam");
24            Console.Write("\n2. Nữ");
25            Console.Write("\n3. Gay");
26            Console.Write("\n4. Less");
27
28            //do
29            //{
30                Console.Write("\nNhập vào giới tính: ");
31                GioiTinh = int.Parse(Console.ReadLine());
```

```
32
33 //     if (GioiTinh < 1 || GioiTinh > 4)
34 //     {
35 //         Console.WriteLine("\nGioi tinh khong hop le. Xin kiem tra lai !");
36 //     }
37 //} while (GioiTinh < 1 || GioiTinh > 4);
38
39 while (true)
40 {
41     try
42     {
43         Console.WriteLine("\nNhap vao gioi tinh: ");
44         GioiTinh = int.Parse(Console.ReadLine());
45
46         if (GioiTinh < 1 || GioiTinh > 4)
47         {
48             Console.WriteLine("\nGioi tinh khong hop le. Xin kiem tra lai !");
49         }
50         else
51         {
52             break; // Hợp lệ hết tất cả mọi thứ => thoát khỏi vòng lặp.
53         }
54     }
55     catch
56     {
57         Console.WriteLine("\nKieu du lieu khong hop le. Xin kiem tra lai !");
58     }
59 }
60
61 //do
62 //{
```

```
63 // Console.WriteLine("\nNhập vào điểm toán: ");
64 // DiemToan = double.Parse(Console.ReadLine());
65
66 // if (DiemToan < 0 || DiemToan > 10)
67 // {
68 //     Console.WriteLine("\nĐiểm toán không hợp lệ. Xin kiểm tra lại!");
69 // }
70 //} while (DiemToan < 0 || DiemToan > 10);
71
72 while (true)
73 {
74     try
75     {
76         Console.WriteLine("\nNhập vào điểm toán: ");
77         DiemToan = double.Parse(Console.ReadLine());
78
79         if (DiemToan < 0 || DiemToan > 10)
80         {
81             Console.WriteLine("\nĐiểm toán không hợp lệ. Xin kiểm tra lại!");
82         }
83         else
84         {
85             break; // Hợp lệ hết tất cả mọi thứ => thoát khỏi vòng lặp.
86         }
87     }
88     catch
89     {
90         Console.WriteLine("\nKiểu dữ liệu không hợp lệ. Xin kiểm tra lại!");
91     }
92 }
93
```



```
94 //do
95 //{
96 //    Console.WriteLine("\nNhap vao diem ly: ");
97 //    DiemLy = double.Parse(Console.ReadLine());
98
99 //    if (DiemLy < 0 || DiemLy > 10)
100 //    {
101 //        Console.WriteLine("\nDiem ly khong hop le. Xin kiem tra lai !");
102 //    }
103 //} while (DiemLy < 0 || DiemLy > 10);
104
105 //do
106 //{
107 //    Console.WriteLine("\nNhap vao diem hoa: ");
108 //    DiemHoa = double.Parse(Console.ReadLine());
109
110 //    if (DiemHoa < 0 || DiemHoa > 10)
111 //    {
112 //        Console.WriteLine("\nDiem hoa khong hop le. Xin kiem tra lai !");
113 //    }
114 //} while (DiemHoa < 0 || DiemHoa > 10);
115
116 while (true)
117 {
118     try
119     {
120         Console.WriteLine("\nNhap vao diem ly: ");
121         DiemLy = double.Parse(Console.ReadLine());
122
123         if (DiemLy < 0 || DiemLy > 10)
124         {
```

```
125         Console.WriteLine("\nDiem ly khong hop le. Xin kiem tra lai !");
126     }
127     else
128     {
129         break; // Hợp lệ hết tất cả mọi thứ => thoát khỏi vòng lặp.
130     }
131 }
132 catch
133 {
134     Console.WriteLine("\nKieu du lieu khong hop le. Xin kiem tra lai !");
135 }
136 }
137
138 while (true)
139 {
140     try
141     {
142         Console.WriteLine("\nNhap vao diem hoa: ");
143         DiemHoa = double.Parse(Console.ReadLine());
144
145         if (DiemHoa < 0 || DiemHoa > 10)
146         {
147             Console.WriteLine("\nDiem hoa khong hop le. Xin kiem tra lai !");
148         }
149         else
150         {
151             break; // Hợp lệ hết tất cả mọi thứ => thoát khỏi vòng lặp.
152         }
153     }
154     catch
155     {
```

```
156         Console.WriteLine("\nKieu du lieu khong hop le. Xin kiem tra lai");
157     }
158 }
159 }
160
161 public void XuatHocSinh()
162 {
163     //Console.WriteLine("\nHo ten: " + HoTen);
164     Console.WriteLine("\nHo ten: {0}", HoTen);
165
166     Console.WriteLine("\nGioi tinh: ");
167     if (GioiTinh == 1)
168     {
169         Console.WriteLine("Nam");
170     }
171     else if (GioiTinh == 2)
172     {
173         Console.WriteLine("Nu");
174     }
175     else if (GioiTinh == 3)
176     {
177         Console.WriteLine("Gay");
178     }
179     else
180     {
181         Console.WriteLine("Less");
182     }
183     Console.WriteLine("\nDiem toan: {0}", DiemToan);
184     Console.WriteLine("\nDiem ly: {0}", DiemLy);
185     Console.WriteLine("\nDiem hoa: {0}", DiemHoa);
186     Console.WriteLine("\n=> Diem trung binh: {0}", TinhDiemTrungBinh());
187 }
```

```

187     }
188
189     public double TinhDiemTrungBinh()
190     {
191         return (DiemToan + DiemLy + DiemHoa)
192     }
193
194     public string XepLoaiHocSinh()
195     {
196         double Dtb = TinhDiemTrungBinh();
197
198         if(Dtb < 2)
199         {
200             return "Kem";
201         }
202         if(Dtb < 5)
203         {
204             return "Yeu";
205         }
206         if(Dtb < 6)
207         {
208             return "Trung Binh";
209         }
210         if(Dtb < 7)
211         {
212             return "Trung Binh Kha";
213         }
214         if(Dtb < 8)
215         {
216             return "Kha";
217         }
218         if(Dtb < 9)
219         {
220             return "Gioi";
221         }
222         return "Xuat Sac";
223     }
224
225 }
226
227 class Program
228 {
229     static void Main(string[] args)
230     {
231         HocSinh hs = new HocSinh();
232         hs.NhapHocSinh();
233         hs.XuatHocSinh();
234         Console.WriteLine("\n\nXep loai: {0}", hs.XepLoaiHocSinh());
235
236         Console.ReadKey();
237     }
238 }
239 }

```

Cũng bài toán trên, ta đi xây dựng ứng dụng Windows Form có giao diện như sau:

Sơ lược các Control sử dụng trong Form

txtHoTen, txtDiemToan, txtDiemLy, txtDiemHoa, txtDiemTrungBinh

rdbNam, rdbNu, rdbGay, rdbLess

lblXepLoai

btnTinh

Bình thường ứng dụng Windows Form ta có thể đi xử lý trực tiếp bên trong nó (Viết code cho các sự kiện) nhưng nếu 1 ứng dụng quy mô lớn phải xử lý nhiều thứ ở bên trong thì bước xử lý ta sẽ giao cho các Class làm, nhiệm vụ của Form chỉ là truyền dữ liệu về cho Class và nhận kết quả từ Class xuất ra.

Giống như bài toán trên, ta sẽ cài đặt Class HocSinh để nhận dữ liệu truyền về từ Form (Thông qua phương thức khởi tạo nhận tham số đầu vào) và cài đặt các phương thức xử lý.

Click phải vào Project → Add → Class → Đặt tên → Finish

File: HocSinh.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace QuanLyHocSinh
8  {
9      class HocSinh
10     {
11         // Khai báo thuộc tính
12         private string HoTen;
13         private int GioiTinh;
14         private double DiemToan, DiemLy, DiemHoa;
15
16         public HocSinh()
17         {
18             HoTen = "Son dep trai";
19             GioiTinh = 1;
20             DiemToan = DiemLy = DiemHoa = 10;
21         }
22
23         public HocSinh(string ht, int gt, double toan, double ly, double hoa)
24         {
```

```
25         HoTen = ht;
26         GioiTinh = gt;
27         DiemToan = toan;
28         DiemLy = ly;
29         DiemHoa = hoa;
30     }
31
32     public double TinhDiemTrungBinh()
33     {
34         return (DiemToan + DiemLy + DiemHoa) / 3;
35     }
36
37     public string XepLoaiHocSinh()
38     {
39         double Dtb = TinhDiemTrungBinh();
40
41         if (Dtb < 2)
42         {
43             return "Kem";
44         }
45         if (Dtb < 5)
46         {
47             return "Yeu";
48         }
49     }
```



```
49         if (Dtb < 6)
50         {
51             return "Trung Binh";
52         }
53         if (Dtb < 7)
54         {
55             return "Trung Binh Kha";
56         }
57         if (Dtb < 8)
58         {
59             return "Kha";
60         }
61         if (Dtb < 9)
62         {
63             return "Gioi";
64         }
65         return "Xuat Sac";
66     }
67 }
68 }
69
```

Sau đó ta quay lại Form và đi viết code cho nút Button Tính như sau:

```
private void btnTinh_Click(object sender, EventArgs e)
{
    string hoten = txtHoTen.Text;
    int gioitinh;
    if (rdbNam.Checked)
    {
        gioitinh = 1;
    }
    else if (rdbNu.Checked)
    {
        gioitinh = 2;
    }
    else if (rdbGay.Checked)
    {
        gioitinh = 3;
    }
    else
    {
        gioitinh = 4;
    }
    double toan = double.Parse(txtDiemToan.Text);
    double ly = double.Parse(txtDiemLy.Text);
    double hoa = double.Parse(txtDiemHoa.Text);

    HocSinh hs = new HocSinh(hoten, gioitinh, toan, ly, hoa);

    txtDiemTrungBinh.Text = hs.TinhDiemTrungBinh().ToString();
    lblXepLoai.Text = "Xếp Loại: " + hs.XepLoaiHocSinh();
}
```

Chạy thử để xem kết quả ☺

Cũng bài toán trên, ta đi làm trên Java Console:

Cũng bài toán trên, ta đi làm trên Java Swing:

Bài toán 2: Viết ứng dụng Windows Form hỗ trợ quản lý nhân viên cho công ty

Giao diện phần mềm

Quản Lý Nhân Viên

Thông Tin Nhân Viên

Mã Nhân Viên

Tên Nhân Viên

Ngày Sinh

Hệ Số Lương

Giới Tính ☐ Nam ☐ Nữ

Phòng Ban

Thêm Xóa Sửa Tìm Kiếm Sắp Xếp Theo Hệ Số Lương Thoát

Mã NV	Tên Nhân Viên	Phái	Ngày Sinh	HS Lương	Phòng Ban	

Mô tả chi tiết phần mềm:

Tính năng Thêm:

- Khi người dùng Click vào nút Thêm thì chương trình sẽ kiểm tra trong vùng Thông Tin Nhân Viên nếu có dữ liệu nào chưa nhập vào thì sẽ xuất ra thông báo lỗi và yêu cầu phải nhập
- Khi tất cả dữ liệu đã được nhập xong thì chương trình sẽ đưa thông tin nhân viên đó vào trong bảng ListView
- Mã số của các nhân viên không được trùng nhau

Tính năng Xóa:

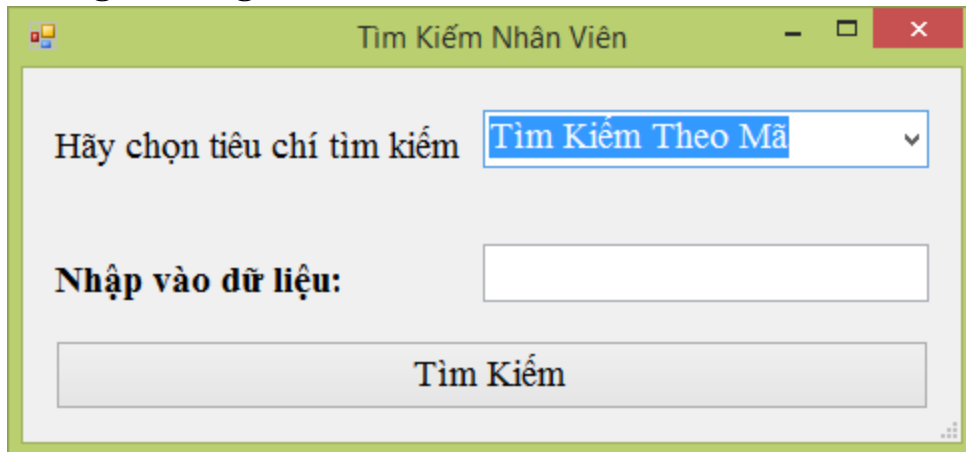
- Click chuột vào ListView và để chọn 1 dòng cụ thể cần xóa, sau đó nhấn nút để xóa dữ liệu đi, dữ liệu trong bảng ListView sẽ mất.

Tính năng Sửa:

- Khi ta chọn 1 dòng trên ListView thì dữ liệu sẽ hiển thị ra trên khung Thông Tin Nhân Viên, ta cứ tùy ý chỉnh sửa lại dữ liệu (ngoại trừ mã nhân viên thì không được phép chỉnh sửa), sau đó ta nhấn nút Sửa để dữ liệu được cập nhật lại trên ListView.

Tính năng Tìm Kiếm:

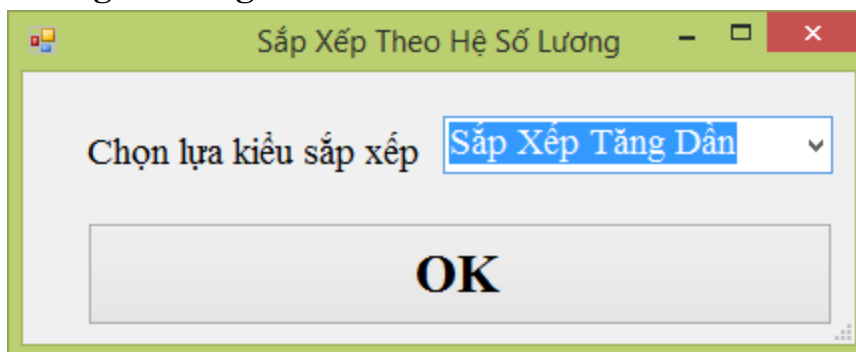
- Khi nhấn vào nút Tìm Kiếm, 1 Form khác sẽ hiển thị ra để lấy nhu cầu của người dùng như sau:



Ta sẽ lựa chọn tiêu chí tìm kiếm (Tìm kiếm theo mã hoặc theo tên), sau đó nhập dữ liệu vào ô TextBox rồi bấm nút tìm kiếm, lúc này Form sẽ đóng lại và trên ListView của Form chính sẽ có dữ liệu hiển thị ra (cả dòng tô màu đỏ)

Tính năng Sắp Xếp Theo Hệ Số Lương:

- Khi nhấn vào nút Sắp Xếp, 1 Form khác sẽ hiển thị ra để lấy nhu cầu của người dùng như sau:



Ta sẽ lựa chọn tiêu chí sắp xếp (Tăng dần hoặc giảm dần), sau đó nhấn nút OK, lúc này Form sẽ đóng và trên ListView của Form chính danh sách các nhân viên đã được sắp xếp

Tính năng Thoát:

- **Chương trình kết thúc**

Đi Tạo ra các Form, đặt tên tương ứng và đi viết code xử lý (Xem trong Project đính kèm)