



LẬP TRÌNH C# 3

BÀI 1: TỔNG QUAN WINDOW FORM

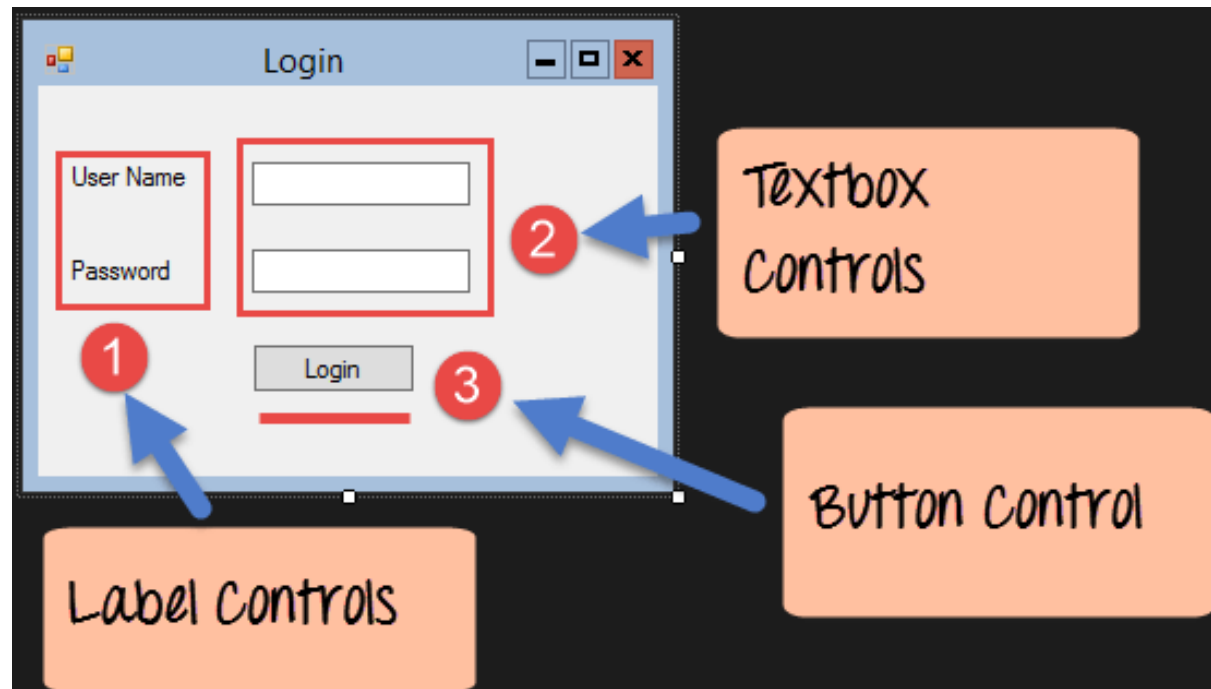
- ⊙ Thiết kế GUI - Window Form
- ⊙ Basic control



- ❑ GUI là viết tắt cho Graphical User Interface có nghĩa là giao diện người dùng đồ họa
 - ❖ Graphical: text, window, menu, button...
 - ❖ User: Người sử dụng chương trình
 - ❖ Interface: Cách tương tác chương trình
- ❑ GUI được thiết kế bằng việc thêm vào Form những điều khiển phù hợp cho yêu cầu của mỗi bài toán đặt ra
- ❑ Form chứa nhiều control khác nhau

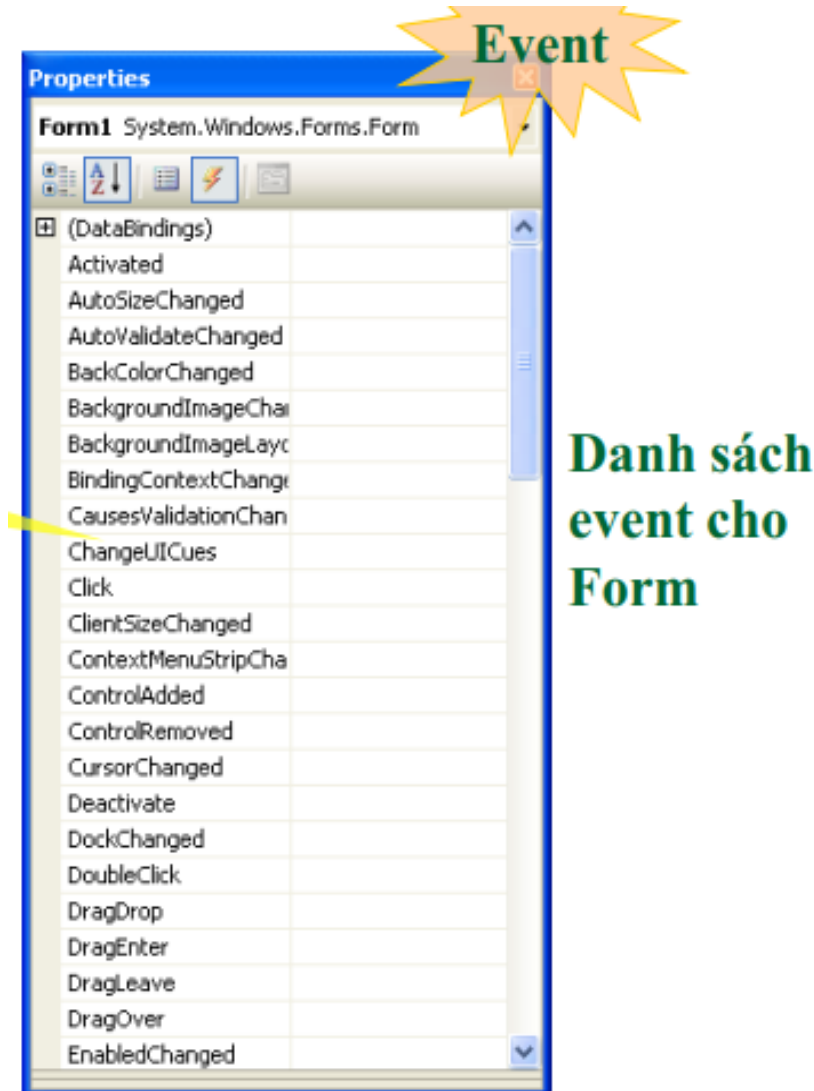
❑ Thành phần đồ họa thông dụng

- ❖ Window: Một vùng bên trong màn hình chính
- ❖ Menu: Liệt kê những chức năng
- ❖ Button: Nút lệnh cho phép click vào
- ❖ TextBox: Cho phép user nhập dữ liệu text



LẬP TRÌNH THEO SỰ KIỆN (EVENT- DRIVEN PROGRAMMING)

- ❑ Chương trình chờ cho event xuất hiện và xử lý
- ❑ Các sự kiện thông dụng:
 - ❖ Mouse move
 - ❖ Mouse click
 - ❖ Mouse double-click
 - ❖ Key press
 - ❖ Button click
 - ❖ Change in focus
 - ❖



- ❑ Form là cửa sổ chính của ứng dụng giao diện người dùng đồ họa
- ❑ Thuộc tính quan trọng

Tên thuộc tính	Ý nghĩa
BackColor/ForeColor	Màu nền / Màu chữ
BackgroundImage	Ảnh nền
AcceptButton	Nút được chọn kích hoạt sự kiện click khi người dùng click Enter
CancelButton	Nút được chọn kích hoạt sự kiện click khi người dùng click Esc
Controls	Danh sách các control con của nó. Cho phép người dùng <u>thêm hoặc xóa</u> động controls
FormBorderStyle	Chọn loại window form
Opacity	Độ mờ của window form

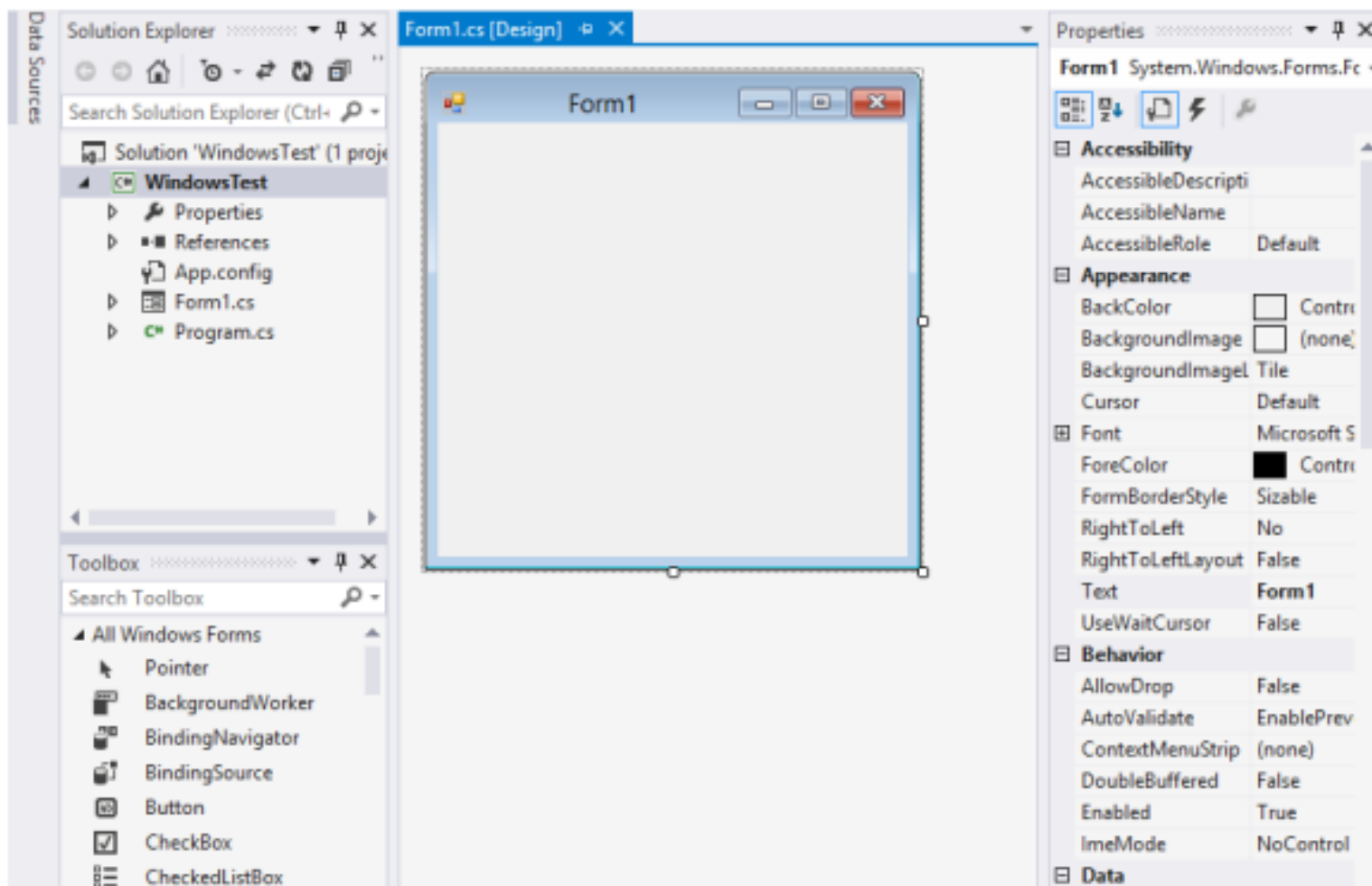
❑ Phương thức thông dụng:

Tên hàm	Ý nghĩa
Show()	Sau khi dialog hiện lên, người dùng có thể chọn focus vào control khác
ShowDialog()	Sau khi dialog hiện lên người dùng không thể focus vào các control khác. Kiểu dữ liệu trả về: DialogResult.
Close()	Đóng Dialog
this.DialogResult = DialogResult.OK	Đóng dialog và trả về kết quả tương ứng OK
This.DialogResult = DialogResult.Cancel	Đóng dialog và trả về kết quả tương ứng Cancel.

□ Các event thông dụng

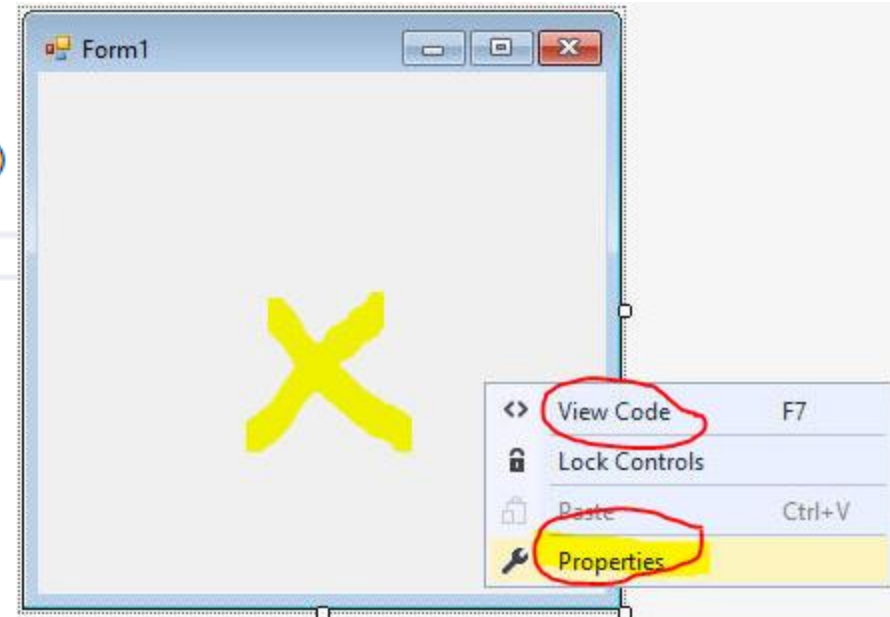
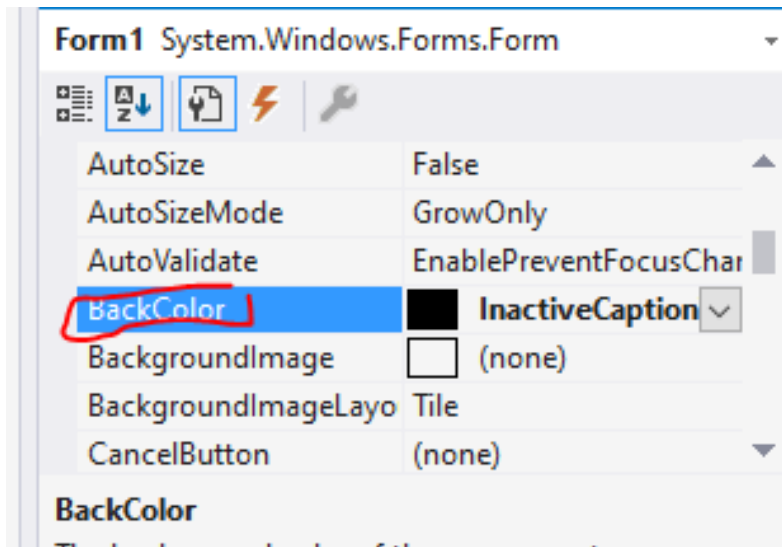
Tên hàm	Ý nghĩa
Load()	Sự kiện được kích hoạt khi Form đã load xong các control. Sự kiện này được bắt khi người dùng muốn khởi tạo biến hoặc giá trị trong Form
Paint()	Sự kiện này được gọi khi Form thực hiện vẽ lại. (Di chuyển chuột, phóng to, thu nhỏ,...) (Sẽ trình bày chi tiết trong GDI +)
KeyPress, KeyDown,..	Các sự kiện về phím
MouseHover, MouseDown, MouseLeave	Các sự kiện chuột

□ Ví dụ tạo form

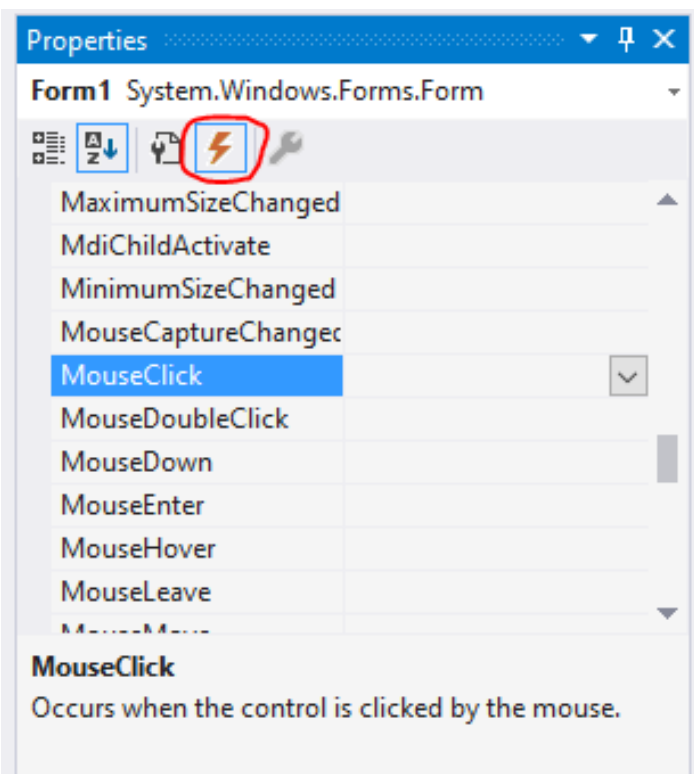


❑ Chỉnh sửa thuộc tính form từ giao diện hoặc từ code c#

```
private void Form1_Load(object sender, EventArgs e)
{
    this.BackColor = Color.Red;
}
```



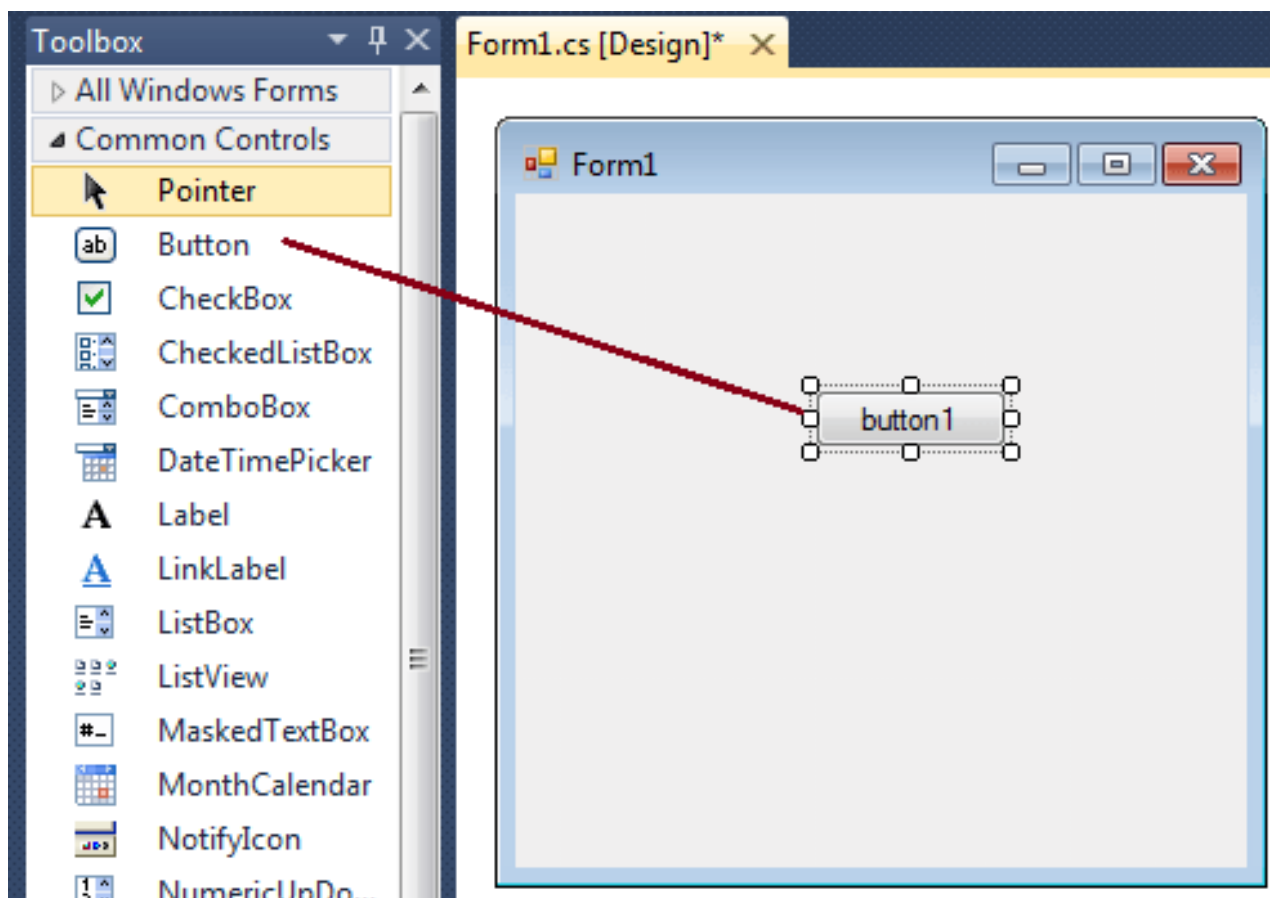
❑ Thiết lập sự kiện cho form



```
1 reference  
private void Form1_MouseClick(object sender, MouseEventArgs e)  
{  
    |  
}  
}
```

- ❑ Điều khiển (control) là các đối tượng nằm trong Form
- ❑ Có các thành phần
 - ❖ Thuộc tính (property)
 - ❖ Phương thức (method)
 - ❖ Sự kiện (event)
- ❑ Tất cả các control chứa trong namespace: `System.Windows.Forms`
- ❑ Có thể thêm control vào form từ thanh toolbox của visual studio hoặc code c#

❑ Thêm control từ toolbox



❑ Thêm control bằng cách code c#

```
using System;
using System.Windows.Forms;
using System.Drawing;
```

```
namespace
```

```
{
    public class Form1 : Form
```

Lớp Form cơ sở

Form1.cs

```
    Label title;
```

```
    public Form1()
```

```
    {
```

```
        this.Text = "Hello World!";
```

```
        this.Size = new Size(400, 200);
```

```
        title = new Label();
```

```
        title.Text = "Hello World";
```

```
        title.Location = new Point(50, 50);
```

```
        this.Controls.Add(title);
```

Control kiểu Label

Thiết kế form & control

Add control vào form

```
    public static void Main(string[] argv)
```

```
    {
```

```
        Application.Run(new Form1());
```

```
    }
```

```
}
```

```
}
```

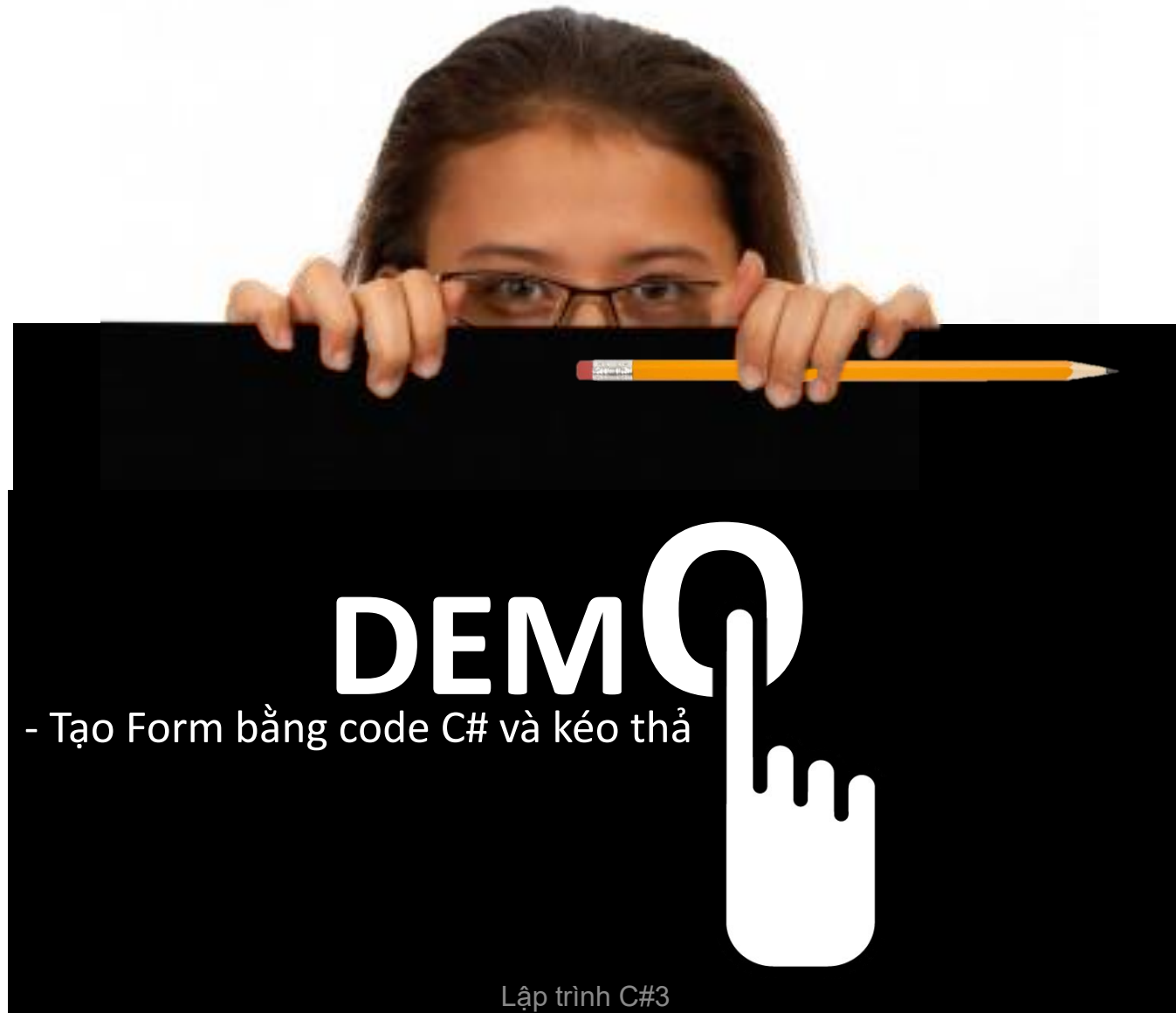
Chạy ứng dụng với
Form1 làm form chính

❑ Một số thuộc tính của control

Common Properties	Description
BackColor	Màu nền của control
BackgroundImage	Ảnh nền của control
ForeColor	Màu hiển thị text trên form
Enabled	Xác định khi control trạng thái enable
Focused	Xác định khi control nhận focus
Font	Font hiển thị text trên control
TabIndex	Thứ tự tab của control
TabStop	Nếu true, user có thể sử dụng tab để select control
Text	Text hiển thị trên form
TextAlign	Canh lề text trên control
Visible	Xác định hiển thị control

❑ Có các dạng control phổ biến:

- ❖ Điều khiển thông thường: Label, TextBox, Combobox, ListBox, CheckBox, RadioButton, Button...
- ❖ Điều khiển đặc biệt: Tooltip, HelpProvider, ErrorProvider, ProgressBar, ListView, TreeView, DateTimePicker, MonthlyCalender...
- ❖ Điều khiển Menu: MenuStrip, StatusStrip, ContextMenuStrip...
- ❖ Điều khiển container: GroupBox, TabControl, Panel...

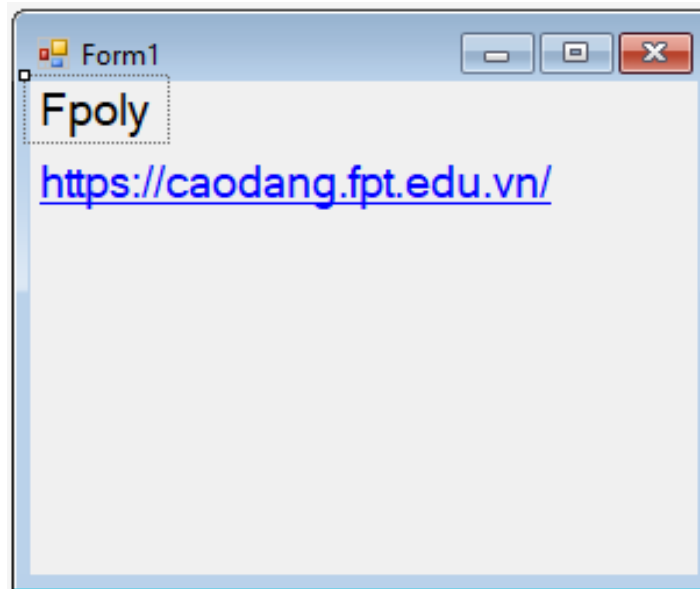
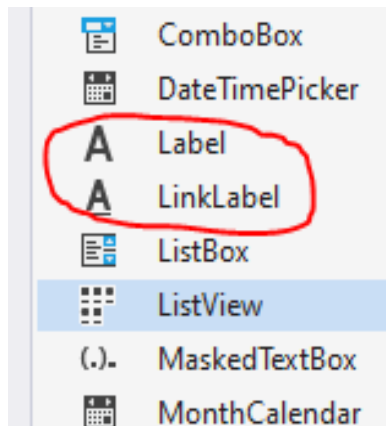




LẬP TRÌNH C# 3

BÀI 1: TỔNG QUAN WINDOW FORM (P2)

- ❑ Trình bày, chú giải tiêu đề
- ❑ Có hai loại Label và LinkLabel



Thuộc tính thường dùng

Font

Text

TextAlign

ForeColor

Visible

Font hiển thị của text

Nội dung text hiển thị

Canh lề text

Màu text

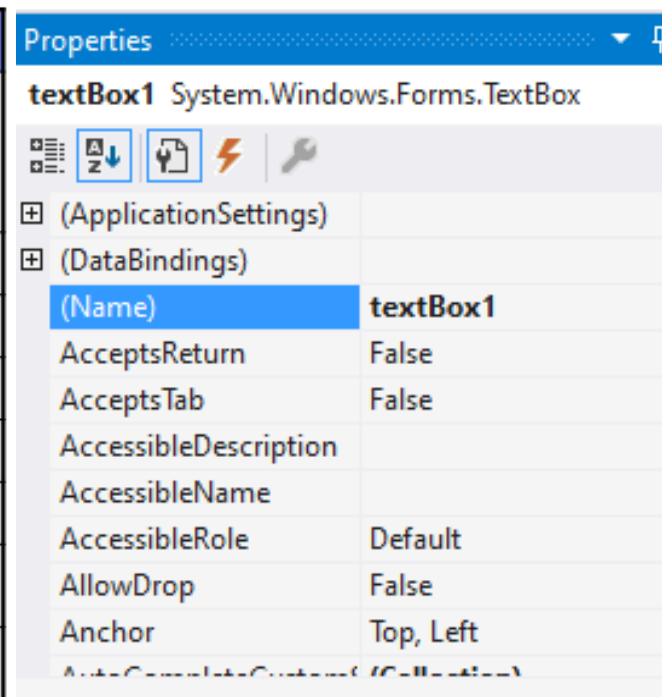
Trạng thái hiển thị

```
Label lb = new Label();
lb.Text = "This is Label Object";
this.Controls.Add(lb);
```

- ❑ Cho phép nhập dữ liệu
- ❑ Hai dạng là TextBox và MaskedTextBox
- ❑ Các hình thức nhập của TextBox là :PasswordChar và Multiline
- ❑ Lấy nội dung nhập : [ObjectName].Text = value

❑ Thuộc tính quan trọng

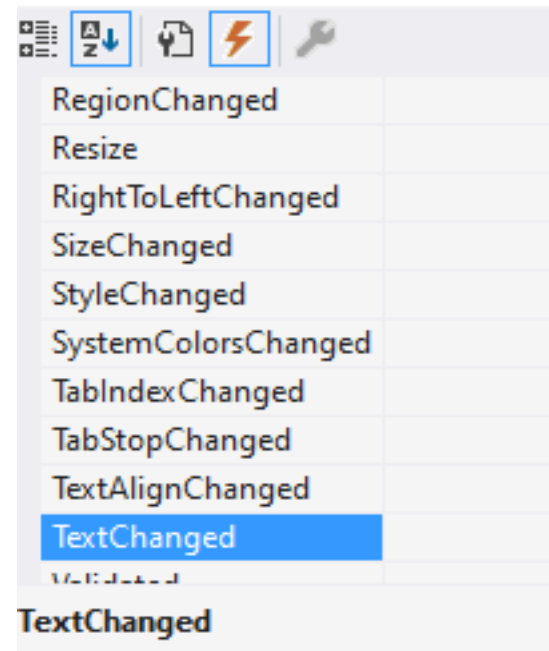
Tên thuộc tính	Ý nghĩa
Text	Lấy hoặc thiết lập giá trị của textbox. Đây là thuộc tính thể hiện dữ liệu mà tất cả các control đều có.
TextLength	Chiều dài tối đa của TextBox
TextAlign	Canh lề chữ trong TextBox
Multiline	Cho phép TextBox có thể nhiều hàng
PasswordChar	Kí tự hiện trong textbox dạng **
SelectedText	Chọn đoạn text được chọn
SelectionStart	Lấy vị trí của đoạn text được chọn
ReadOnly	Chỉ đọc, không thể thay đổi nội dung Text

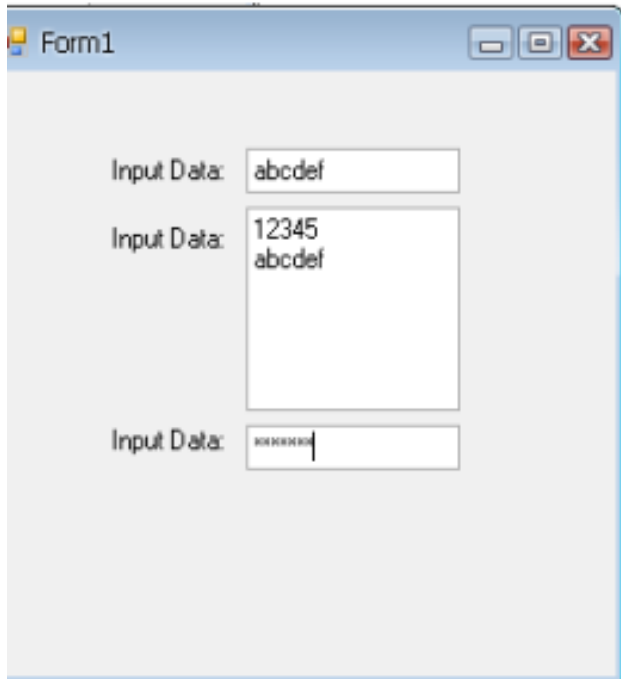


☐ Sự kiện quan trọng

Tên thuộc tính	Ý nghĩa
TextChanged	Sự kiện phát sinh khi người dùng thay đổi nội dung text.
Validating	Khi mất focus thì sự kiện này phát sinh để kiểm tra dữ liệu hợp lệ. Tham số: <code>object sender, CancelEventArgs e</code>
KeyPress	Bắt sự kiện gõ phím Tham số: <code>object sender, KeyPressEventArgs e</code>

textBox1 System.Windows.Forms.Text





Properties

textBox2 System.Windows.Forms.TextBox

AllowDrop	False
CharacterCasing	Normal
ContextMenuStrip	(none)
Enabled	True
HideSelection	True
ImeMode	NoControl
MaxLength	32767
Multiline	True
PasswordChar	
ReadOnly	False
ShortcutsEnabled	True
TabIndex	2
TabStop	True
UseSystemPasswordChar	False

Multiline
Controls whether the text of the edit control can span more than one line.

Properties

textBox3 System.Windows.Forms.TextBox

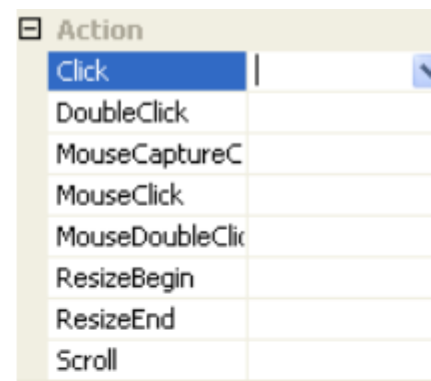
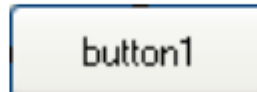
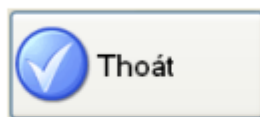
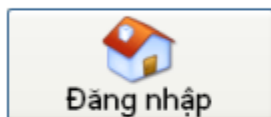
HideSelection	True
ImeMode	NoControl
MaxLength	32767
Multiline	False
PasswordChar	*
ReadOnly	False
ShortcutsEnabled	True
TabIndex	4
TabStop	True
UseSystemPasswordChar	False
Visible	True
WordWrap	True

PasswordChar
Indicates the character to display for password input for single-line edit controls.

```
private void textBox1_TextChanged(object sender, EventArgs e)
{
    //Code Here
}
```

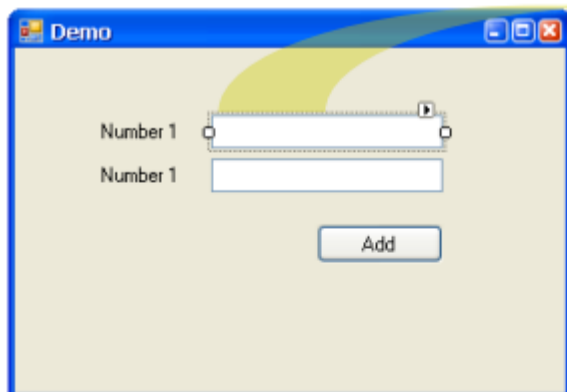
- ❑ Button sẽ ra lệnh cho ứng dụng thực hiện một hành động (method) khi có sự kiện tương tác (Mouse click, Key Press...)

- Dùng chuột nhấn để thao tác
- Có thể show Icon trên button

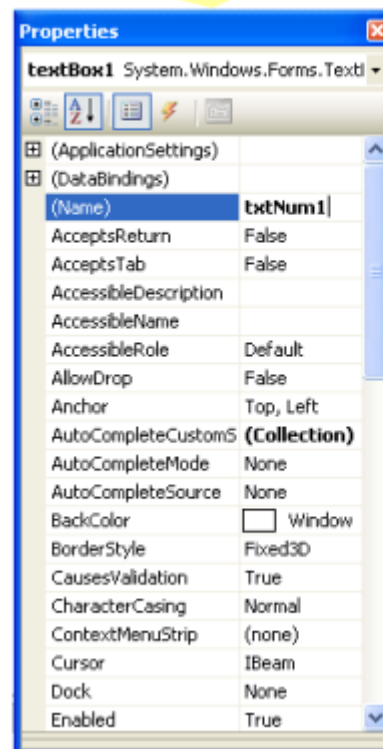


Button	
<i>Thuộc tính thường dùng</i>	
Text	Chuỗi hiển thị trên bề mặt button
<i>Event thường dùng</i>	
Click	Kích hoạt khi user kích vào button, khai báo mặc định khi người lập trình kích đúp vào button trong màn hình Design View của Form.

❑ Demo xây dựng ứng dụng tính tổng 2 số nguyên



Thay đổi các giá trị qua cửa sổ properties -> VS tự cập nhật code



Đổi tên thành
txtNum1

□ Phần xử lý

❖ Khi click vào Add -> cộng 2 giá trị và xuất kết quả

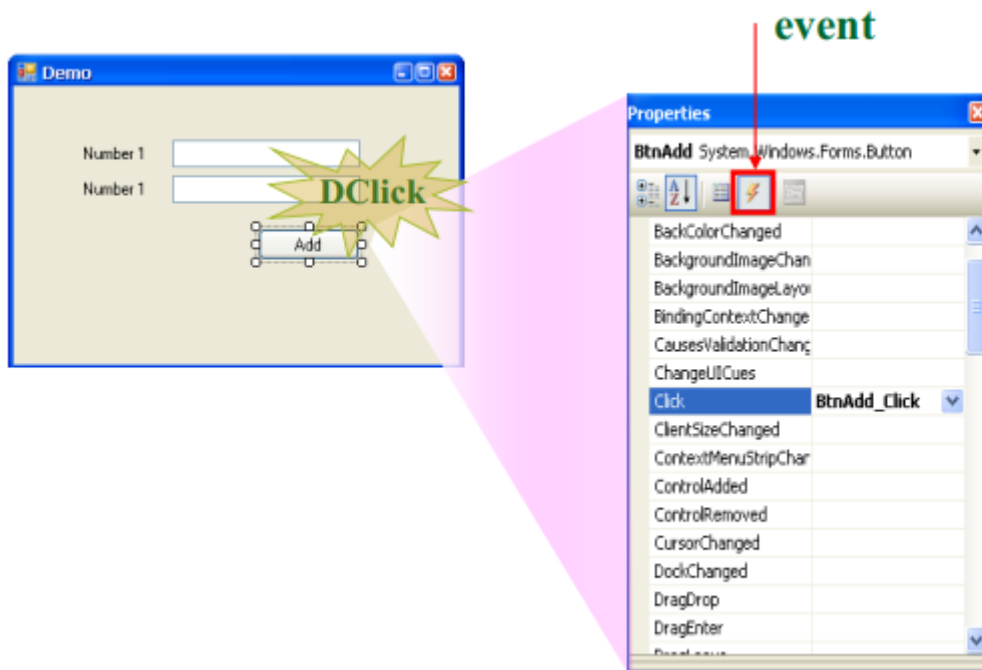
❖ Thực hiện

- Button Add cung cấp sự kiện click
- Form sẽ được cảnh báo khi Add được click
- Form sẽ được cảnh báo khi Add được click

❖ Cơ chế event

- Button đưa ra sự kiện click: đối tượng publish
- Form quan tâm đến sự kiện click của button, Form có sẽ phần xử lý ngay khi button click
- Phần xử lý của form gọi là Event Handler

- ❑ Khai báo event handler: Kích đúp vào button Add trên màn hình thiết kế cho phép tạo event handler cho sự kiện này



**Cửa sổ quản lý
event của BtnAdd**

□ Khai báo event handler:

```
//  
// BtnAdd  
//  
this.BtnAdd.Location = new System.Drawing.Point(180, 105);  
this.BtnAdd.Name = "BtnAdd";  
this.BtnAdd.Size = new System.Drawing.Size(75, 23);  
this.BtnAdd.TabIndex = 4;  
this.BtnAdd.Text = "Add";  
this.BtnAdd.UseVisualStyleBackColor = true;  
this.BtnAdd.Click += new System.EventHandler(this.BtnAdd_Click);
```

InitializeComponent

Sự kiện click

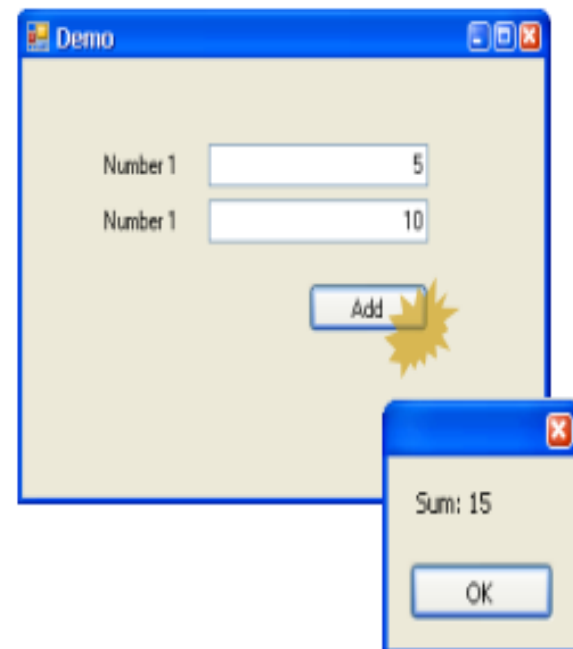
**Trình xử lý được gọi
khi event xảy ra**

Delegate chuẩn cho event handler

- ❑ Phần xử lý của Form1 khi button click: Lấy giá trị của 2 textbox, cộng kết quả và xuất ra MeesageBox

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

    private void BtnAdd_Click(object sender, EventArgs e)
    {
        int sum;
        sum = int.Parse(txtNum1.Text) + int.Parse(txtNum2.Text);
        MessageBox.Show("Sum: " + sum.ToString());
    }
}
```

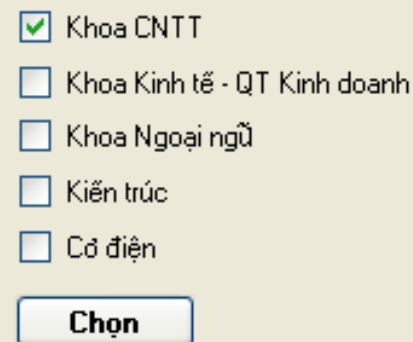


- ☐ Là điều khiển cho phép người sử dụng có thể chọn một trong 2 giá trị true/false hay yes/no
- ☐ Thuộc tính chính:

Thuộc tính	Ý nghĩa
Appearance	Hình dạng của điều khiển CheckBox
Checked	Trạng thái chọn (true) hay không chọn (false)
CheckState	Trạng thái của điều khiển: <ul style="list-style-type: none"> - Checked: chọn - Unchecked: không chọn - Indeterminate: vô định
ThreeState	Cho phép hay không cho phép 3 trạng thái.

❑ Ví dụ điều khiển CheckBox

```
private void btnChon_Click(object sender,
EventArgs e)
{
    string chon = "";
    foreach (Control c in Controls)
    {
        if (c is CheckBox)
        {
            if (((CheckBox)c).Checked == true) chon
+= c.Text + ", ";
        }
    }
    MessageBox.Show("Bạn đang chọn " + chon ,
"Thông báo!");
}
```



☒ Khoa CNTT
☐ Khoa Kinh tế - QT Kinh doanh
☐ Khoa Ngoại ngữ
☐ Kiến trúc
☐ Cơ điện

Chọn

- ❑ Là điều khiển cho phép người sử dụng có thể chọn tùy chọn trong danh sách, tuy nhiên trong khi **CheckBox** cho phép chọn nhiều tùy chọn cùng lúc thì **RadioButton** chỉ cho phép chọn 1 trong số các tùy chọn
- ❑ Thuộc tính chính

Thuộc tính	Ý nghĩa
Appearance	Hình dạng của điều khiển CheckBox
Checked	Trạng thái chọn (true) hay không chọn (false)

❑ Ví dụ sử dụng radiobutton



```
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton1.Checked)    MessageBox.Show("You are Single");
}

private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton2.Checked) MessageBox.Show("You are Married");
}

private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton2.Checked) MessageBox.Show("You are Divorce");
}
```



Tổng kết bài học

◎Thiết kế GUI - Window Form

◎Basic control





KẾT THÚC