

Ts. Trần Anh Tuấn

## Window Form (P1)

L/O/G/O

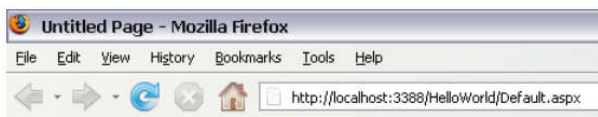
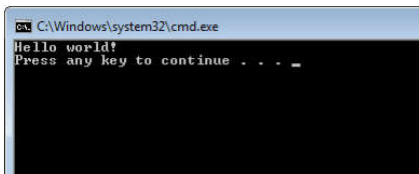
<http://dichvudanhvanban.com>

## Các loại ứng dụng C#

- Chương trình Console
  - Giao tiếp với người dùng bằng bàn phím
  - Không có giao diện đồ họa (GUI)
- Chương trình Windows Form
  - Giao tiếp với người dùng bằng bàn phím và mouse
  - Có giao diện đồ họa và xử lý sự kiện
- Chương trình Web Form
  - Kết hợp với ASP.NET, C# đóng vai trò xử lý bên dưới (underlying code)
  - Có giao diện đồ họa và xử lý sự kiện

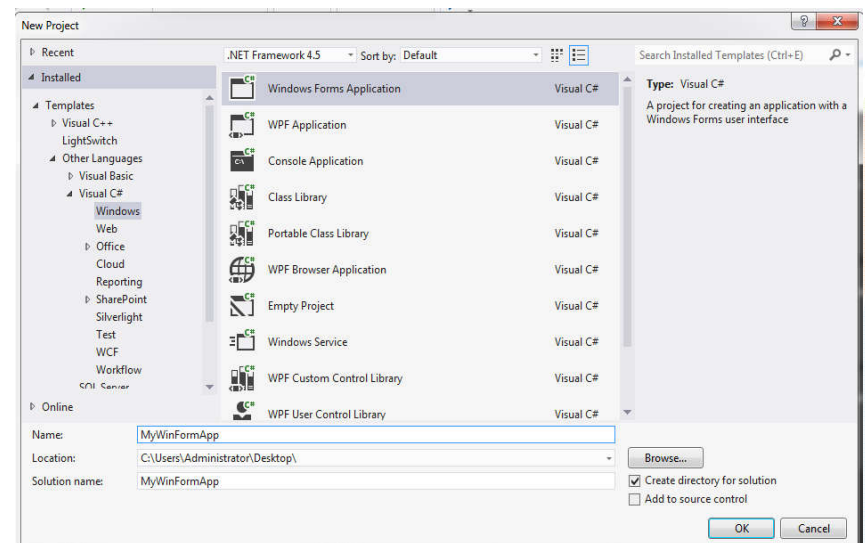


## Các loại ứng dụng C#



Hello World!

## Ứng dụng Window Form



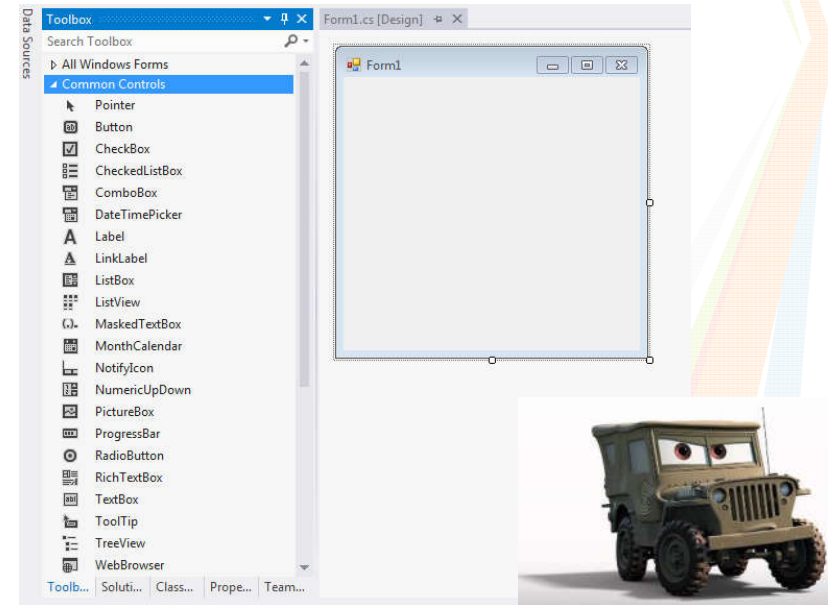
## Ứng dụng Window Form

- Để xem các control hỗ trợ ứng dụng Window Form ta vào View > ToolBox
  - Control là một thành phần cơ bản trên form
  - Có các thành phần – Thuộc tính (property) – Phương thức (method) – Sự kiện (event)
  - Tất cả các control chứa trong namespace: System.Windows.Forms



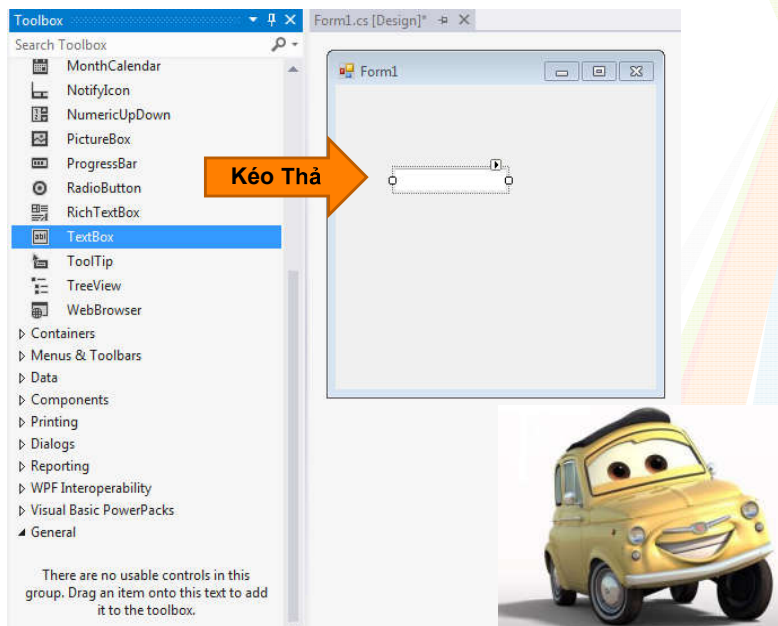
<http://dichvudanhvanban.com>

## Ứng dụng Window Form



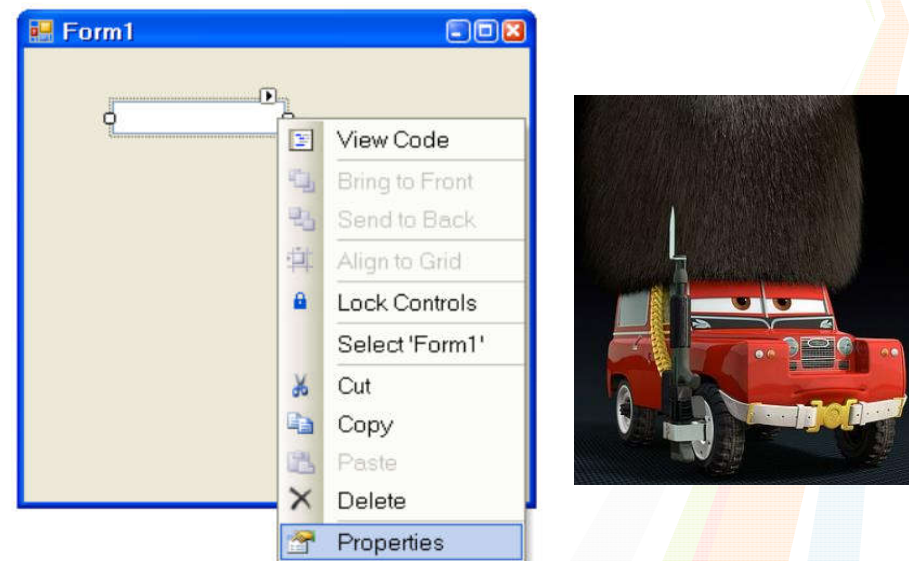
<http://dichvudanhvanban.com>

## Thêm Control vào Form



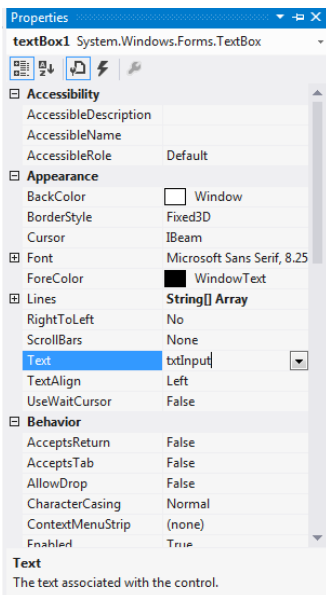
<http://dichvudanhvanban.com>

## Xem thuộc tính của một control



<http://dichvudanhvanban.com>

## Xem thuộc tính của control



## Thuộc tính của Control

- Ngoài thuộc tính quan trọng nhất Name: định nghĩa tên Control để sử dụng còn các thuộc tính khác sau

### Một số thuộc tính của control

- Text:** mô tả text xuất hiện trên control
- Focus:** phương thức chuyển focus vào control
- TabIndex:** thứ tự của control nhận focus
  - Mặc định được VS.NET thiết lập
- Enable:** thiết lập trạng thái truy cập của control
- Visible:** ẩn control trên form, có thể dùng phương thức Hide
- Anchor:**
  - Neo giữ control ở vị trí xác định
  - Cho phép control di chuyển theo vị trí
- Size:** xác nhận kích thước của control



## Thuộc tính của Control

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


## Thuộc tính của Form

Tên thuộc tính	Ý nghĩa
AutoSize	Cho phép kéo giãn form
CauseValidation	Cho phép các control con của nó có thể phát sinh sự kiện Validate. Dùng để kiểm tra dữ liệu hợp lệ.
TopMost	Cho phép Form hiện trên tất cả các form khác
WindowState	{Normal, Minimized, Maximized}. Thể hiện của Form khi xuất hiện
Cursor	Thiết lập con trỏ chuột
Icon	Thiết lập Icon của form



## ➤ Thuộc tính của Control

- Mỗi control sẽ được xác định bằng thuộc tính Name. Thuộc tính này dùng để xác định control khi lập trình
- Lưu ý: Cần đặt tên Name sao cho gọi nhớ

Control	Tiếp đầu ngữ	Biểu tượng
Textbox	<b>txt</b> Ví dụ: txtHoTen, txtTuoi	 TextBox
Button	<b>btn</b> Ví dụ: btnOK, btnCancel	 Button
Label	<b>lbl</b> Ví dụ: lblThongBao	 Label

<http://dichvudanhvanban.com>

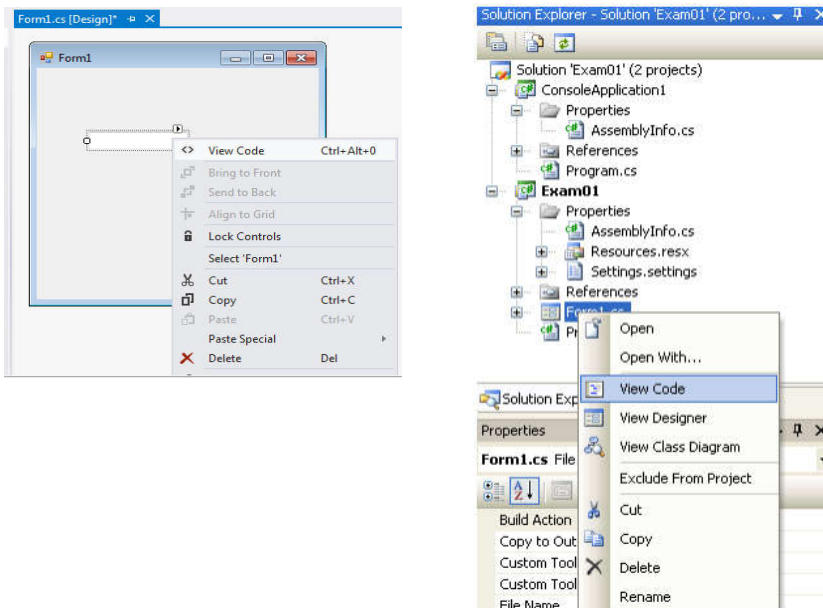
## ➤ Ứng dụng Window Form

- Một window form có 2 view:
  - Design View: Dùng để thiết kế giao diện
  - Code View: Dùng để lập trình trên form (Xác định các control trên design view thông qua thuộc tính Name của từng control)



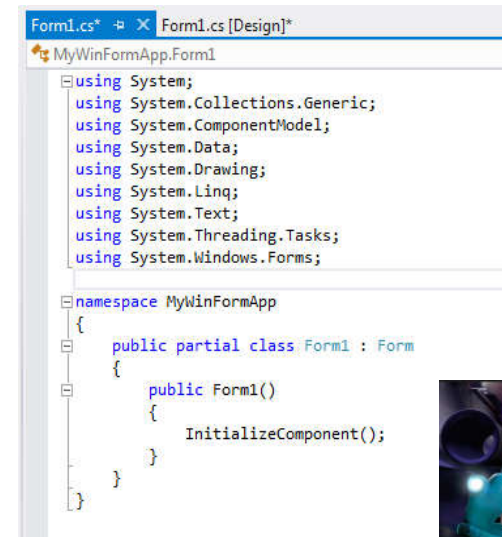
<http://dichvudanhvanban.com>

## ➤ Ứng dụng Window Form



<http://dichvudanhvanban.com>

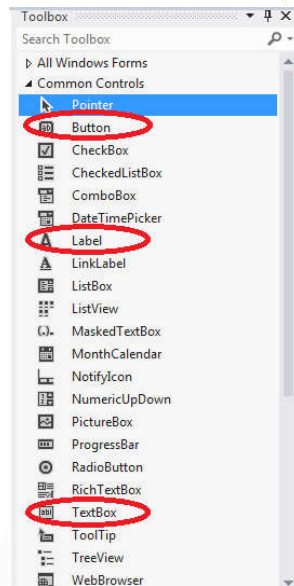
## ➤ Ứng dụng Window Form



<http://dichvudanhvanban.com>

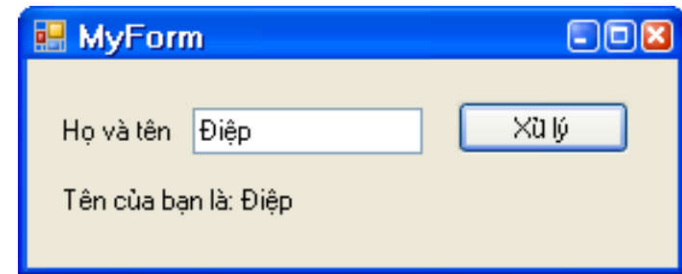
## Các control cơ bản

- Form : Khung chính ứng dụng
- Button : Nút nhấn thao tác
- Textbox: Nhập thông tin
- Label : Nhãn hiển thị thông tin



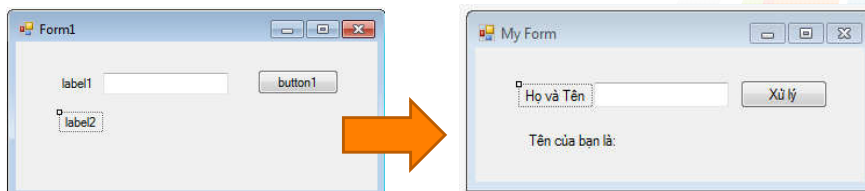
## Ví dụ 1

- Xây dựng ứng dụng cho phép nhập thông tin tên người dùng và hiển thị tên đó thông qua việc nhấn nút Xử Lý



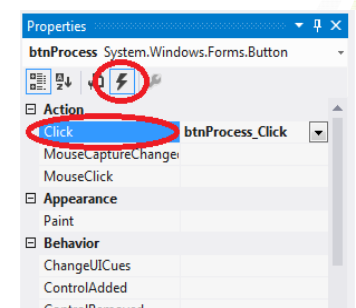
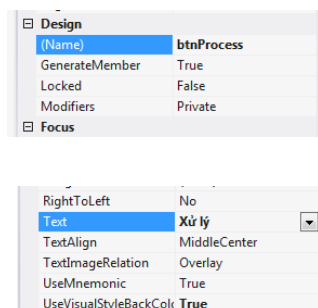
## Ví dụ 1

- Kéo thả các control: Label, Button, Textbox vào khung Form
- Đặt tên các control, thay đổi thuộc tính Text để hiển thị
  - Button: Xử lý (Text), btnProcess(Name)
  - Label: Họ và Tên (Text), lblName (Name) ; Tên của bạn là (Text), lblInfo (Name)
  - Textbox: txtName (Name)



## Ví dụ 1

- Chọn nút Event(hình tia sét) và hành động Click. Nhấp đúp chuột vào Click hoặc nhập tên function mong muốn

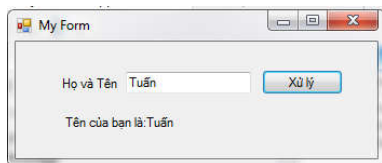


## Ví dụ 1

- Viết code cho nút xử lý: Hiển thị thông tin người dùng nhập từ textbox txtName trên label lblInfo

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

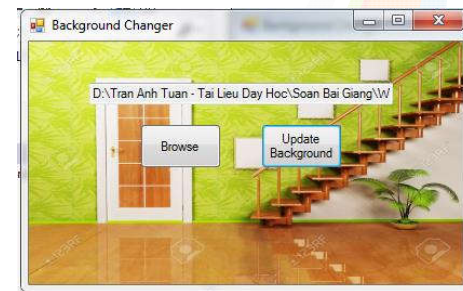
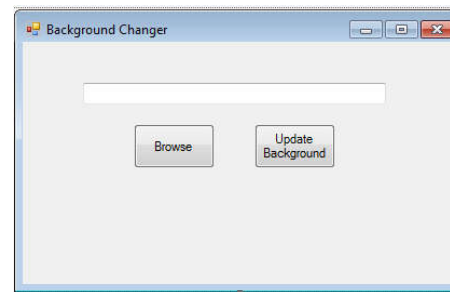
        private void btnProcess_Click(object sender, EventArgs e)
        {
            lblInfo.Text = "Tên của bạn là:" + txtName.Text;
        }
    }
}
```



<http://dichvudanhvanban.com>

## Ví dụ 2

- Xây dựng ứng dụng cho phép người dùng chọn một ảnh từ thư mục dự án và dùng ảnh đó thay đổi màu nền của Form khi người dùng nhấn nút "Update Background"



<http://dichvudanhvanban.com>

## Ví dụ 2

- Đặt Name và Text cho các control và Form
- Form

(ApplicationSettings)	
(DataBindings)	
(Name)	FrmMain
AcceptButton	(none)
AccessibleDescription	

Tag	
Text	Background Changer
TopMost	False
TransparencyKey	
UseWaitCursor	False

Leave	
Load	FrmMain_Load
LocationChanged	
MaximizedBoundsChang	

- TextBox



(DataBindings)	
(Name)	txtImagePath
AcceptsReturn	False
AcceptsTab	False

## Ví dụ 2

- Button Browse

(DataBindings)	
(Name)	btnBrowse
AccessibleDescription	
AccessibleName	

Tag	
Text	Browse
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseCompatibleTextRender	False

Click	btnBrowse_Click
ClientSizeChanged	

- Button

(DataBindings)	
(Name)	btnUpdateBg
AccessibleDescription	
AccessibleName	

Tag	
Text	Update Background
TextAlign	MiddleCenter
TextImageRelation	Overlay

Click	btnUpdateBg_Click
ClientSizeChanged	
ContextMenuStripChang	

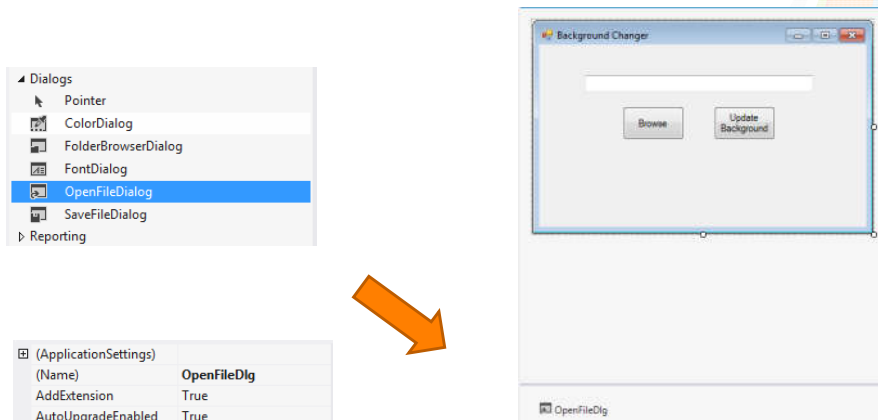
<http://dichvudanhvanban.com>

<http://dichvudanhvanban.com>



## Ví dụ 2

- Do chúng ta có dùng chức năng chọn file theo thư mục của Window nên chúng ta kéo thả thêm đối tượng OpenFileDialog nằm trong Toolbox>Dialogs



<http://dichvudanhvanban.com>

## Ví dụ 2

- Trong sự kiện Form Load ta dùng lớp Directory để lấy đường dẫn thư mục ứng dụng

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;

public partial class FrmMain : Form
{
    string startupPath;

    public FrmMain()
    {
        InitializeComponent();
    }

    private void FrmMain_Load(object sender, EventArgs e)
    {
        startupPath = Directory.GetParent(Directory.GetCurrentDirectory()).Parent.FullName;
        txtImagePath.ReadOnly = true;
    }
}
```

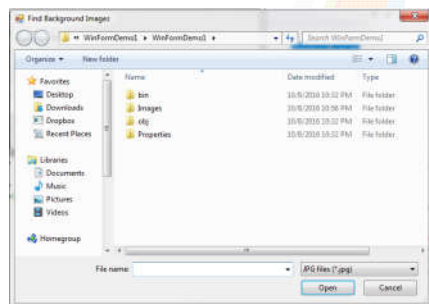
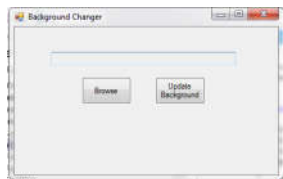


<http://dichvudanhvanban.com>

## Ví dụ 2

- Trong sự kiện Browse Click dùng OpenFileDialog để định thư mục mở, loại file mở và lưu file ảnh chọn vào Textbox

```
private void btnBrowse_Click(object sender, EventArgs e)
{
    OpenFileDialog.Title = "Find Background Images";
    OpenFileDialog.Filter = "JPG files|*.jpg";
    OpenFileDialog.InitialDirectory = startupPath;
    if (OpenFileDialog.ShowDialog() == DialogResult.OK)
    {
        txtImagePath.Text = OpenFileDialog.FileName;
    }
}
```

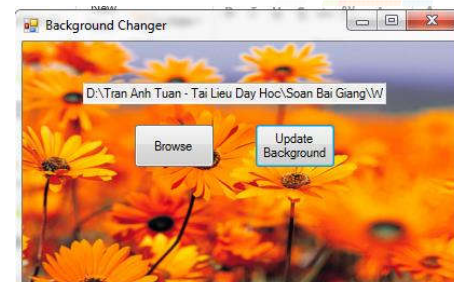
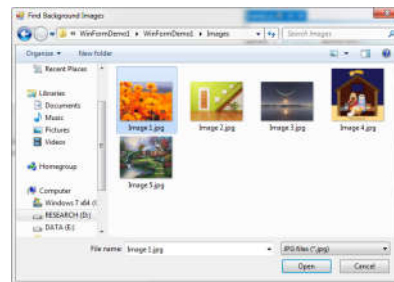


<http://dichvudanhvanban.com>

## Ví dụ 2

- Trong sự kiện nhấn nút Update Background ta thực hiện thay đổi thuộc tính BackgroundImage của Form

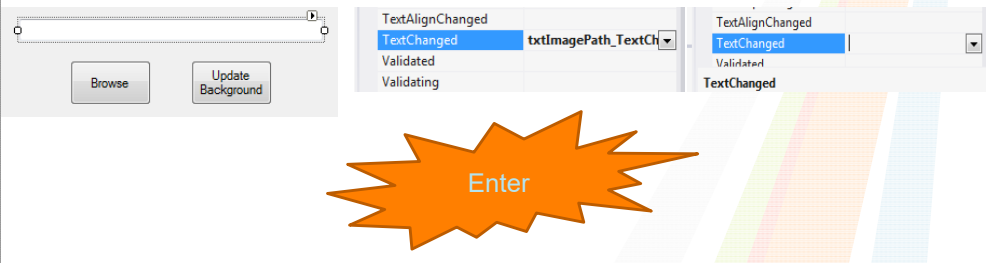
```
private void btnUpdateBg_Click(object sender, EventArgs e)
{
    BackgroundImage = Image.FromFile(txtImagePath.Text);
    BackgroundImageLayout = ImageLayout.Stretch;
}
```



<http://dichvudanhvanban.com>

## Ví dụ 2

- Trong TH ta lỡ tạo một event của một control mà muốn xóa chúng đi thì thực hiện các bước sau:
  - Chọn control
  - Vào mục event (hình tia chớp)
  - Chọn phần tên function mà event đó thực hiện. Xóa đi và nhấn enter
  - Vào Form.cs kiểm tra lại thấy event đó mất là thành công



## Ví dụ 2

- Xem thử tập tin Form1.Designer.cs hàm InitializeComponent

```
private void InitializeComponent()
{
    this.txtImagePath = new System.Windows.Forms.TextBox();
    this.btnBrowse = new System.Windows.Forms.Button();
    this.btnUpdateBg = new System.Windows.Forms.Button();
    this.OpenFileDialog = new System.Windows.Forms.OpenFileDialog();
    this.SuspendLayout();
    // txtImagePath
    this.txtImagePath.Location = new System.Drawing.Point(56, 38);
    this.txtImagePath.Name = "txtImagePath";
    this.txtImagePath.ScrollBars = System.Windows.Forms.ScrollBars.Horizontal;
    this.txtImagePath.Size = new System.Drawing.Size(281, 20);
    this.txtImagePath.TabIndex = 0;

    // btnBrowse
    this.btnBrowse.Location = new System.Drawing.Point(103, 76);
    this.btnBrowse.Name = "btnBrowse";
    this.btnBrowse.Size = new System.Drawing.Size(75, 41);
    this.btnBrowse.TabIndex = 1;
    this.btnBrowse.Text = "Browse";
    this.btnBrowse.UseVisualStyleBackColor = true;
    this.btnBrowse.Click += new System.EventHandler(this.btnBrowse_Click);

    // btnUpdateBg
    this.btnUpdateBg.Location = new System.Drawing.Point(215, 76);
    this.btnUpdateBg.Name = "btnUpdateBg";
    this.btnUpdateBg.Size = new System.Drawing.Size(75, 41);
    this.btnUpdateBg.TabIndex = 2;
    this.btnUpdateBg.Text = "Update Background";
    this.btnUpdateBg.UseVisualStyleBackColor = true;
    this.btnUpdateBg.Click += new System.EventHandler(this.btnUpdateBg_Click);

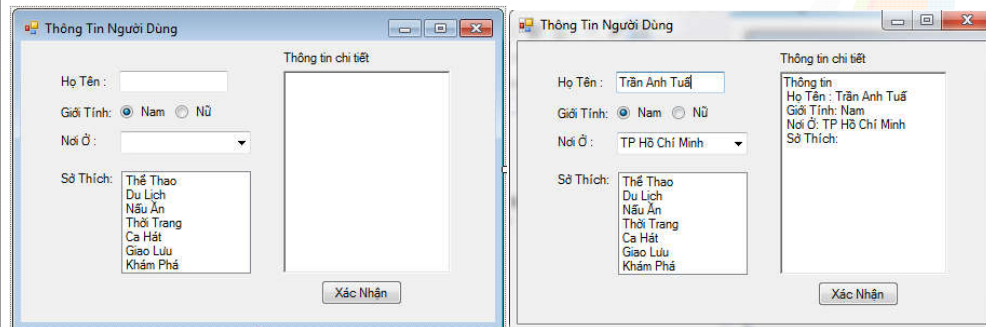
    // FrmMain
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(402, 224);
    this.Controls.Add(this.btnBrowse);
    this.Controls.Add(this.txtImagePath);
    this.Name = "FrmMain";
    this.Text = "Background Changer";
    this.Load += new System.EventHandler(this.FrmMain_Load);
    this.ResumeLayout(false);
    this.PerformLayout();

}

private System.Windows.Forms.TextBox txtImagePath;
private System.Windows.Forms.Button btnBrowse;
private System.Windows.Forms.Button btnUpdateBg;
private System.Windows.Forms.OpenFileDialog openFileDialog;
```

## Ví dụ 3

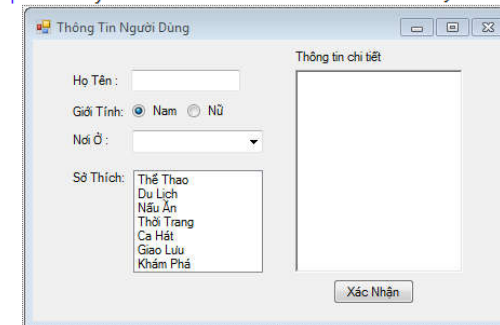
- Xây dựng ứng dụng cho phép người dùng nhập thông tin và những thay đổi sẽ được cập nhật ở khung "Thông tin chi tiết"
- Sau khi nhấn nút xác nhận các thông tin đó sẽ được hiển thị qua một hộp thoại để nhắc người dùng xác nhận lại.



## Ví dụ 3

- Thiết lập các Name và Text cho từng control. Sau đây là các Name tương ứng (xem trong Form1.Designer.cs)

```
private System.Windows.Forms.TextBox txtName;
private System.Windows.Forms.RadioButton rdFemale;
private System.Windows.Forms.RadioButton rdMale;
private System.Windows.Forms.ComboBox cmbAccommodation;
private System.Windows.Forms.ListBox lsHobby;
private System.Windows.Forms.RichTextBox rtSummary;
private System.Windows.Forms.Button btnConfirm;
```





## Ví dụ 3

- Viết hàm UpdateInputInfo để cập nhật là nội dung sẽ xuất khi có bất kỳ sự thay đổi thông tin nào từ người dùng

```
public string UpdateInputInfo()
{
    string str;
    str = String.Format("Thông tin \n Họ Tên : {0} \n Giới Tính: {1} \n Nơi Ở: {2} \n",
        txtName.Text,
        (rdMale.Checked ? "Nam" : "Nữ"),
        cmbAccommodation.SelectedItem.ToString());
    str += " Sở Thích: \n\t";
    foreach (Object selecteditem in lsbHobby.SelectedItems)
    {
        str += selecteditem.ToString() + "\n\t";
    }
    return str;

    private void Form1_Load(object sender, EventArgs e)
    {
        rtxtSummary.Text = UpdateInputInfo();
    }
}
```

<http://dichvudanhvanban.com>

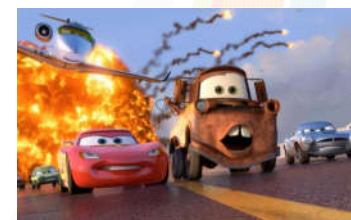
## Ví dụ 3

- Gọi hàm này trong các sự kiện tương ứng với mỗi control

```
private void txtName_TextChanged(object sender, EventArgs e)
{
    rtxtSummary.Text = UpdateInputInfo();
}

private void cmbAccommodation_SelectedIndexChanged(object sender, EventArgs e)
{
    rtxtSummary.Text = UpdateInputInfo();
}

private void lsbHobby_SelectedIndexChanged(object sender, EventArgs e)
{
    rtxtSummary.Text = UpdateInputInfo();
}
```



<http://dichvudanhvanban.com>

## Ví dụ 3

- Riêng đối với hai radio button rdMale và rdFemale, ta có thể gộp chung event thành một event handler (hay một phương thức)
  - Chọn control rdMale
  - Chọn sự kiện **CheckedChanged** trong tập các sự kiện (hình tia chớp)
  - Nhập tên Event Handler là **GenderChanged** xong enter
  - Chọn control rdFemale
  - Trong sự kiện **CheckedChanged** của rdFemale, tao nhấn combobox xổ xuống và chọn lại hàm **GenderChanged**
  - Như vậy cả hai event của rdMale và rdFemale đều thực thi hành động event handler (phương thức) **GenderChanged**

## Ví dụ 3



```
private void GenderChanged(object sender, EventArgs e)
{
    rtxtSummary.Text = UpdateInputInfo();
}
```

<http://dichvudanhvanban.com>

<http://dichvudanhvanban.com>

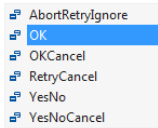
## Ví dụ 3

- Riêng trong sự kiện nhấn nút Xác nhận, ta gọi hàm `MessageBox.Show` để hiện thị thông tin qua hộp thoại

```
private void btnConfirm_Click(object sender, EventArgs e)
{
    MessageBox.Show(UpdateInputInfo(), "Thông Tin Cá Nhân", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

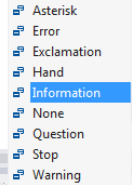
```
MessageBoxButtons.OK, MessageBoxIcon.Information);
```

);



```
MessageBoxButtons.OK, MessageBoxIcon.Information);
```

);



## Ví dụ 3



# Thank You!

L/O/G/O

<http://dichvudanhvanban.com>