



CHƯƠNG 4: WINDOWS FORM

Thời gian: 120 phút

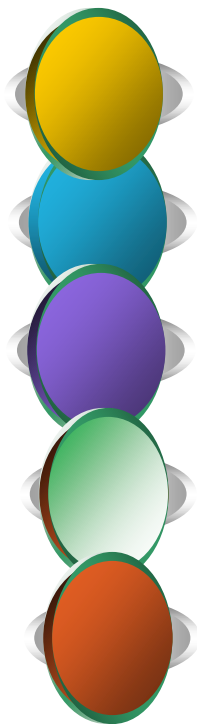
Người trình bày: HỒ LÊ VIỆT NIN

Email: holevietnin@duytan.edu.vn

Điện thoại: 0905455246



NỘI DUNG



Graphical User Interface (GUI)

Giới thiệu Winform

Form đơn giản

Bài Tập

Thảo luận



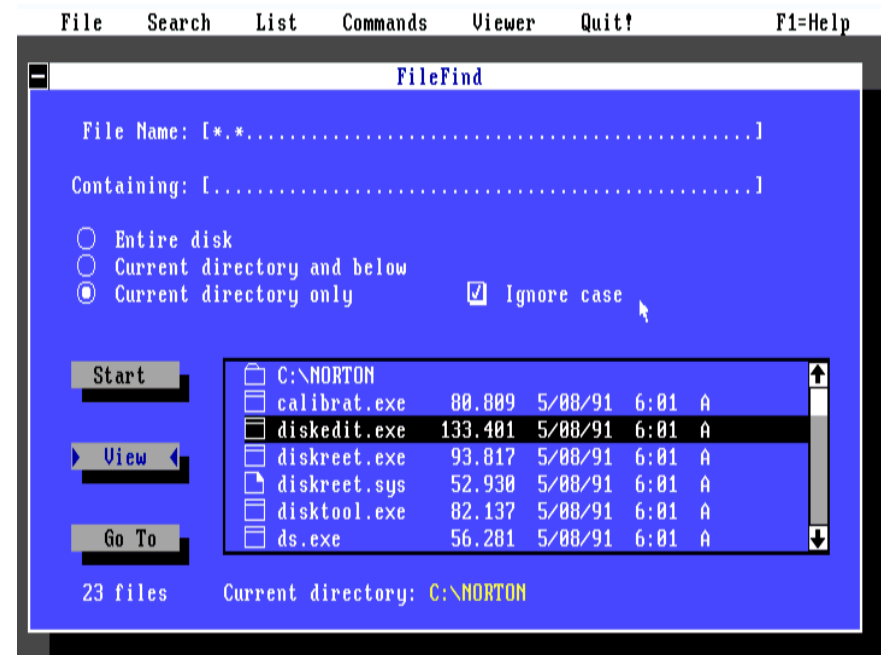
GUI

Command line interface: CLI

```
--- rr.chtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/ndev = 112.076/112.076/112.076/0.000 ms
bash-2.05b$ grep -i /dev/sda /etc/fstab | cut --fields=3
/dev/sda1          /mnt/usbkey
/dev/sda2          /mnt/ipod
bash-2.05b$ date
Wed May 25 11:36:56 PDT 2005
bash-2.05b$ lsmod
Module                Size  Used by
joydev                 8256  0
ipw2200              175112  0
ieee80211             44228  1 ipw2200
ieee80211_crypt       4872   2 ipw2200,ieee80211
e1000                 84468  0
bash-2.05b$
```

Tương tác qua keyboard
Thực thi tuần tự

Text user interface: TUI



GUI dựa trên text
Mức độ tương tác cao hơn



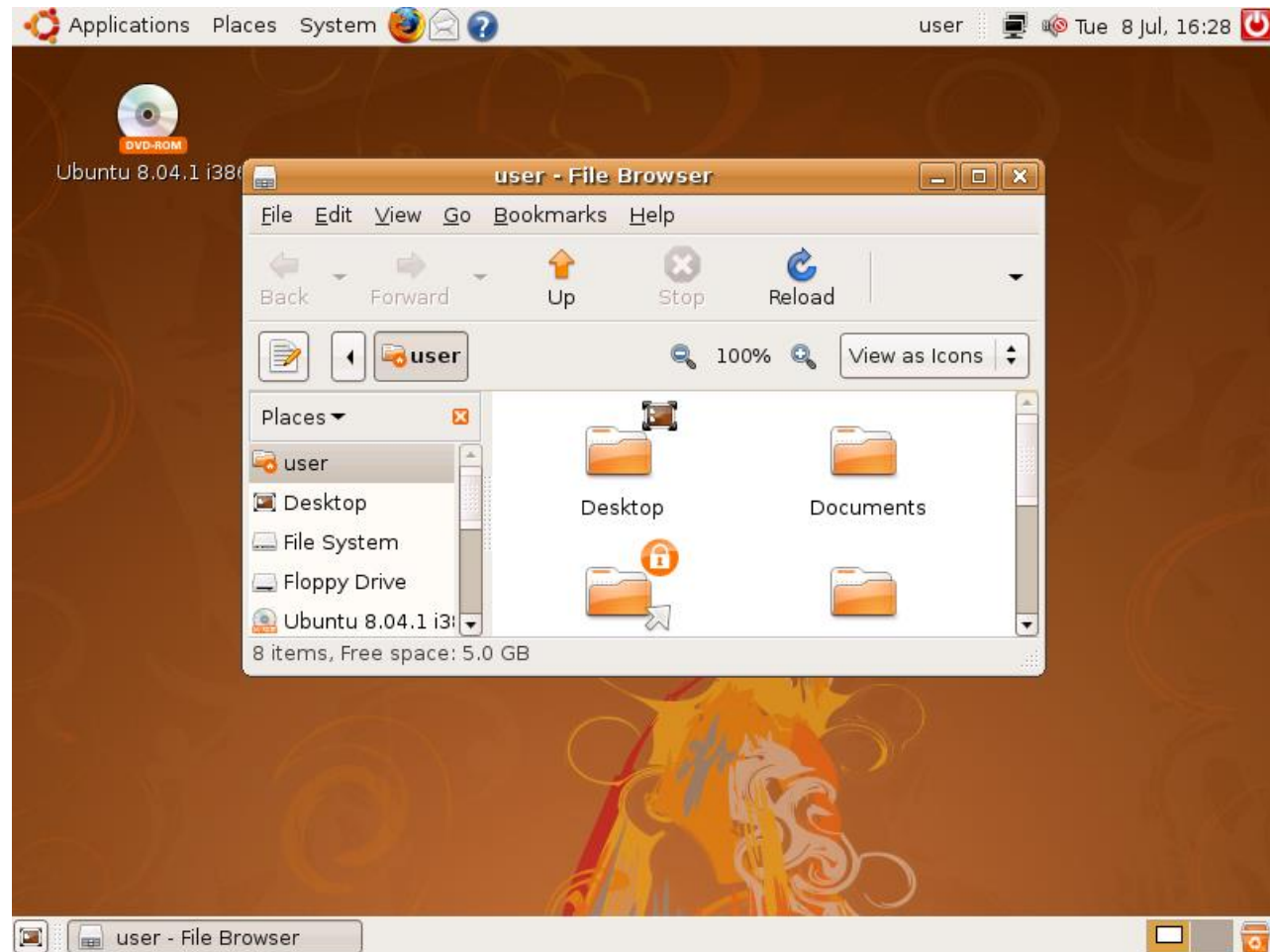
GUI

Graphical User Interface: GUI

Tương tác qua
giao diện đồ họa
độ phân giải cao

Đa số các hệ OS
hiện đại đều dùng
GUI

Cho phép user dễ
dàng thao tác





GUI

- ❖ Chương trình hiện đại đều dùng GUI
- ❖ **Graphical**: text, window, menu, button...
- ❖ **User**: người sử dụng chương trình
- ❖ **Interface**: cách tương tác chương trình
- ❖ Thành phần đồ họa điển hình



GUI Application

❖ Windows Form là nền tảng GUI cho ứng dụng desktop

- (Ngược với Web Form ứng dụng cho Web)
- Single Document Interface (SDI)
- Multiple Document Interface (MDI)

❖ Các namespace chứa các lớp hỗ trợ GUI trong .NET

- **System.Windows.Forms**: Chứa GUI components/controls và form
- **System.Drawing**: Chức năng liên quan đến tô vẽ cho thành phần GUI, cung cấp chức năng truy cập đến GDI+ cơ bản



Event- Driven Programming

Cách truyền thống

Danh sách các lệnh thực thi tuần tự

Việc kế tiếp xảy ra chính là lệnh tiếp theo trong danh sách

Chương trình được thực thi bởi máy tính

Event-Driven Programming

Các đối tượng có thể kích hoạt sự kiện và các đối tượng khác phản ứng với những sự kiện đó

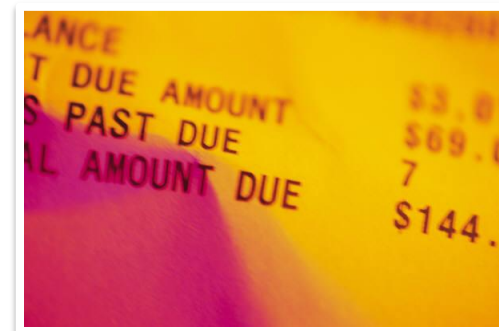
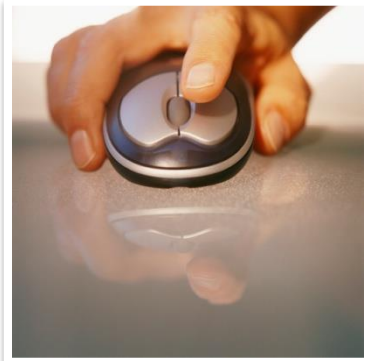
Việc kế tiếp xảy ra phụ thuộc vào sự kiện kế tiếp

Luồng chương trình được điều khiển bởi sự tương tác User-Computer



Event-Driven Programming

- ❖ Chương trình GUI thường dùng Event-Drive Programming
- ❖ Chương trình chờ cho event xuất hiện và xử lý
- ❖ Ví dụ sự kiện:

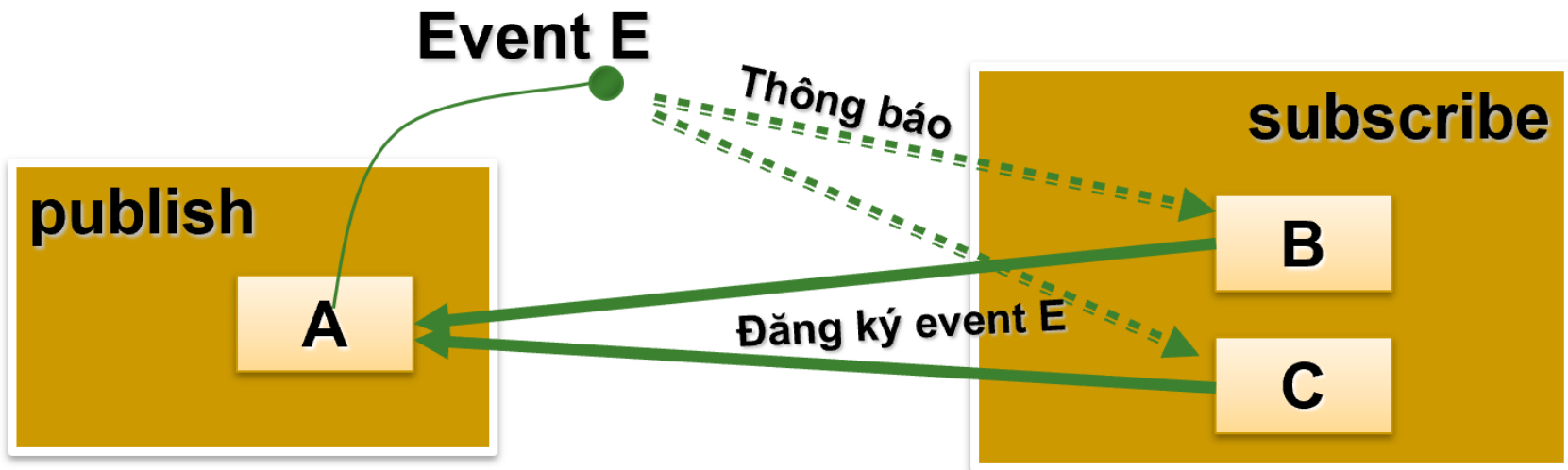


- ❖ Firing an event: khi đối tượng khởi tạo sự kiện
- ❖ Listener: đối tượng chờ cho sự kiện xuất hiện
- ❖ Event handler: phương thức phản ứng lại sự kiện



Event-Driven Programming

❖ Trong C#, Event-Driven Programming được thực thi bởi event (*xem slide Delegate & Event*)





Event-Driven Programming

❖ Minh họa xử lý trong form

*User nhập text
vào textbox →
click Button để
add chuỗi nhập
vào listbox*

Click

invoke

Lấy dữ liệu từ
textbox
Add vào listbox

Button đưa ra sự kiện click

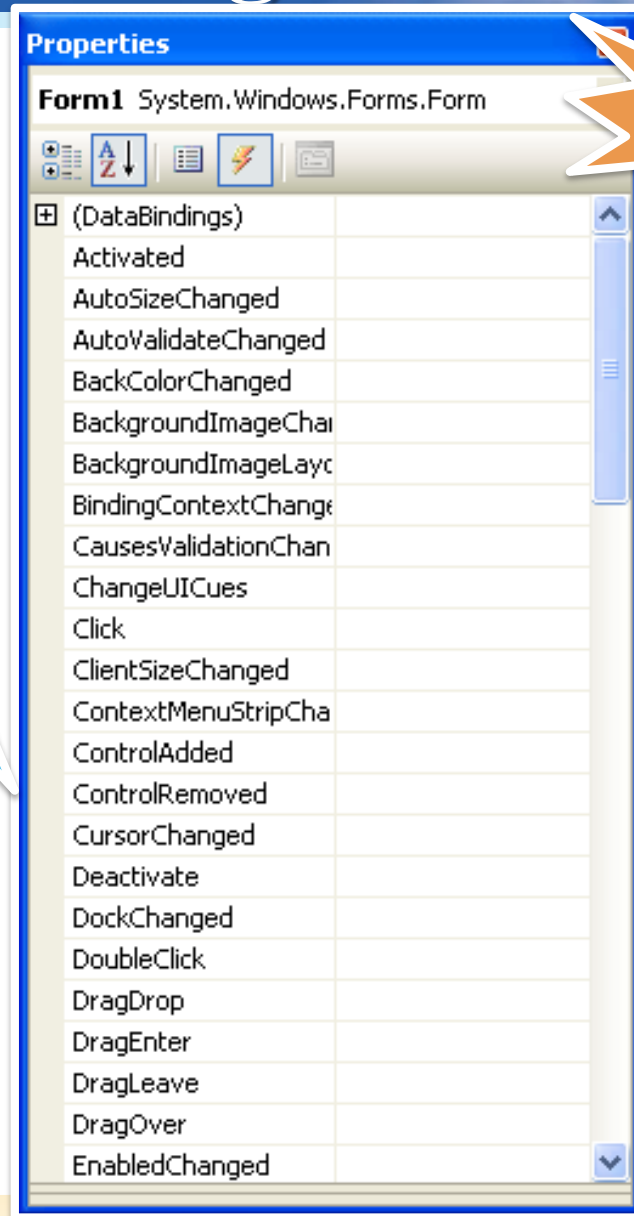
Form có event handler cho click của button



Event-Driven Programming

❖ GUI-based events

- Mouse move
- Mouse click
- Mouse double-click
- Key press
- Button click
- ...



Danh sách
event cho
Form



GIỚI THIỆU WINFORM

❖ Video tham khảo

https://www.youtube.com/watch?time_continue=52&v=dtYVRWfGhzI&feature=emb_logo



GIỚI THIỆU WINFORM



Windows Forms Application



GIỚI THIỆU WINFORM

- ❖ Sử dụng GUI làm nền tảng
- ❖ Event-driven programming cho các đối tượng trên form
- ❖ Ứng dụng dựa trên một “**form**” chứa các thành phần
 - ❖ Menu
 - ❖ Toolbar
 - ❖ StatusBar
 - ❖ TextBox, Label, Button...
- ❖ Lớp cơ sở cho các form của ứng dụng là **Form**

System.Windows.Forms. Form

Namespace Class



GIỚI THIỆU WINFORM

Màn hình chính

Hệ thống Xem danh mục Quản lý danh mục hóa đơn Quản lý danh mục theo nhóm Thông tin

QUẢN LÝ BÁN HÀNG

Người thực hiện:

Email liên hệ:



GIỚI THIỆU WINFORM

The screenshot shows a web-based application titled "Employees". The interface is divided into several sections for data entry and management:

- Employee Info:** Fields for First Name (Shirley Ann), Last Name (Smith), Gender (Female), Position (Secretary), DOB (16 March 1955), Age (51 yrs 10 months), Date Started (10 April 2003), Service (1 yrs 3 months), and Date Left.
- Address:** Fields for Address (1 Sea View), City (Basingstoke), County (East Yorkshire), and Post Code (RG22 0UY).
- Contact Info:** Fields for Phone ((01185) 185111) and Email (someone@winform.co.uk).
- Holiday Entitlement:** A field for Days Per Year (20).
- Paye Info:** Fields for Tax Code (0N), New / Month / Basic (checked), and Pay Per Hr (8.00).
- Personal Details:** Fields for Member (YK24894), NTable (A), and a checkbox for "Check if Male".
- Notes:** A large text area for additional information.

Navigation icons at the bottom include "New", "List All", "First", "Previous", "Next", "Last", "Help", "Print", "Delete", "Add", "Edit", and "Home".



GIỚI THIỆU WINFORM

PHẦN MỀM QUẢN LÝ CAFE v3.3 TUKI - HOTLINE (08) 383 363 76

HỆ THỐNG HOẠT ĐỘNG KHO HÀNG QUỸ NHÂN SỰ QUẢN TRỊ TRỢ GIÚP

Sử dụng dịch vụ Quản lý bán hàng Doanh thu Thống kê mặt hàng Tồn kho Ghi chú nhanh Mẹo vặt Báo cáo Thoát

Sử dụng dịch vụ

Khu A Khu B

Bàn A1 Bàn A2 Bàn A3 Bàn A4 Bàn A5 Bàn A6

Bàn A7 Bàn A8 Bàn A9

Khu C

Bàn C1 Bàn C2 Bàn C3 Bàn C4 Bàn C5 Bàn C6

Bàn C7 Bàn C8 Bàn C9

Mô bản Ngày: 25/05/2017 Số TT: 1

Khách hàng: An

Đặt số lượng Đổi giá bán Ghi chú

Mặt hàng SL Đơn giá Thành tiền

Số lượng 1

<< Thêm Giảm >> Xóa

In chế biến

Thanh toán (F11)

Thống kê (F9)

Tìm (F3):

- Tất cả
- COM
- Trà
- Sinh Tố
- Nước ép hoa quả
- Kem
- Nước Hoa Quả
- Hoa Quả Dầm

Mặt hàng	ĐVT	Giá
B52	ly	30,000
B53	ly	50,000
B54	ly	60,000
Bạc sữa	ly	20,000
Bát vị la hán trà	ly	30,000
Bí đao	ly	15,000
Bia Heniken	ly	22,000
Bia HN	ly	15,000
Bim bim	gói	5,000
Bơ dầm	ly	35,000
Bò húc	ly	15,000
Bột sắn	ly	20,000
Cafe bột sắn	ly	20,000
Cafe đen	ly	16,000
Cafe nâu	ly	18,000
Cafe sữa	ly	0
Cam lon	ly	15,000
Cam rum	ly	40,000

Tiền hàng: 0

Phí dịch vụ: 0 0

Giảm giá: 0.00 0



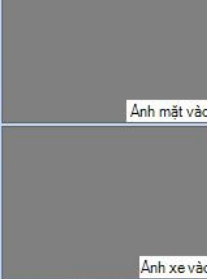









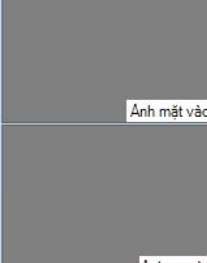

Tổng cộng: 0

Phiên bản dùng thử, hết hạn trong 4 ngày.

CSDL: DEMO.FDB Tài khoản: Admin

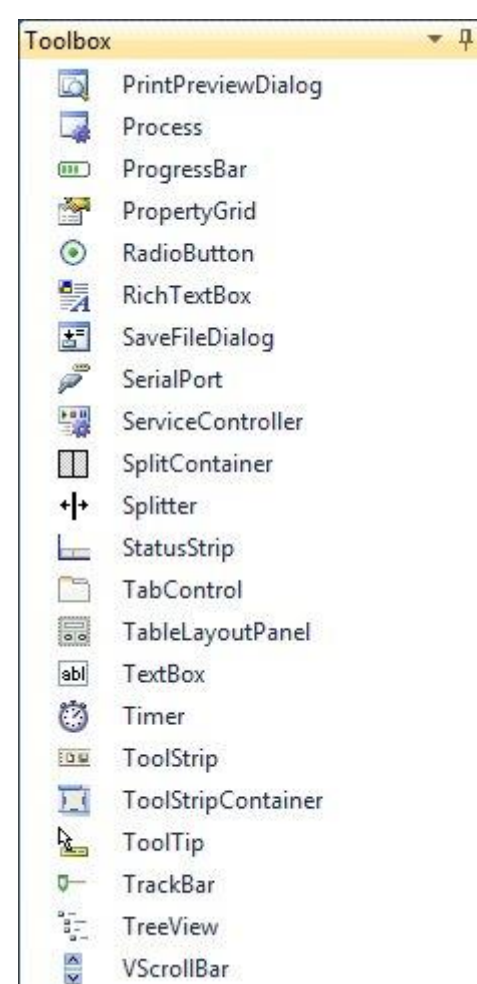
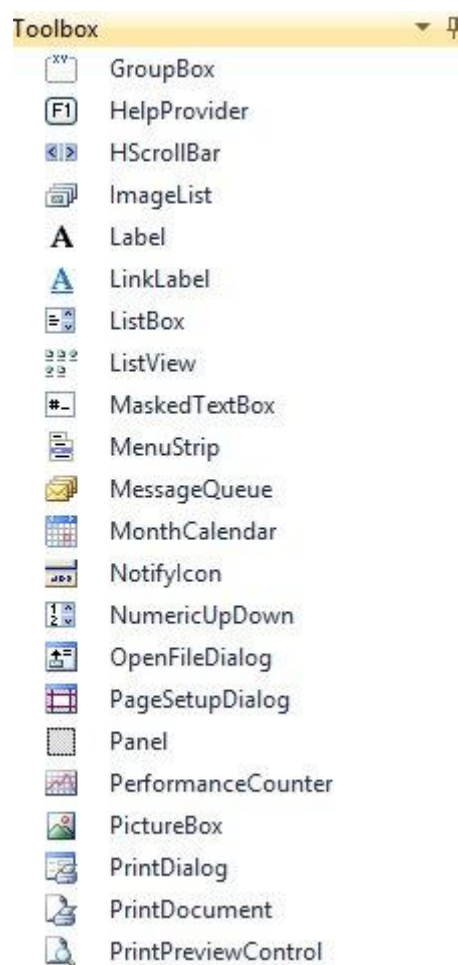
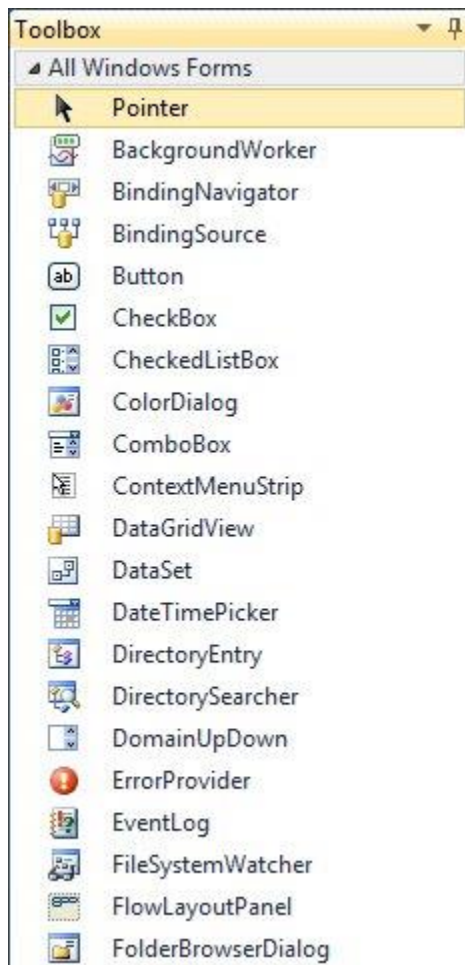


GIỚI THIỆU WINFORM

Camera biển	Camera mặt		F5	Thông tin xe lần 1	
				Công ty : <input type="text"/>	Anh biển ra
				Biển ĐK : <input type="text"/>	Anh biển vào
				Vào : <input type="text"/>	
				Ra : <input type="text"/>	
				0 ngày 0 giờ 0 phút	
				0 VND	
				Không có xe vào	
Camera biển	Camera mặt		F6	Thông tin xe lần 2	
				Công ty : G5	Anh biển ra
				Biển ĐK : 23R8-5689	Anh biển vào
				Vào : 08:30 01/03/2014	
				Ra : 08:31 01/03/2014	
				Thanh toán : 0 ngày 0 giờ 0 phút	
				00 VND	
				Vé tháng	Cập nhật thông tin xe ra thành công
Camera biển	Camera mặt		F7	Thông tin xe ô tô lần 3	
				Công ty : <input type="text"/>	Anh biển ra
				Biển ĐK : <input type="text"/>	Anh biển vào
				Vào : <input type="text"/>	
				Ra : <input type="text"/>	
				0 ngày 0 giờ 0 phút	
				0 VND	
				Không nhận được biển số	



GIỚI THIỆU WINFORM



Toolbox của Visual Studio .NET 2010



GIỚI THIỆU WINFORM





GUI Components/Controls

- ❖ **Components/controls được tổ chức vào các lớp thừa kế, cho phép dễ dàng chia sẻ các thuộc tính**
- ❖ **Mỗi component/control định nghĩa các**
 - **Property (có thể thay đổi trong màn hình thiết kế)**
 - **Method**
 - **Event**



GUI Components/Controls

- ❖ Cách dễ nhất là sử dụng VS .NET Toolbox để thêm control và component vào form (in design view)
- ❖ Sử dụng code để thêm các control/component vào form (in code view)



UD WinForm đơn giản

Form1.cs

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

```
namespace HaGiang
{
```

```
    public class Form1 : Form
```

```
    {
        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);
```

```
        }
```

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

```
        {
```

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

```
        }
```

```
    }
```

Lớp Form cơ sở

Control kiểu Label

Thiết kế form & control

Add control vào form

Chạy ứng dụng với
Form1 làm form chính

Các bước tạo UD WinForm cơ bản

- ❖ Tạo lớp kế thừa từ lớp Form cơ sở
- ❖ Bổ sung các control vào form
 - Thêm các label, menu, button, textbox...
- ❖ Thiết kế layout cho form (bố trí control)
 - Hiệu chỉnh kích thước, trình bày, giao diện cho
 - form
 - Control chứa trong form



Các bước tạo UD WinForm cơ bản

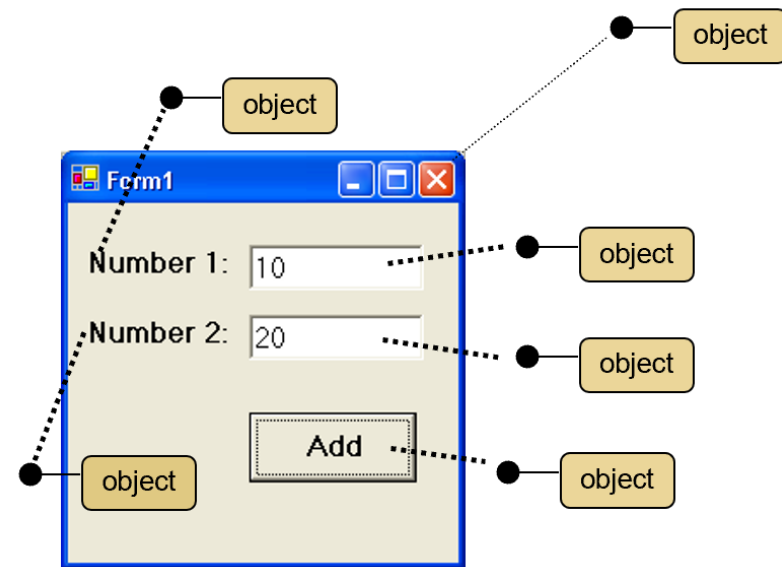
- ❖ **Viết các xử lý cho các control trên form và các xử lý khác**
- ❖ **Hiển thị Form**
 - Thông qua lớp Application gọi phương thức Run

Nên sử dụng IDE hỗ trợ thiết kế GUI!



Form và control

- ❖ Tất cả các thành phần trên form đều là đối tượng
- ❖ Các control là những lớp của FCL
 - System.Windows.Forms.Label
 - System.Windows.Forms.TextBox
 - System.Windows.Forms.Button





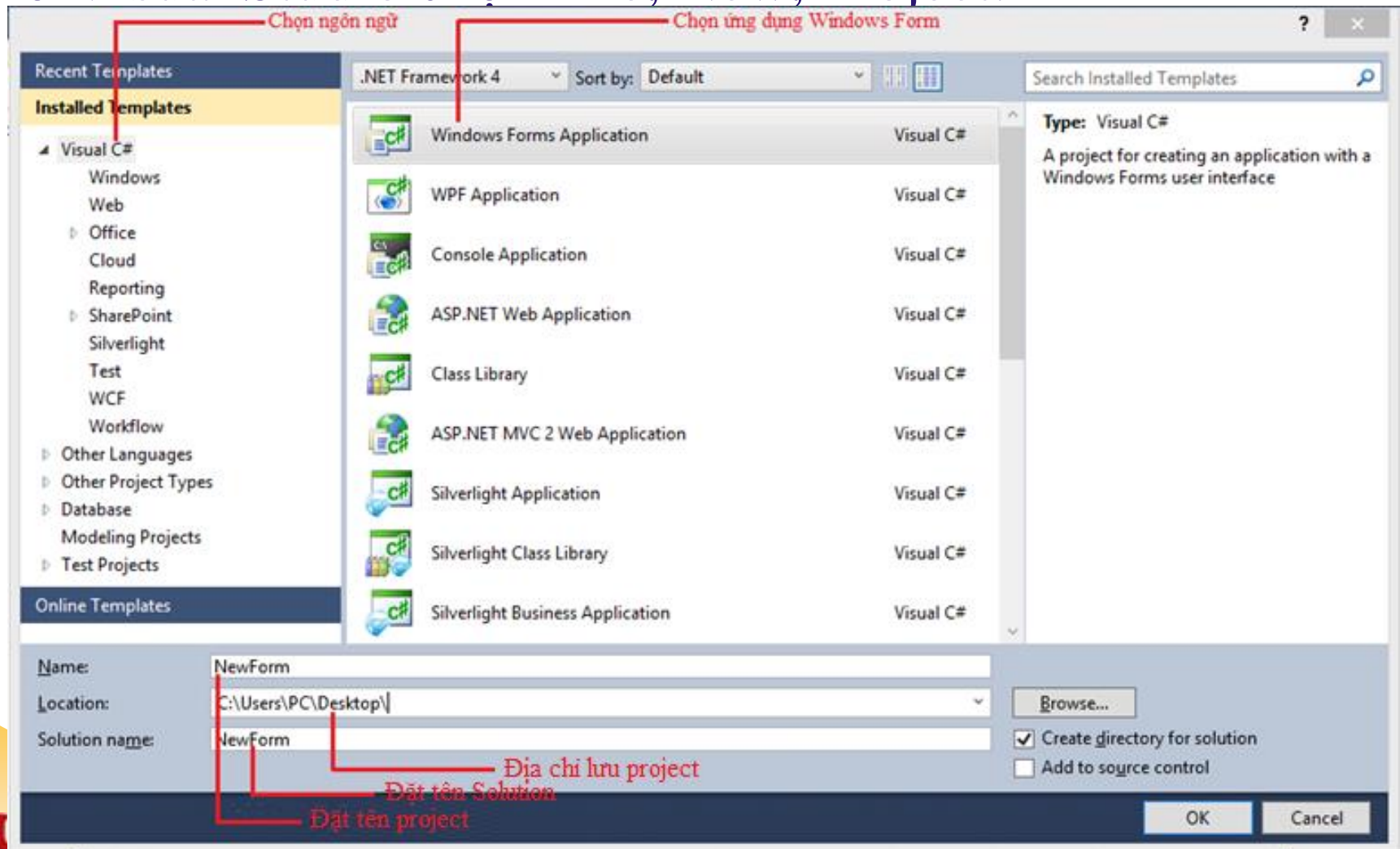
Các thuộc tính của Form

❖ Video tham khảo

<https://www.youtube.com/watch?v=Szk4W6F4zhk>

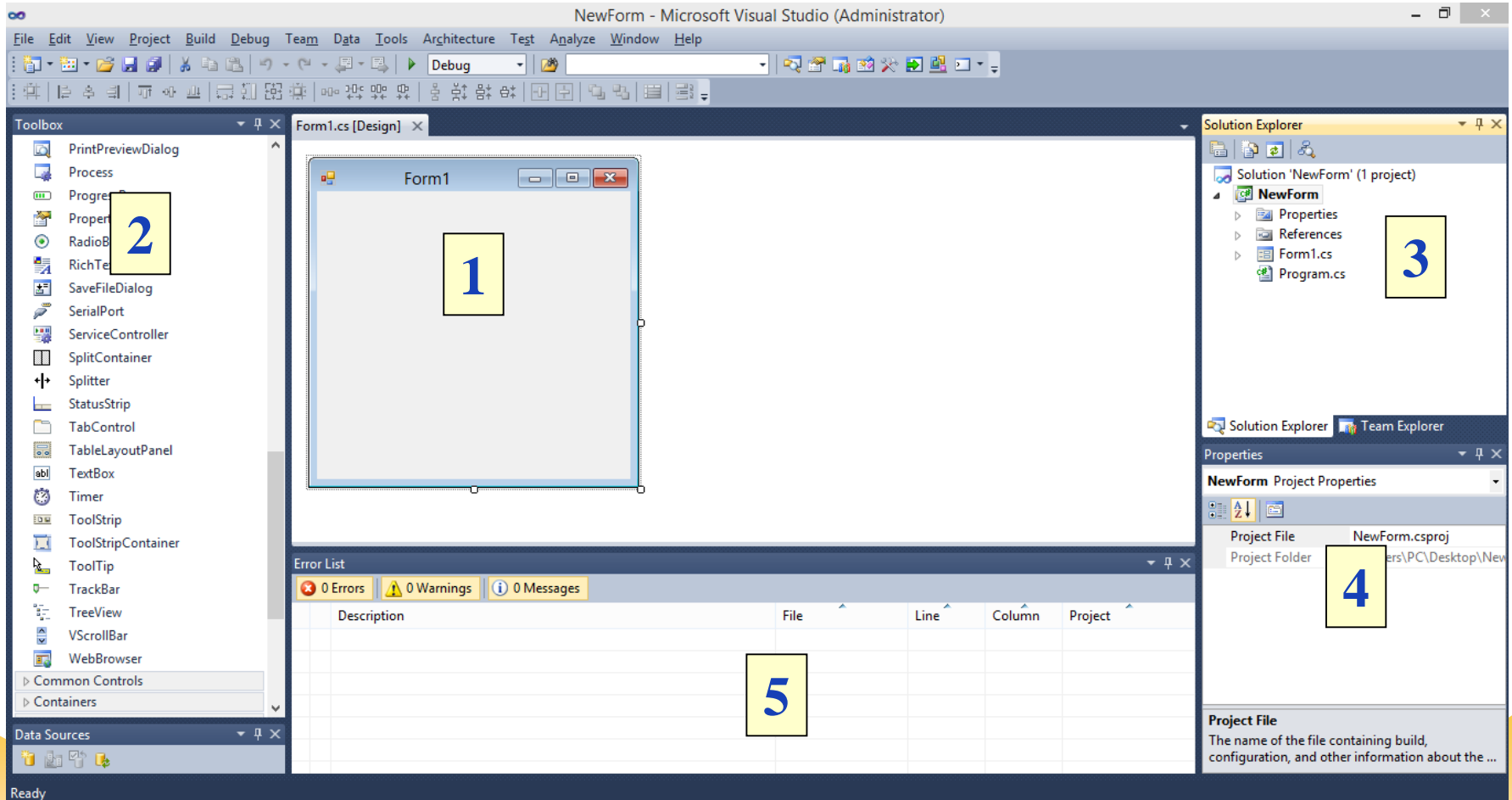
TAO MỘT ỨNG DỤNG WINFORM

Mở Visual Studio chọn File, New, Project.



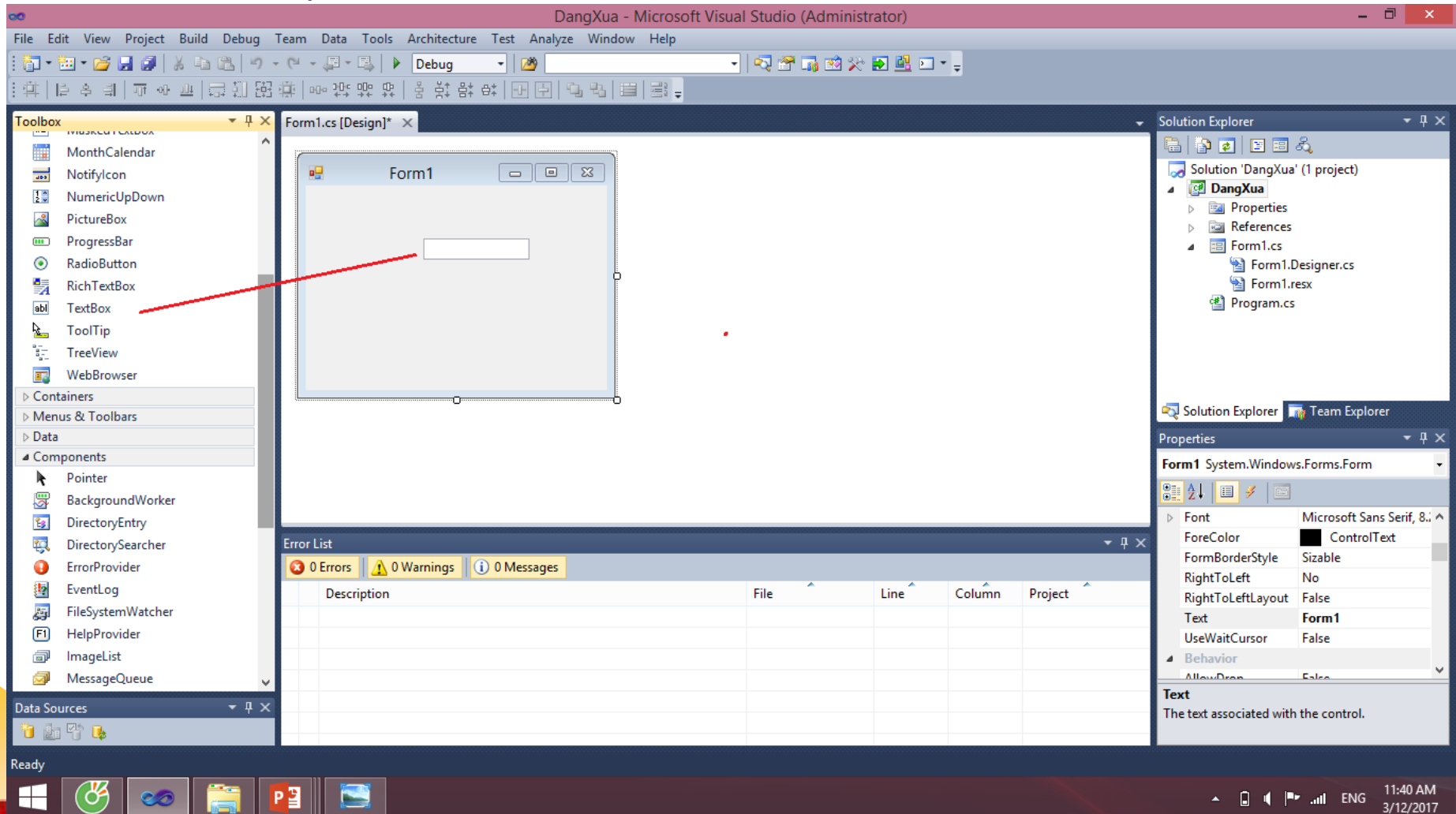
TAO MỘT ỨNG DỤNG WINFORM

Màn hình chính Winform



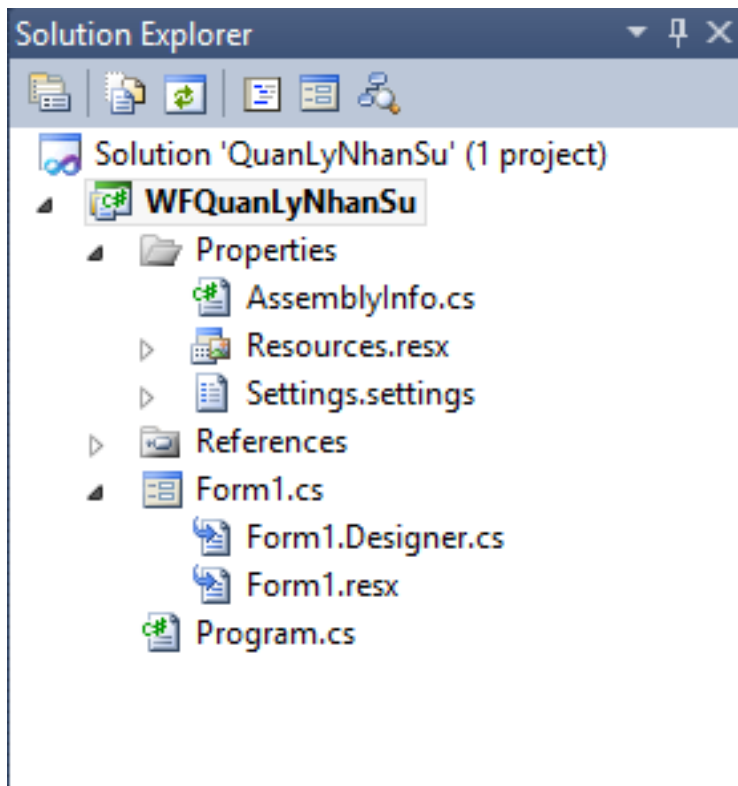
TAO MỘT ỨNG DỤNG WINFORM

Kéo thả một control



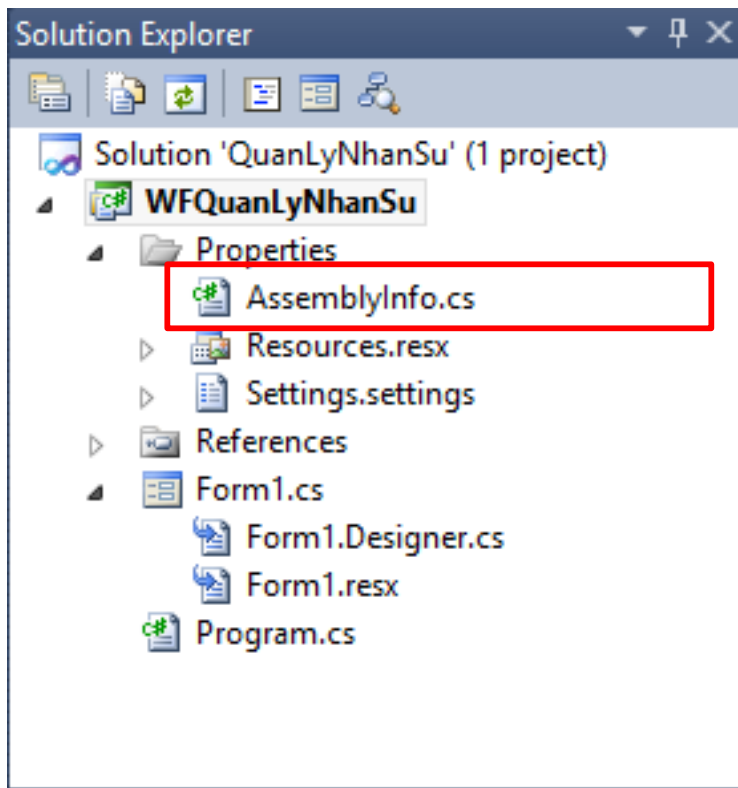


PROJECT CỦA CT C#





PROJECT CỦA CT C#



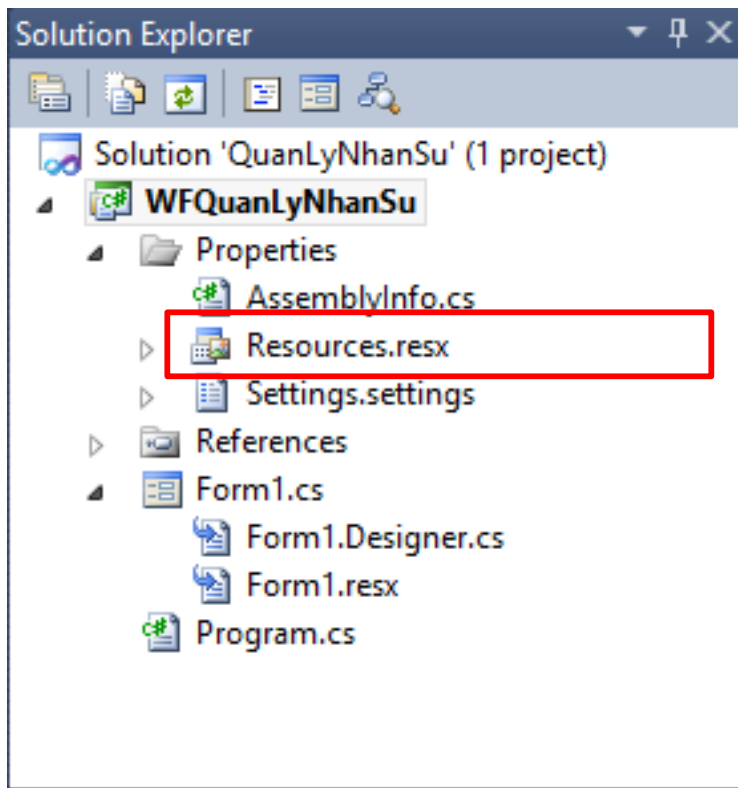
AssemblyInfo.cs

Dùng để lưu thông tin về Assembly

```
[assembly: AssemblyTitle("WFQuanLyNhanSu")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("WFQuanLyNhanSu")]
[assembly: AssemblyCopyright("Copyright © 2017")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]
[assembly: ComVisible(false)]
[assembly: Guid("b7e419c1-3207-494d-a5d4-261f5c5c2598")]
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

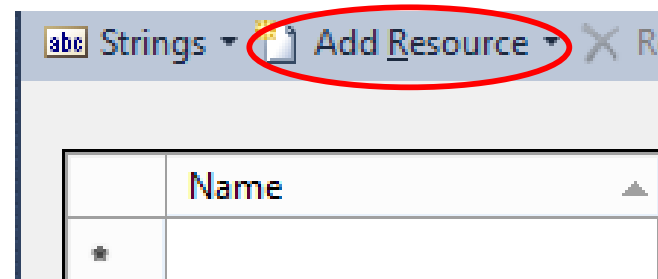


PROJECT CỦA CT C#



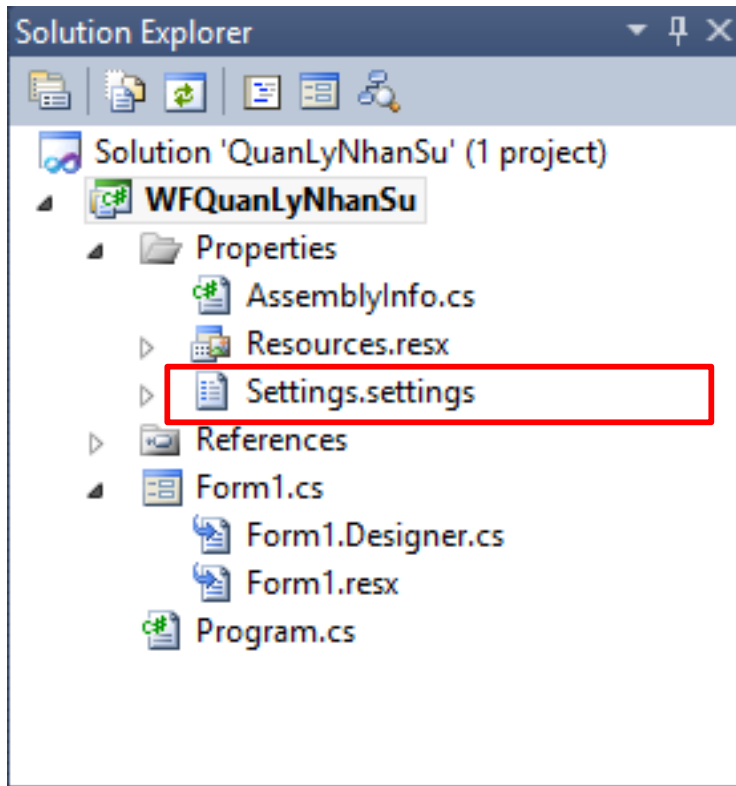
Resources.resx

Dùng để thêm, sửa tài nguyên cho dự án.





PROJECT CỦA CT C#



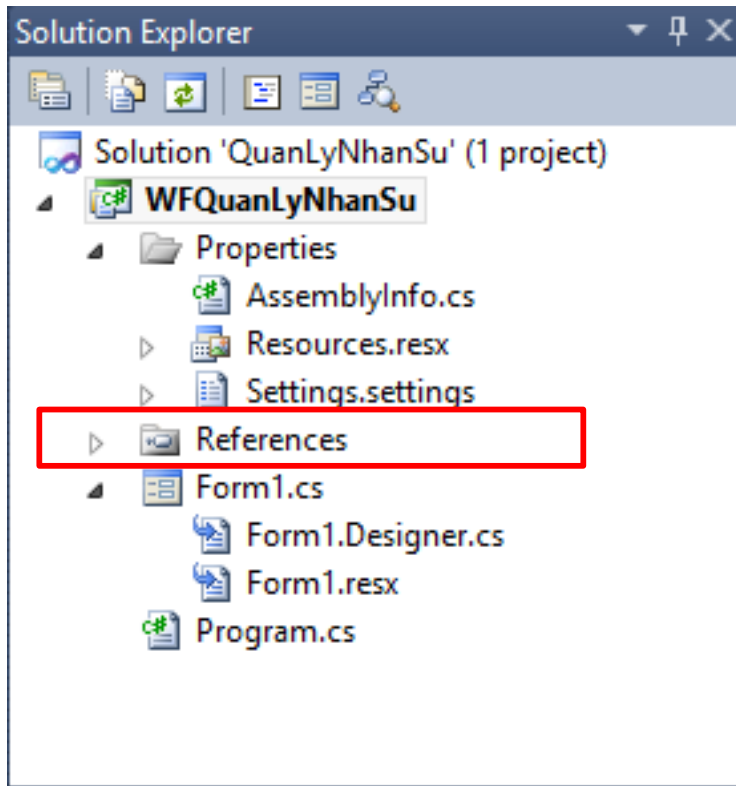
Settings.settings

Đây là nơi lưu trữ các Preference lên ổ cứng, viết ra đĩa và sau đó đọc lại nếu chương trình được chạy lại

Name	Type	Scope	Value
Setting	string	User	

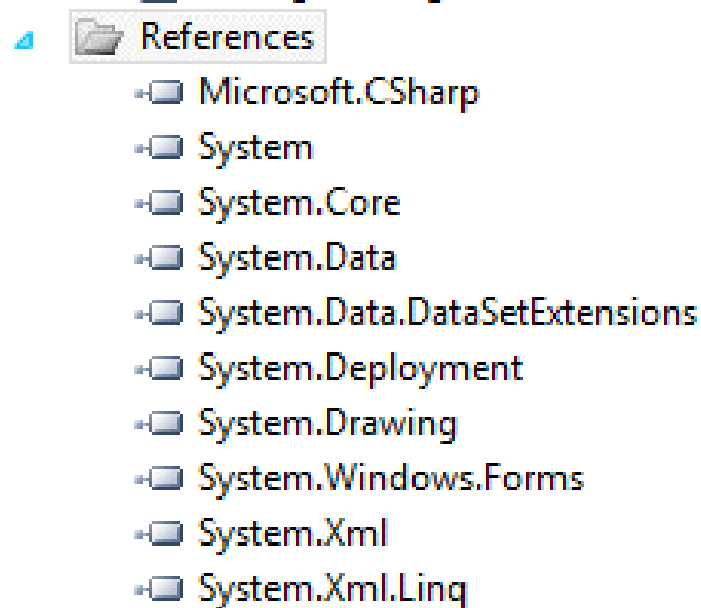


PROJECT CỦA CT C#



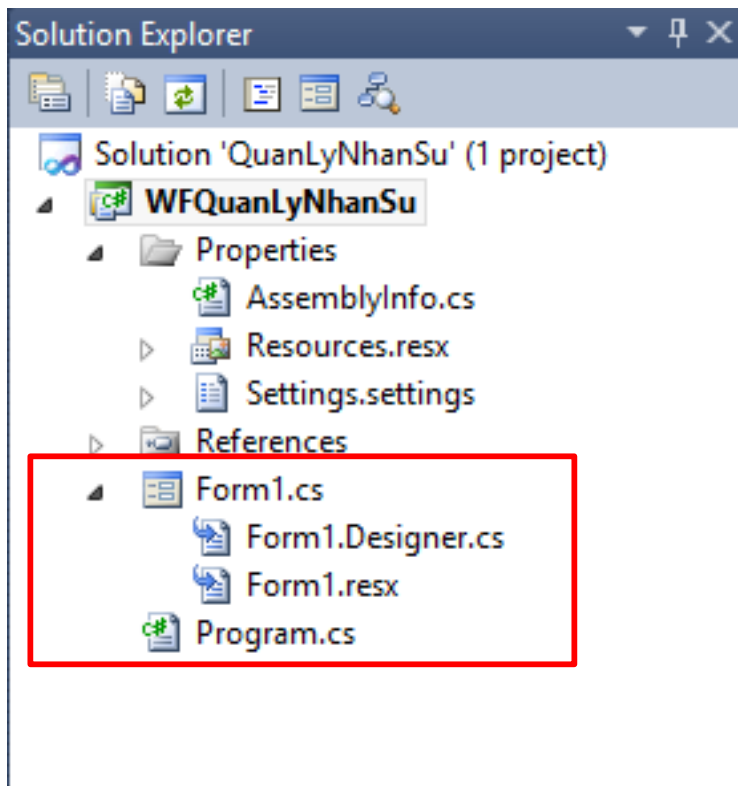
References

Đây là nơi dùng để lưu những thư viện đã sử dụng trong chương trình.



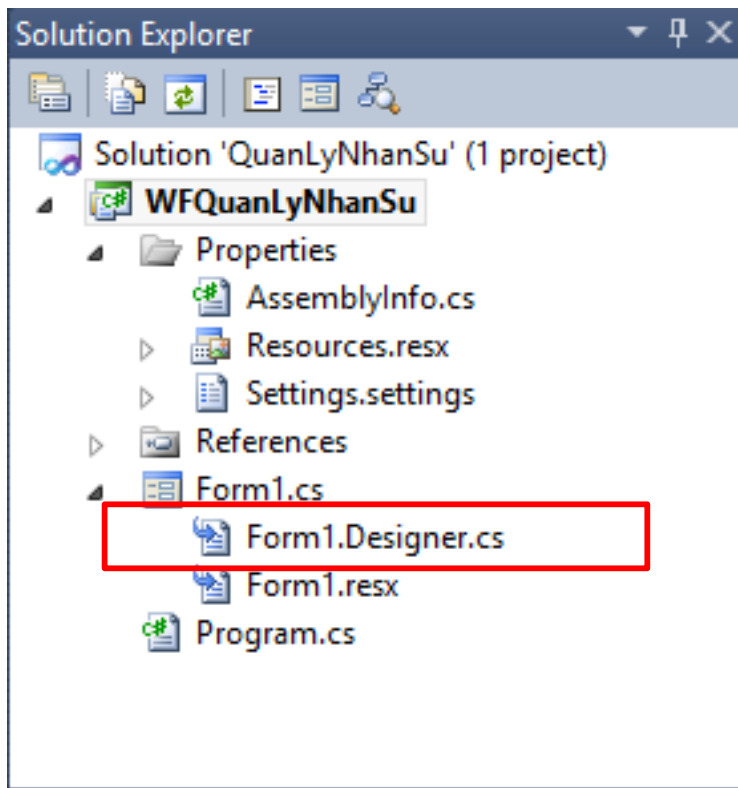


PROJECT CỦA CT C#





PROJECT CỦA CT C#

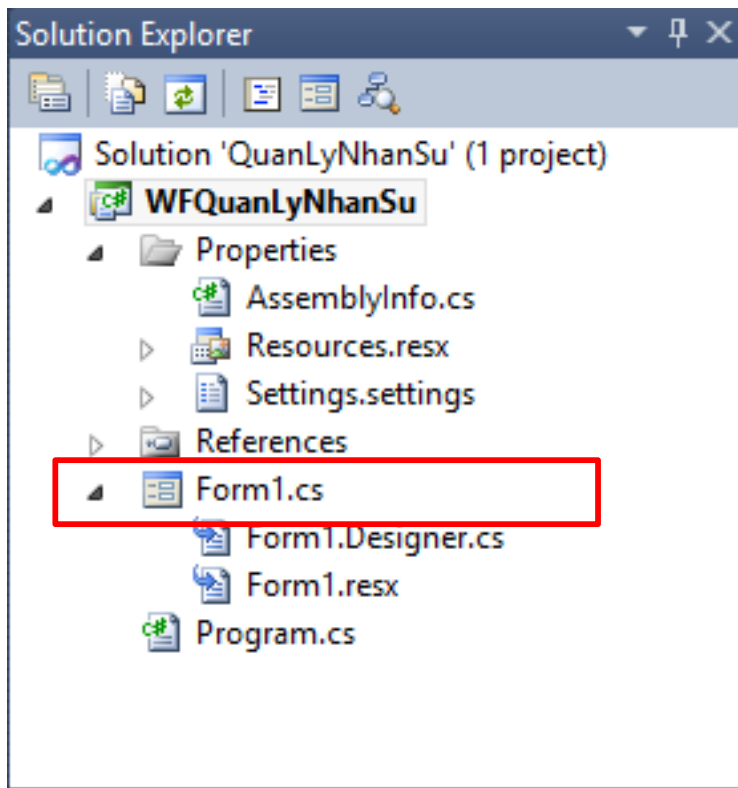


Form1.Designer.cs

Đây là nơi chứa code khởi tạo của các thành phần trên Form khi người lập trình kéo thả hoặc thay đổi các thuộc tính trên GUI



PROJECT CỦA CT C#



Form1.cs

Đây là nơi chứa code thực thi các phương thức, sự kiện trên từng các đối tượng trên Form.



PROJECT CỦA CT C#

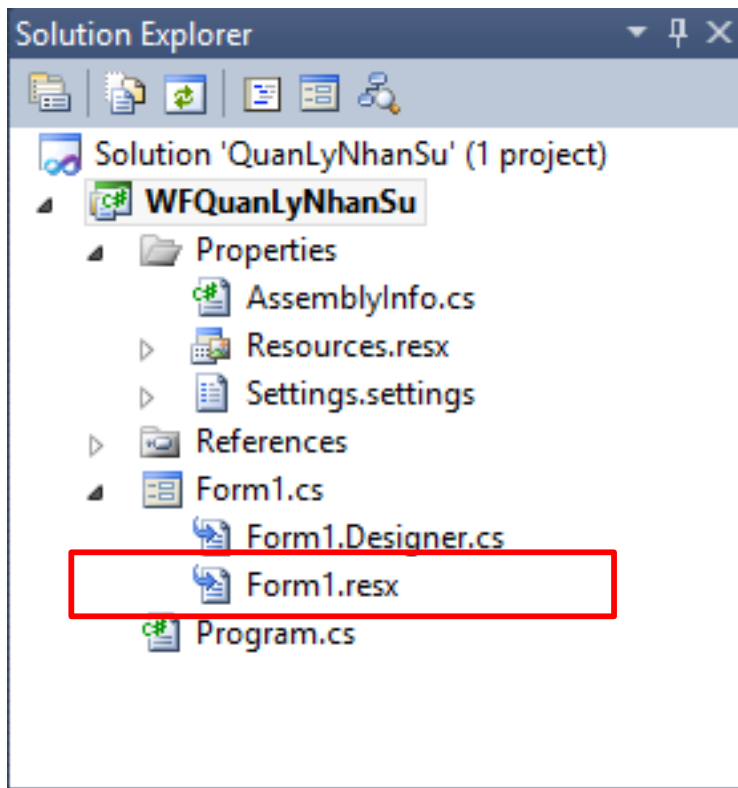
• frm_Main.cs

Frm_Main.Designer.cs
namespace WFQuanLyNhanSu
{
 partial class Frm_Main
 {
 }
}

Frm_Main.cs
namespace WFQuanLyNhanSu
{
 public **partial** class frm_Main: Form
 {
 }
}



PROJECT CỦA CT C#

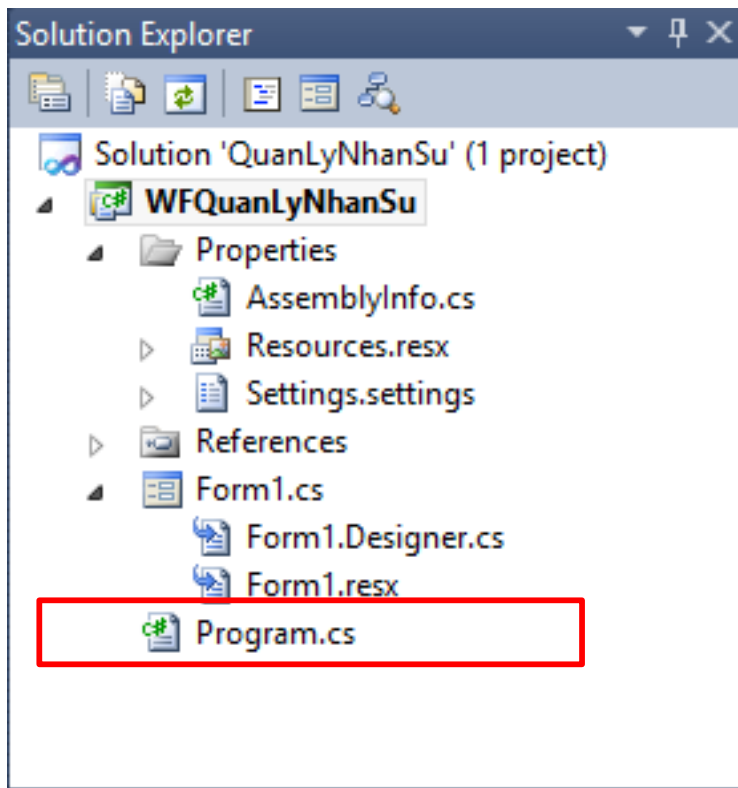


Form1.resx

Đây là nơi chứa tài nguyên của Form



PROJECT CỦA CT C#



Program.cs

Đây là nơi chứa hàm Main, là nơi khởi động chương trình với lệnh chạy truyền vào tham số là một Form bắt đầu

```
static void Main()  
{  
    Application.Run(new Form1());  
}
```



GIỚI THIỆU WINFORM

- Lợi ích của việc sử dụng IDE?

- Tất cả những code được tạo tự động dựa trên sự thao tác thiết kế form của người dùng
- Rút ngắn nhiều thời gian cho việc thao tác giao diện form



**Có được ứng dụng form
mặc dù chưa viết code!**



FORM

- **Form** là cửa sổ chính của ứng dụng giao diện người dùng đồ họa
- Tất cả các thành phần trên form đều là đối tượng

Form1

Tên đăng nhập

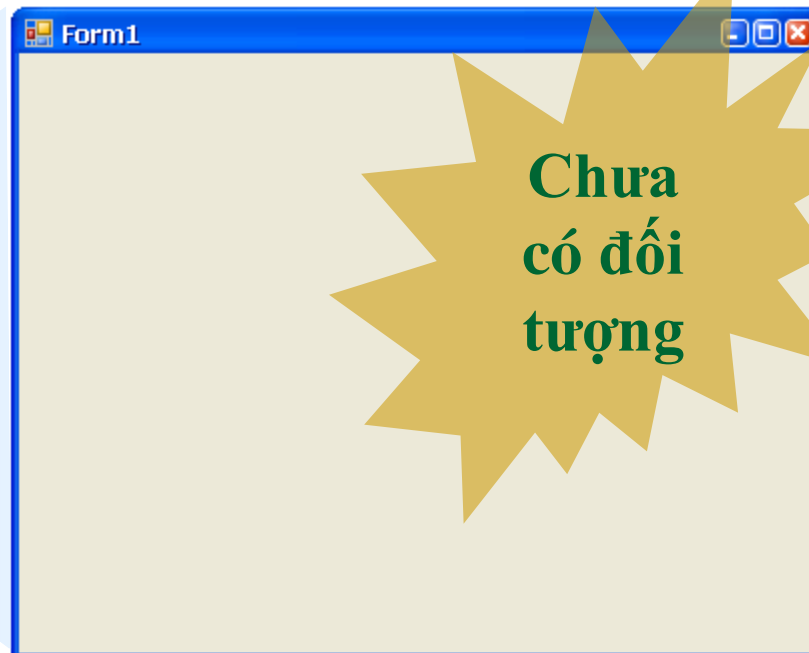
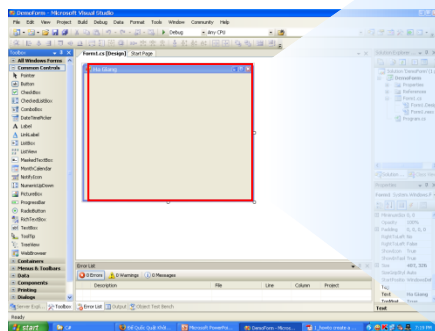
Mật khẩu

Đăng nhập

Thoát



GAO DIỆN THIẾT KẾ FORM



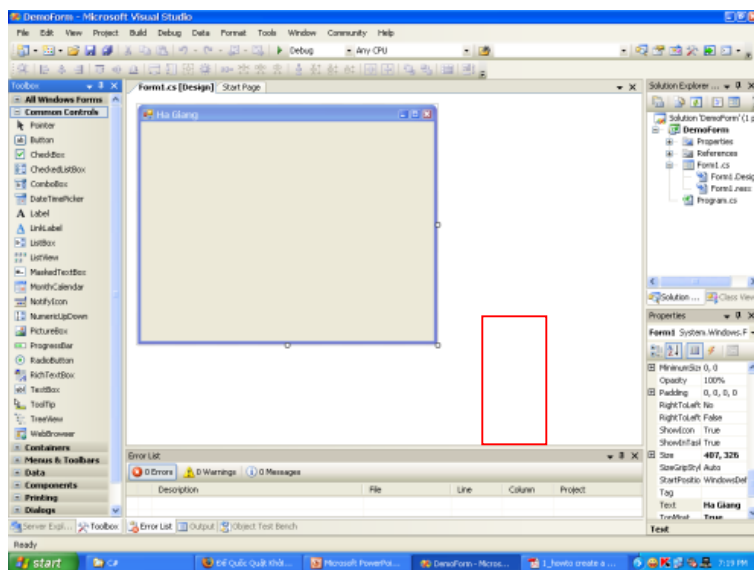
Chưa
có đối
tượng

Form chính của ứng dụng



CỬA SỔ PROPERTIES

Cửa sổ
properties
của form



Properties

Form1 System.Windows.Forms.Form

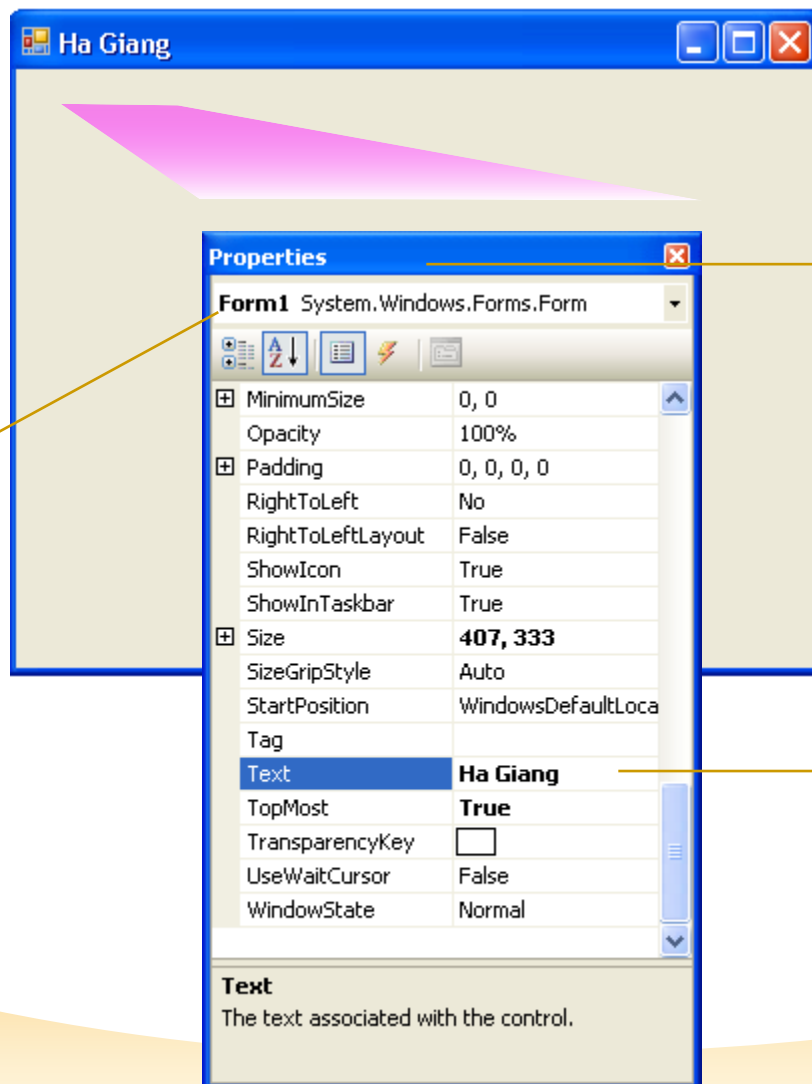
(ApplicationSetting)

(DataBindings)

(Name)	Form1
AcceptButton	(none)
AccessibleDescripti	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
AutoScaleMode	Font
AutoScroll	False
AutoScrollMargin	0, 0
AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
AutoValidate	EnablePreventFocu
BackColor	Control
BackgroundImage	(none)
BackgroundImageL	Tile
CancelButton	(none)
CausesValidation	True
ContextMenuStrip	(none)



CỬA SỔ PROPERTIES



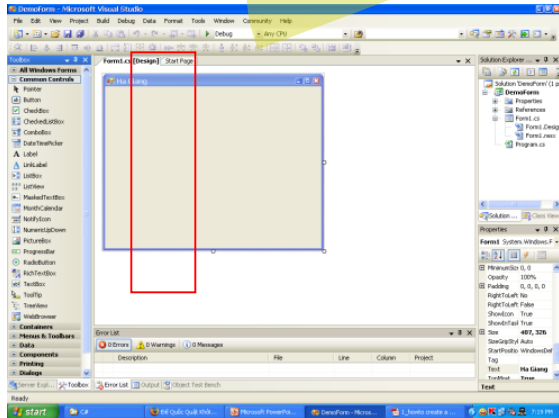
Tên của form
chính là tên
lớp

Để dàng hiệu
chỉnh form
thông qua cửa
sổ Properties

Thay đổi title



THIẾT KẾ GIAO DIỆN



Demo

Number 1

Number 1

Add

A Label

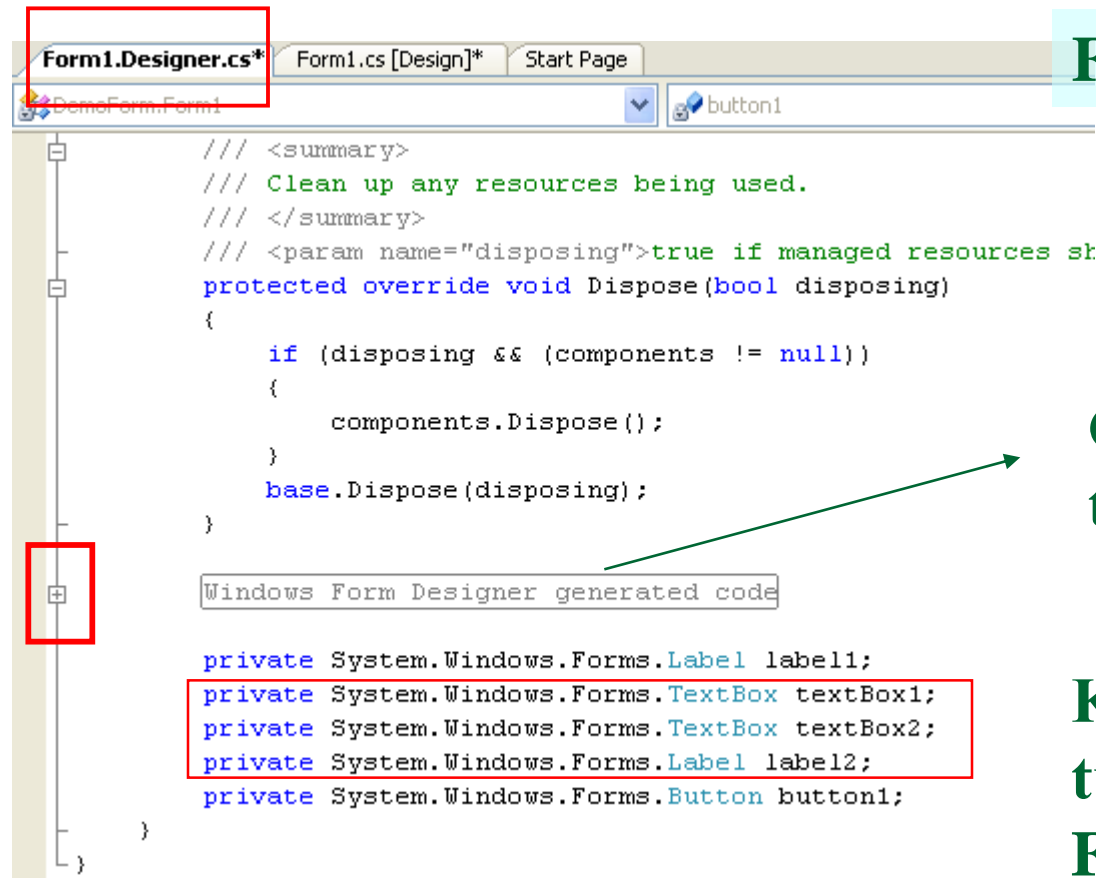
abl TextBox

ab Button



THIẾT KẾ GIAO DIỆN

- Phần code thiết kế Form1 được tạo tự động



Form1.Designer.cs

Chứa code khởi tạo control

Khai báo các đối tượng control trên Form1



THIẾT KẾ GIAO DIỆN

- Phần code thiết kế Form1 được tạo tự động

InitializeComponent

```
Form1.Designer.cs* Form1.cs [Design]* Start Page
DemoForm.Form1 button1

/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.textBox1 = new System.Windows.Forms.TextBox();
    this.textBox2 = new System.Windows.Forms.TextBox();
    this.label2 = new System.Windows.Forms.Label();
    this.button1 = new System.Windows.Forms.Button();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.AutoSize = true;
    this.label1.Location = new System.Drawing.Point(48, 43);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(53, 13);
    this.label1.TabIndex = 0;
    this.label1.Text = "Number 1";
    //
    // textBox1
    //
}
```

Tạo đối tượng

**Lần lượt khai
báo các thuộc
tính cho các
control**



THIẾT KẾ GIAO DIỆN

- Phần code thiết kế Form1 được tạo tự động

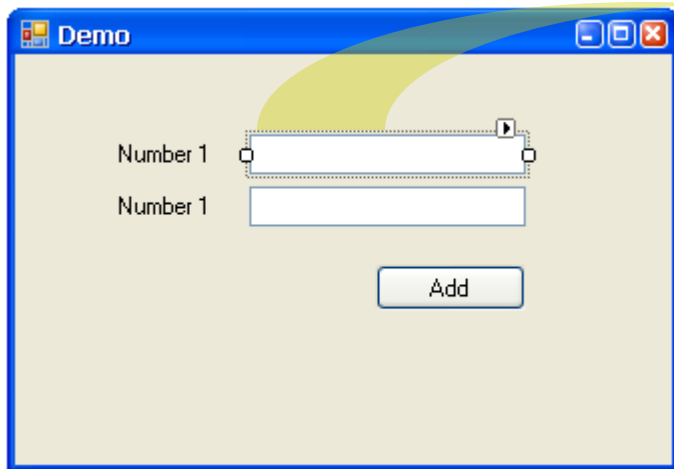
InitializeComponent

```
//  
// Form1  
//  
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);  
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;  
this.ClientSize = new System.Drawing.Size(328, 205);  
this.Controls.Add(this.BtnAdd);  
this.Controls.Add(this.txtNum2);  
this.Controls.Add(this.label2);  
this.Controls.Add(this.txtNum1);  
this.Controls.Add(this.label1);  
this.Name = "Form1";  
this.Text = "Demo";  
this.TopMost = true;  
this.ResumeLayout(false);  
this.PerformLayout();
```

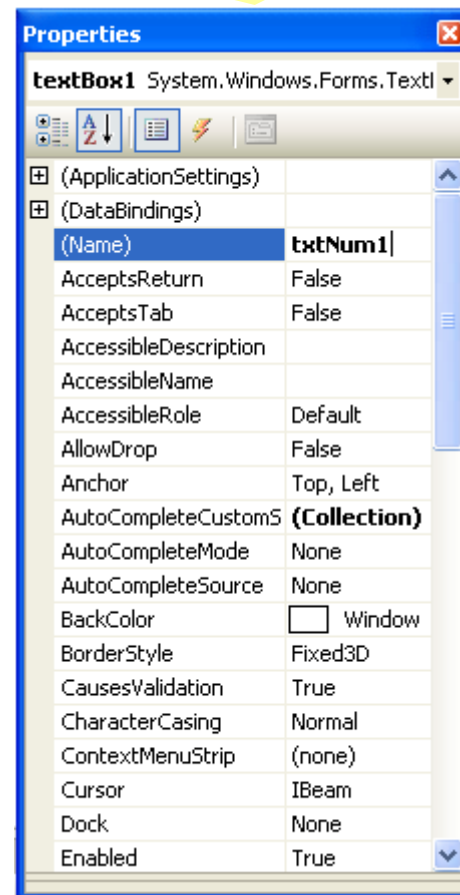
Đưa các control vào
danh sách control
của Form1



THUỘC TÍNH CHO CONTROL



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



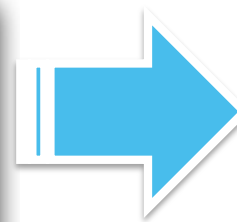
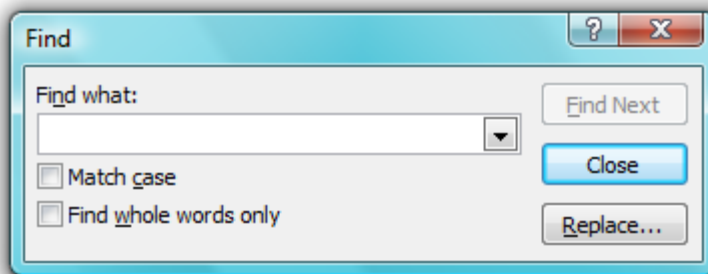
Đổi tên thành txtNum1



Phương thức của lớp Form

❖ Các hành động có thể thực hiện trên form

- Activate: cho form nhận focus
- Close: đóng và giải phóng resource
- Hide: ẩn form
- Refresh: tô vẽ lại
- Show: cho form show ra màn hình (modeless) và activate
- ShowDialog: hiển thị dạng modal



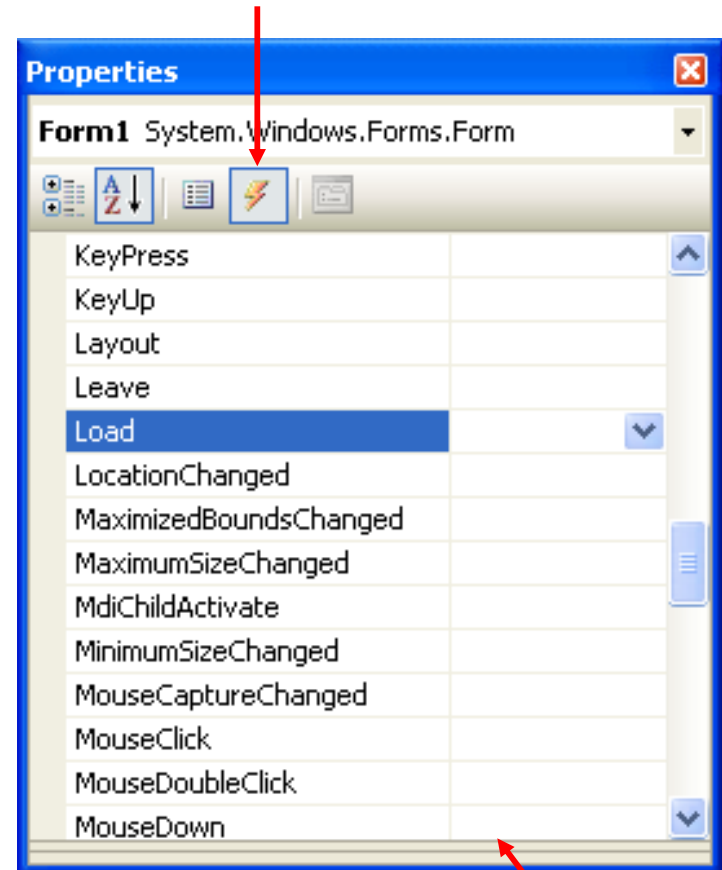
Modeless



Event của Form

❖ Tạo xử lý cho event

- Trong cửa sổ properties
- Chọn biểu tượng event
- Kích đúp vào tên event



Tên event Trình xử lý nếu có



Event của Form

❖ Event thường dùng

- **Load**: phát sinh trước khi form xuất hiện lần đầu tiên
- **Closing**: xuất hiện khi form đang chuẩn bị đóng
- **Closed**: xuất hiện khi form đã đóng
- **Resize**: xuất hiện sau khi user resize form
- **Click**: xuất hiện khi user click lên nền form
- **KeyPress**: xuất hiện khi form có focus và user nhấn phím



BÀI TẬP

- **Tạo một Form Demo như hình vẽ**

- Câu 1: Thực hiện tính toán cho 4 nút $+$, $-$, $*$, $/$, và xuất ra kết quả
- Câu 2: khi đóng ứng dụng thì hiển thị thông báo cảnh báo người dùng

The image shows a Windows application window titled "Form1". The window has a blue background and contains the following elements:

- Three text input fields arranged vertically, labeled "Số A", "Số B", and "Kết quả" from top to bottom.
- Four buttons with mathematical operators: "+", "-", "*", and "/", arranged horizontally below the input fields.
- A button labeled "Thoát" (Exit) in the bottom right corner.



CÁCH THỰC HIỆN

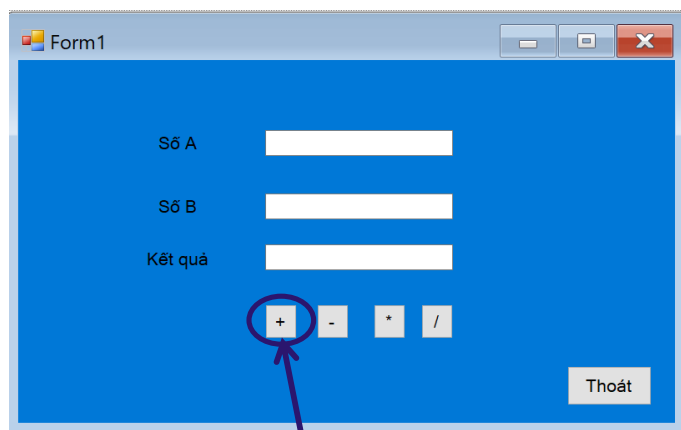
Câu 1: Thực hiện

- Button + cung cấp sự kiện click
- Form sẽ được cảnh báo khi Button + được click
- Form sẽ lấy dữ liệu từ 2 textbox và cộng → kết quả

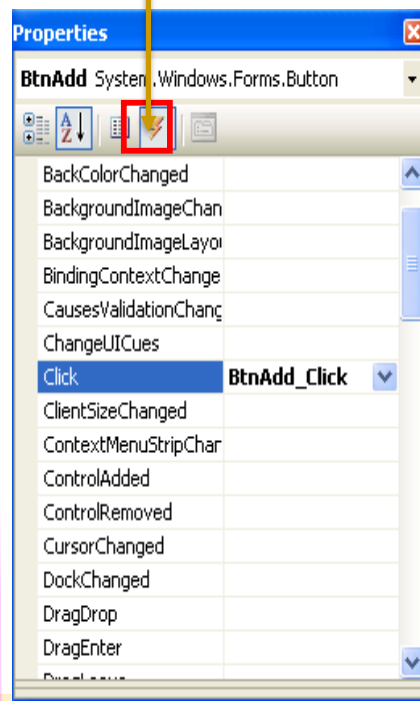


CÁCH THỰC HIỆN

- Đặt tên (Name) theo chuẩn cho các Button và TextBox
- Kích đúp vào Button + trên màn hình thiết kế cho phép tạo event handler cho sự kiện này.



event



Cửa sổ quản lý
event của btnAdd



CÁCH THỰC HIỆN

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

```
private void BtnAdd_Click(object sender, EventArgs e)
{
}
}
```



Phương thức điều khiển sự kiện cho button Add



CÁCH THỰC HIỆN

InitializeComponent

```
//  
// 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);
```

Sự kiện click

Trình xử lý được
gọi khi event xảy
ra

Tham chiếu phương thức (Delegate) chuẩn cho event handler



CÁCH THỰC HIỆN

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

```
    int SoA, SoB, SoC; → Tạo các biến Int
```

```
    public Form1()
```

```
    {
```

```
        InitializeComponent();
```

```
    }
```

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

```
        SoA = int.Parse(txtA.Text); Lấy giá trị Int từ txtA cho SoA
```

```
        SoB = int.Parse(txtB.Text); Lấy giá trị Int từ txtB cho SoB
```

```
        SoC = SoB + SoA; Số C = Số B + Số A
```

```
        txtC.Text = SoC.ToString(); Chuyển số C về kiểu chuỗi  
và ép ngược lên lại txtC
```



CÁCH THỰC HIỆN

- **Hiện bảng thông báo khi đóng ứng dụng**
 - ❑ **Kích đúp vào item FormClosing trong cửa sổ event**
 - ❑ **Hàm Form1_FormClosing được tạo và gắn với sự kiện FormClosing**
 - ❑ **Viết code cho event handler Form1_FormClosing**

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    DialogResult r;
    r = MessageBox.Show("Bạn thật sự muốn th  

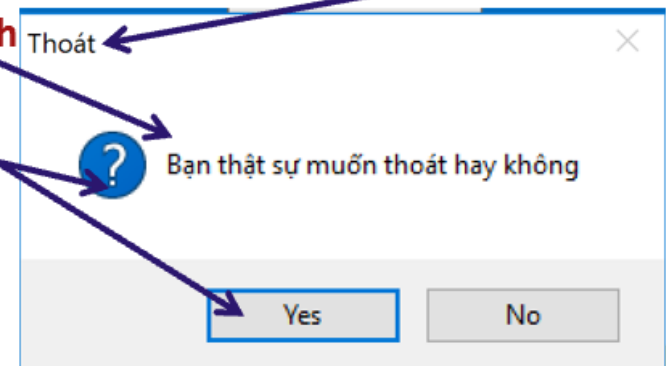
    MessageBoxButtons.YesNo,  

    MessageBoxIcon.Question);  

    if (r == DialogResult.No)  

        e.Cancel = true;  

}
```





Tài liệu tham khảo

- [1] Giáo trình lập trình Winform với C#.NET Lê Trung Hiếu, Nguyễn Thị Minh Thi
- [2] Giáo trình lập trình C#.net Phạm Hữu Khang
- [3] C# Language Reference, Anders Hejlsberg and Scott Wiltamuth, Microsoft Corp.
- [4] Professional C#, 2nd Edition, Wrox Press Ltd.
- [5] Web site www.Codeproject.com
- [6] Web site www.CodeGuru.com