



LẬP TRÌNH WINFORM VỚI C#

PHAN TRỌNG TIỀN
BM Công nghệ phần mềm
Khoa Công nghệ thông tin, VNUA
Email: phantien84@gmail.com
Website: <http://timoday.edu.vn>



Giới thiệu chung

- ☐ Sử dụng IDE (Integrated Development Environment) trong VS để phát triển nhanh ứng dụng
- ☐ Sử dụng các thư viện của Windows Forms để phát triển các ứng dụng desktop và client server chạy trên môi trường Windows
- ☐ Thiết kế giao diện sử dụng IDE và viết code C#
- ☐ Các tiêu chuẩn của một ứng dụng dạng Windows Forms
- ☐ Tìm hiểu các control và các thuộc tính thông dụng của các control



Nội dung chính

1. Tổng quan
2. Tại sao sử dụng Windows Forms
3. Cấu trúc Windows Forms
4. Sử dụng Windows Forms
5. Demo: Thực hiện Windows Forms
6. Sử dụng các Control
7. Demo: Thực thi chức năng kéo và thả
8. Thừa kế của Windows Forms
9. Demo: Sử dụng thừa kế Windows Forms

7/5/16

Lập trình WinForm với C#

3



Mục đích của chương

- ☐ Chương này cung cấp sinh viên kiến thức cần thiết để tạo các ứng dụng Winform.
- ☐ Sau bài này sinh viên có thể:
 - ☐ Mô tả các lợi ích của Windows Forms
 - ☐ Cách sử dụng các thuộc tính, phương thức và các sự kiện của lớp Form và các control cơ bản
 - ☐ Sử dụng các thuộc tính mới và các phương thức của Windows Forms
 - ☐ Viết code cho các sự kiện điều khiển
 - ☐ Sử dụng các control mới và các control cải tiến
 - ☐ Thêm và chỉnh sửa menu
 - ☐ Tạo một form được thừa kế từ form khác

7/5/16

Lập trình WinForm với C#

4



1. Tổng quan

- ☐ Bạn sẽ nắm được các đặc điểm có sẵn trong Windows Forms, cách thay đổi các form và control, và các thuộc tính, các phương thức, các sự kiện.
- ☐ Bạn sẽ học cách tạo một vài dạng hộp thoại chuẩn của Windows.
- ☐ Nắm được tính thừa kế, cho phép bạn sử dụng kỹ thuật lập trình hướng đối tượng vào các form của bạn.

7/5/16

Lập trình WinForm với C#

5



2. Tại sao sử dụng Windows Forms

- | | |
|---|--|
| ☐ Có tập hợp các control phong phú | ☐ Hỗ trợ khả năng truy cập qua các thuộc tính của control |
| ☐ Nhiều kiểu giao diện | ☐ Hỗ trợ thừa kế |
| ☐ Hỗ trợ cải tiến việc in ấn | ☐ Các đối tượng có thể cải tiến |
| ☐ Hỗ trợ cải tiến về đồ họa – GDI+ | ☐ Thuận lợi cho thiết kế forms |

7/5/16

Lập trình WinForm với C#

6



3. Cấu trúc của Windows Forms

- ❑ Phân cấp các lớp của Windows Forms
- ❑ Sử dụng lớp `Windows.Forms.Application`
- ❑ Nghiên cứu Code Behind của Windows Forms

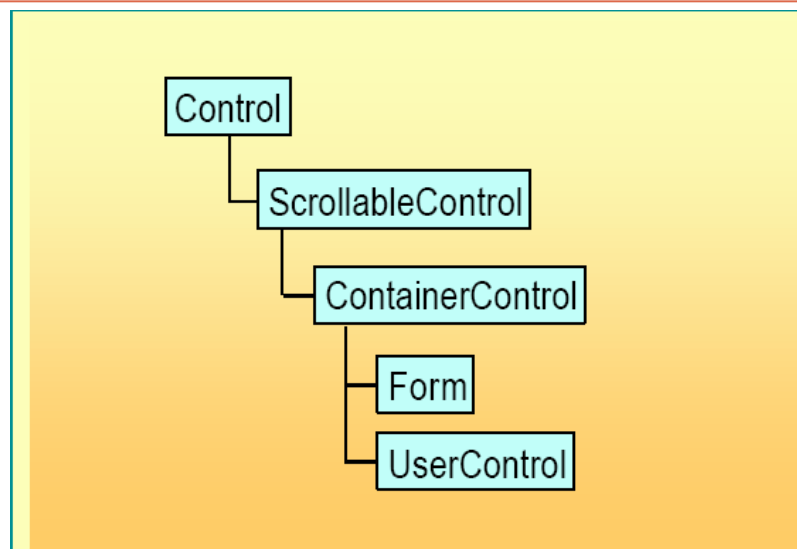
7/5/16

Lập trình WinForm với C#

7



Phân cấp các lớp của Windows Forms



7/5/16

Lập trình WinForm với C#

8



Sử dụng lớp `Windows.Forms.Application`

- ❑ Bắt đầu và kết thúc ứng dụng


```
static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new Form1());
}
```
- ❑ Thiết lập thông tin và truy vấn thông tin ứng dụng


```
string strAppPath = Application.StartupPath;
```

lấy đường dẫn chứa nơi cài file chạy

7/5/16

Lập trình WinForm với C#

9



Nghiên cứu Code Behind của Windows Forms

- ❑ Imports
 - ❑ Truy cập các chức năng trong namespace tham chiếu trong assemblies


```
using System.Windows.Forms
```
- ❑ Class
 - ❑ Thừa kế từ `System.Windows.Forms.Form`
 - ❑ Constructor – `public Form1()`
 - ❑ Initializer – `void InitializeComponent()`
 - ❑ Destructor – `void Dispose()`

7/5/16

Lập trình WinForm với C#

10



4. Sử dụng Windows Forms

- ☐ Sử dụng Form Properties
- ☐ Sử dụng Form Methods
- ☐ Sử dụng Form Events
- ☐ Điều khiển sự kiện
- ☐ Tạo Form MDI
- ☐ Sử dụng các dạng hộp thoại chuẩn

7/5/16

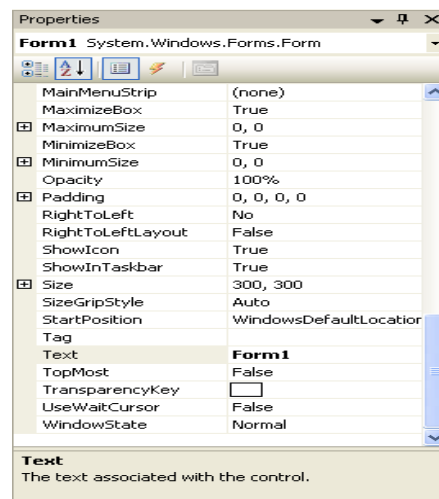
Lập trình WinForm với C#

11



Sử dụng Form Properties

- ☐ Kích trên Form hoặc Control sẽ có hộp thoại Form Properties tương ứng



7/5/16

Lập trình WinForm với C#

12



Sử dụng Form Properties

- ☐ DialogResult
- ☐ Font
- ☐ Opacity
- ☐ MaximumSize và MinimumSize
- ☐ TopMost
- ☐ AcceptButton và CancelButton

7/5/16

Lập trình WinForm với C#

13



Sử dụng Form Methods

- ☐ CenterToScreen và CenterToParent
- ☐ Close
- ☐ Show và ShowDialog

7/5/16

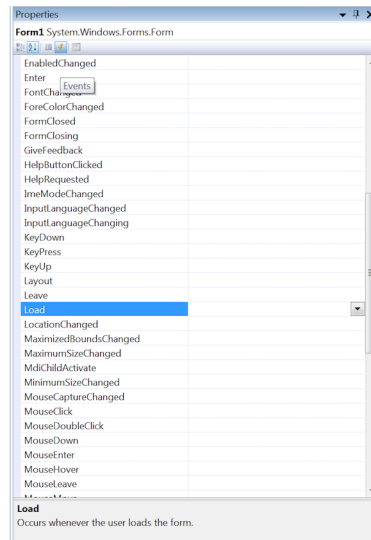
Lập trình WinForm với C#

14



Sử dụng Form Events

- ❑ Trên cửa sổ **Properties**
 - > Chọn biểu tượng **Events**



7/5/16

Lập trình WinForm với C#

15



Sử dụng Form Events

- ❑ Activated và DeActivate
 - ❑ Activated là xảy ra khi Form được kích hoạt hoặc người dùng tương tác
 - ❑ DeActive là xảy ra khi Form mất focus.
- ❑ FormClosing
 - ❑ Xảy ra khi Form đang chuẩn bị đóng.
- ❑ FormClosed
 - ❑ Xảy ra sau sự kiện Closing và trước Dispose
- ❑ MenuStart và MenuComplete
 - ❑ Xảy ra khi menu nhận và mất focus.

7/5/16

Lập trình WinForm với C#

16



Điều khiển sự kiện

❑ Điều khiển nhiều sự kiện với một thủ tục

```
private void Form1_Activated(object sender, EventArgs e)
{
    Debug.WriteLine("Form1 Activated!");
    this.Text = "Form Activated!";
}
```

❑ Giải thích:

- ❑ “sender”: tham chiếu tới đối tượng mà được gọi bởi sự kiện. Nó hữu ích trong trường hợp có nhiều đối tượng sử dụng chung một sự kiện
- ❑ “e”: chứa các thông tin về control mà người dùng có thể nhận diện được các hành vi và cách thực của control
- ❑ Tùy vào mỗi sự kiện khác nhau mà tham số thứ 2 có thể khác nhau

7/5/16

Lập trình WinForm với C#

17



Demo dùng sự kiện

❑ Tạo một ứng dụng dạng WinForms hiển thị các thông tin trong cửa sổ

❑ Code cho các sự kiện

- ❑ **Form1_Activated** Debug.WriteLine("Activated")
- ❑ **Form1_Closed** Debug.WriteLine("Closing")
- ❑ **Form1_Deactivate** Debug.WriteLine("Deactivated")
- ❑ **Form1_SizeChanged** Debug.WriteLine("Size changed")

7/5/16

Lập trình WinForm với C#

18



Tạo Form MDI

- ❑ Tạo Form cha
 - ❑ Bạn có thể đặt thuộc tính *IsMdiContainer*
 - ❑ Hoặc code ở sự kiện *Form_Load*

```
this.IsMdiContainer = True
this.WindowState = FormWindowState.Maximized
```
- ❑ Tạo các Form con


```
frmChild frm = new frmChild();
frm.MdiParent = this;
frm.Show();
```
- ❑ Truy cập các Form con
- ❑ Sắp xếp các Form con

7/5/16

Lập trình WinForm với C#

19



Sử dụng các dạng hộp thoại chuẩn

- ❑ Lớp *MessageBox*

```
if(MessageBox.Show("Bạn có muốn tiếp tục?", "Thông
bao", MessageBoxButtons.OKCancel) == DialogResult.OK)
{
    //Bạn viết code ở đây
}
```

7/5/16

Lập trình WinForm với C#

20



5. Demo: Sử dụng Windows Forms

- ☐ Bạn sẽ học cách dùng *Properties* và *Methods* của Window Form, bao gồm tạo cả Form riêng, độ trong suốt của Form, và thanh cuộn tự động cho Form.



6. Sử dụng các control

- ☐ Các control căn bản
- ☐ Các control mới
- ☐ Sử dụng Properties của Control
- ☐ Sử dụng Methods của Control
- ☐ Tạo Menus
- ☐ Cung cấp Help cho người sử dụng
- ☐ Thực hiện chức năng Kéo – Thả



Các control căn bản

- ☐ Label: hiển thị icons hoặc text không sửa đc
- ☐ Textbox: nhập dữ liệu từ bàn phím hoặc hiển thị
- ☐ Button: bắt các sự kiện khi kích chuột
- ☐ CheckBox: cho người dùng chọn hoặc không chọn
- ☐ ComboBox: một danh sách đồ xuống
- ☐ ListBox: một danh sách các item để chọn
- ☐ Panel: cho phép bố trí các control khác trên nó
- ☐ PictureBox: hiển thị ảnh

7/5/16

Lập trình WinForm với C#

23



Các thuộc tính và sự kiện thông dụng

Control	Mô tả
Label	
Thuộc tính: Font	Thay đổi Font chữ trên Label
Thuộc tính: Text	Text hiển thị trên Label
Thuộc tính: TextAlign	Căn chỉnh vị trí của Text (left, center, right, bottom, middle, top)
Thuộc tính: Name	Tên dùng để viết code
TextBox	
Thuộc tính: Font	Thay đổi Font trên control
Thuộc tính: Text	Hiển thị text trên control
Thuộc tính: Multiline	Nếu là true thì textbox nhập nhiều dòng
Thuộc tính: ReadOnly	Nếu là true thì textbox không thể nhập
Sự kiện: Keypress	Xảy ra khi bàn phím được ấn
Sự kiện: TextChanged	Khi dữ liệu text được thay đổi

7/5/16

Lập trình WinForm với C#

24



Các thuộc tính và sự kiện thông dụng

Control	Mô tả
Button	
Thuộc tính: Text	Text được hiển thị trên control
Sự kiện: Click	Xảy ra khi người dùng click vào nút
CheckBox	
Thuộc tính: Checked	Chỉ định control đã được chọn
Thuộc tính: Text	Text được hiển thị trên control
Sự kiện: CheckChanged	Xảy ra khi checkbox được chọn hoặc không chọn
ComboBox và ListBox	
Thuộc tính: Items	Tập các item của control
Sự kiện: SelectedIndexChanged	Xảy ra khi một item của control được chọn

7/5/16

Lập trình WinForm với C#

25



Các thuộc tính và sự kiện thông dụng

Control	Mô tả
Panel	
Thuộc tính: AutoScroll	Chỉ định khi Panel không đủ lớn chứa các control khác thì sẽ xuất hiện thanh cuộn
Thuộc tính: BorderStyle	Thiết lập đường viền của Panel
PictureBox	
Thuộc tính: Image	Thiết lập ảnh hiển thị trên PictureBox
Thuộc tính: SizeMode	Kích thước và vị trí ảnh so với control Các giá trị: Normal, StretchImage, AutoSize, CenterImage
Sự kiện: Click	Xảy ra khi picturebox được kích

7/5/16

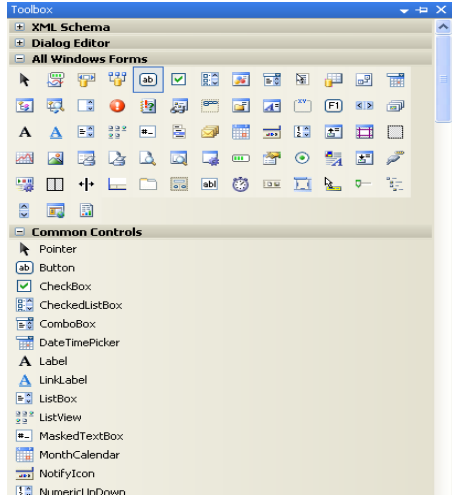
Lập trình WinForm với C#

26



Các control mới

- ☐ CheckedListBox
- ☐ LinkLabel
- ☐ Splitter
- ☐ ToolTip
- ☐ NotifyIcon



7/5/16

Lập trình WinForm với C#

27



Sử dụng Properties của Control

- ☐ Đặt vị trí Control
 - ☐ Anchor
 - ☐ Location
- ☐ Thuộc tính Text

`Button1.Text = "Click Me"`

7/5/16

Lập trình WinForm với C#

28



Sử dụng Methods của Control

☐ BringToFront và SendToBack

Button1.BringToFront()

Button2.SendToBack()

☐ Focus

TextBox1.Focus()

TextBox1.SelectAll()

7/5/16

Lập trình WinForm với C#

29



Tạo Menus

☐ Các lớp Menu

☐ Tạo Menu lúc thiết kế

☐ Sử dụng Menu Designer

☐ Tạo Menu lúc chạy

```
private System.Windows.Forms.MenuStrip menuStrip1;
private System.Windows.Forms.ToolStripMenuItem mnuArrange;
private System.Windows.Forms.ToolStripMenuItem mnuArrangeCasscade;
this.menuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[]
{this.mnuArrange});
this.mnuArrange.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] { this.mnuArrangeCasscade});
this.Controls.Add(this.menuStrip1);
this.MainMenuStrip = this.menuStrip1;
```

7/5/16

Lập trình WinForm với C#

30



Cung cấp Help cho người sử dụng

❑ Control **ErrorProvider**

- ❑ Icon lỗi sẽ xuất hiện ở control kế tiếp, và message xuất hiện giống như ToolTip khi chuột di chuyển qua Icon.
- ❑ Được sử dụng kiểm tra dữ liệu đầu vào.

❑ Control **HelpProvider**

- ❑ Gắn các file trợ giúp dạng *.chm, *.hlp, *.html
- ❑ Control cung cấp thông tin trợ giúp qua thuộc tính **HelpString** hoặc **HelpTopic**

7/5/16

Lập trình WinForm với C#

31



Demo: Sử dụng Controls

- ❑ Cách bố trí các thuộc tính của control Button.
- ❑ Thực thi các sự kiện cho control.
- ❑ Cuối cùng, cung cấp các trợ giúp người sử dụng qua control **HelpProvider** và **ToolTip** và cách lập trình tạo context menu.

7/5/16

Lập trình WinForm với C#

32



Thực thi chức năng Kéo – Thả

- ❑ Xử lý đầu tiên
 - ❑ Dùng sự kiện **MouseDown** và phương thức **DoDragDrop** trong sự kiện của *control kéo*.
 - ❑ Tiếp tục dùng sự kiện **DragOver** của control thả, thiết lập thuộc tính **Effect** của tham số **DragEventArgs** trong sự kiện.
 - ❑ Sự kiện cuối cùng **DragDrop** thực hiện khi thả dữ liệu
 - ❑ Dùng phương thức **Data.GetData** kết hợp tham số **DataFormats** để lấy được dữ liệu kéo tới

7/5/16

Lập trình WinForm với C#

33



Demo: Thực thi chức năng Kéo – Thả

- ❑ Thực thi chức năng kéo thả qua một ví dụ đơn giản.

7/5/16

Lập trình WinForm với C#

34



8. Thừa kế của Windows Forms

- ☐ Tái tạo thừa kế từ một Form
- ☐ Tạo một Form cơ sở (Form Base)
- ☐ Tạo một Form được thừa kế
- ☐ Thay đổi Form Base

7/5/16

Lập trình WinForm với C#

35



Tại sao thừa kế từ một Form

- ☐ Một Form là một class, vì vậy nó có thể dùng thừa kế
- ☐ Các ứng dụng sẽ có một giao diện và hoạt động theo chuẩn.
- ☐ Các thay đổi ở form cơ sở sẽ tác động tới các form được thừa kế.
- ☐ Các ví dụ:
 - ☐ Các form Wizard
 - ☐ Các form Logon

7/5/16

Lập trình WinForm với C#

36



Tạo một Form cơ sở (Form Base)

- ☐ Lên kế hoạch Form Base một cách cẩn thận
- ☐ Tạo Form Base như form thông thường
- ☐ Thiết lập các thuộc tính truy cập cho các control
 - ☐ Private – Control chỉ truy cập trong form Base
 - ☐ Protected – control chỉ được truy cập trong các form kế thừa
 - ☐ Public – Control được truy cập trong bất kỳ module.
- ☐ Thêm từ khóa **Overridable** tới các Method một cách thích hợp
- ☐ Build Solution cho Form Base

7/5/16

Lập trình WinForm với C#

37



Tạo form được thừa kế

- ☐ Đảm bảo form base đã được hoàn thành
- ☐ Tham chiếu tới Assembly
- ☐ Tạo form mới được thừa kế
- ☐ Thay đổi thuộc tính khi cần thiết
- ☐ Viết chồng các Method hoặc Event khi có yêu cầu

7/5/16

Lập trình WinForm với C#

38



Thay đổi Form Base

- ☐ Thay đổi Form Base
 - ☐ Các thay đổi sẽ tác động tới các form kế thừa khi Rebuilt
- ☐ Kiểm tra các form được kế thừa
 - ☐ Kiểm tra các thay đổi trước khi rebuilt lại ứng dụng
 - ☐ Kiểm tra lại sau khi rebuilt lại ứng dụng

7/5/16

Lập trình WinForm với C#

39



9. Demo: Sử dụng thừa kế Windows Forms

- ☐ Tạo một lớp Base phục vụ cho mục đích thừa kế
- ☐ Viết chồng các Property, Method của các control form base
- ☐ Thay đổi form base sau khi nó đã được thừa kế.

7/5/16

Lập trình WinForm với C#

40



Tổng kết

- ☐ Các lợi ích Windows Forms?
- ☐ Lớp **ContainerControl** là lớp cơ sở cho các control khác đúng hay sai?
- ☐ Viết code để truy cập tới đường dẫn file chạy ứng dụng
- ☐ Viết code để gọi btnOK khi người sử dụng ấn phím Enter.
- ☐ Liệt kê các control cung cấp trợ giúp cho người sử dụng
- ☐ Viết code để tạo một menu **Help** với một menu con **About** lúc chạy chương trình

7/5/16

Lập trình WinForm với C#

41



Tài liệu tham khảo

- ☐ Windows Forms Programming With C# - WIN0095 – Aptech Worldwide
- ☐ [https://msdn.microsoft.com/en-gb/library/aa288436\(v=vs.71\).aspx](https://msdn.microsoft.com/en-gb/library/aa288436(v=vs.71).aspx)
- ☐ http://www.worldbestlearningcenter.com/index_files/csharp_tutorial_lesson.htm

7/5/16

Lập trình WinForm với C#

42