



| Bài giảng môn học

Bài 4: Windows Form

Lương Trần Hy Hiển

FIT, HCMUP

Lập trình Windows Form với C#

Tổng quan controls

- **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**

Tổng quan controls

■ 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 controls

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 lè text trên control
Visible	Xác định hiển thị control

Form Properties

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

Form Properties

Tên thuộc tính	Ý nghĩa
AutoSize	Cho phép kéo dãn 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

Form Method

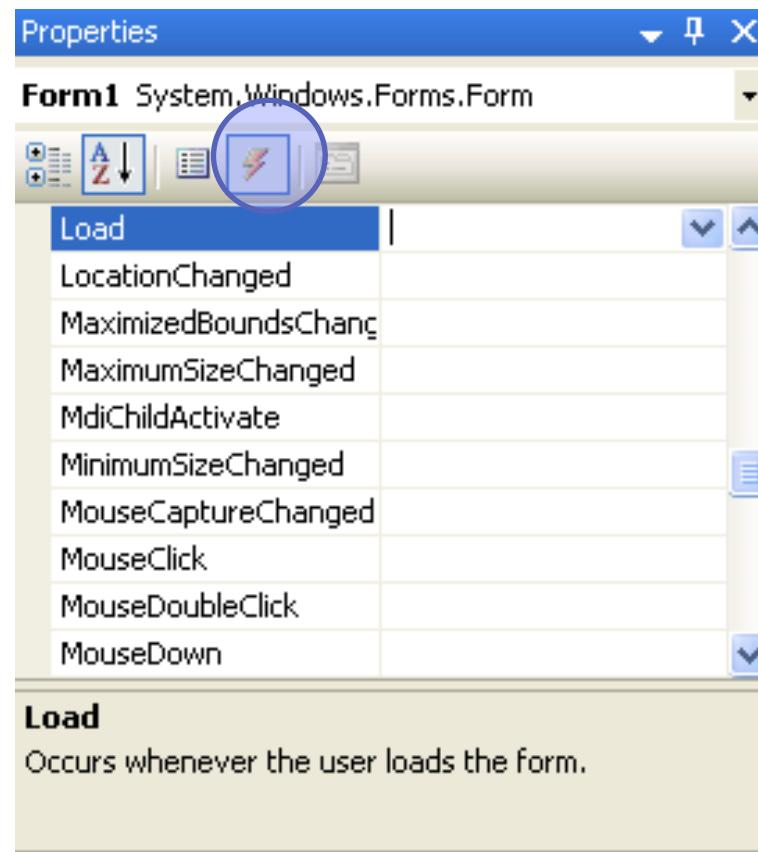
Tên hàm	Ý nghĩa
Show()	Sau khi dialog hiện lên, người dùng có thể chọn focus vào control khác
ShowDialog()	Sau khi dialog hiện lên người dùng không thể focus vào các control khác. Kiểu dữ liệu trả về: DialogResult.
Close()	Đóng Dialog
this.DialogResult = DialogResult.OK	Đóng dialog và trả về kết quả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

Form Event

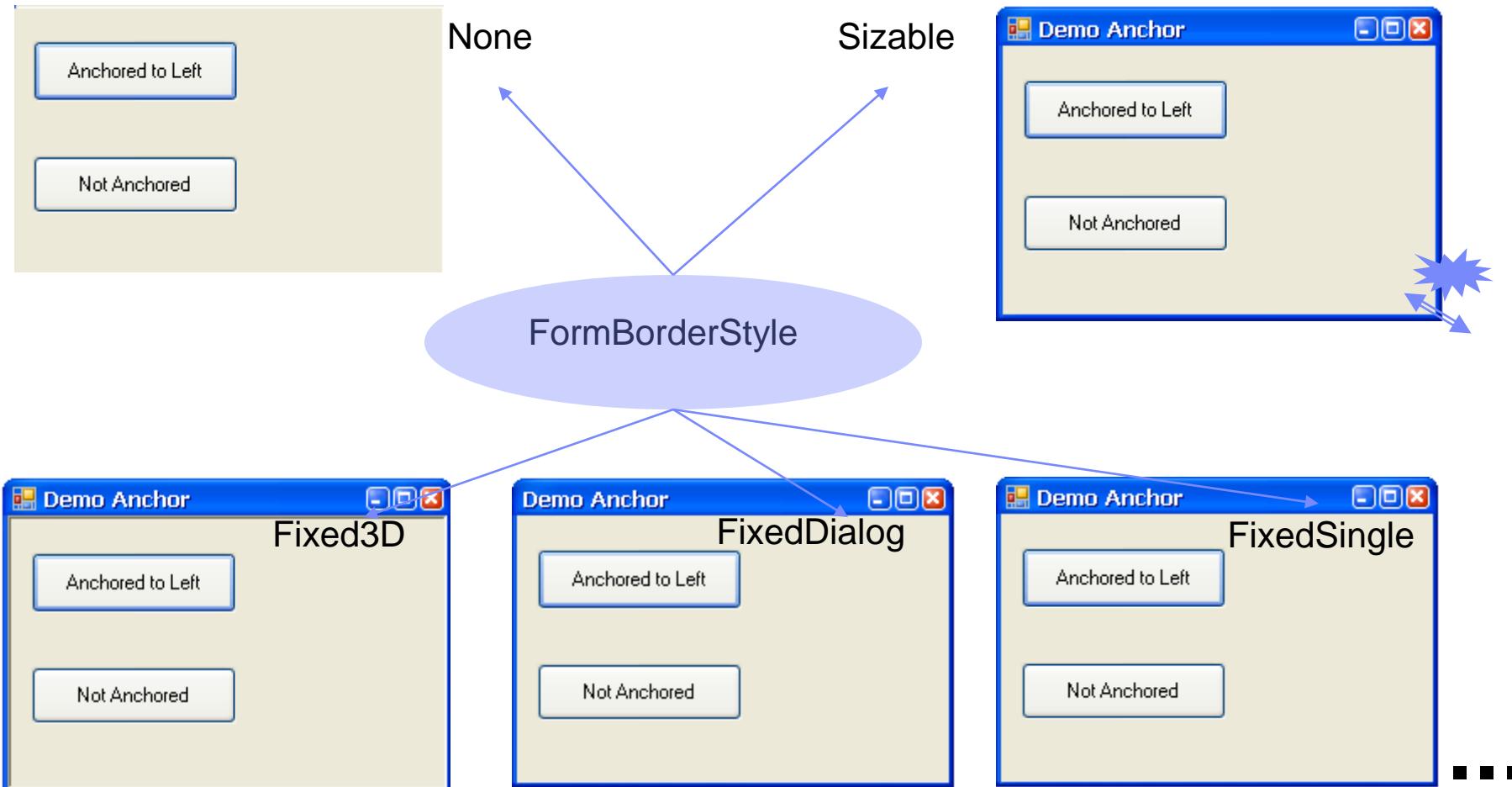
Tên hàm	Ý nghĩa
Load()	Sự kiện được kích hoạt khi Form đã load xong các control. Sự kiện này được bắt khi người dùng muốn khởi tạo biến hoặc giá trị trong Form
Paint()	Sự kiện này được gọi khi Form thực hiện vẽ lại. (Di chuyển chuột, phóng to, thu nhỏ,...) <i>(See trình bày chi tiết trong GDI +)</i>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

Form Event

- Click Click
- DoubleClick
- KeyDown
- MouseHover
- Paint
- Resize
-



Control Layout - Anchor

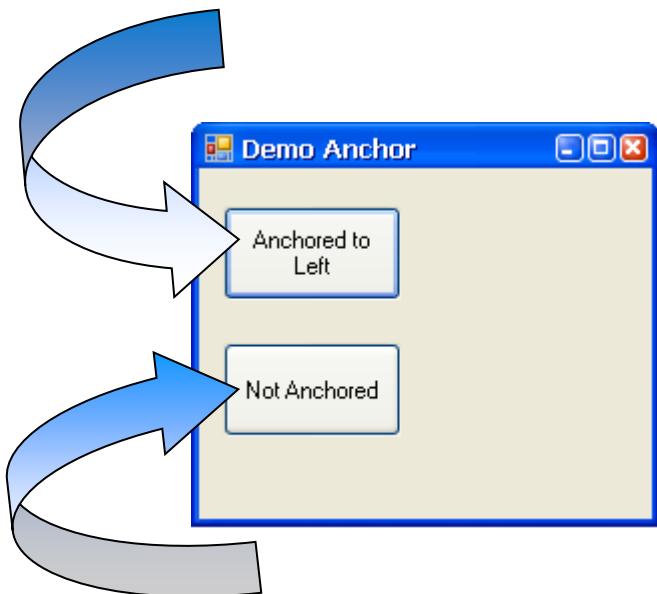


Control Layout - Anchor

- Khi FormBorderStyle = **Sizable**, form cho phép thay đổi kích thước khi Runtime
 - Sự bố trí của control cũng thay đổi!
- Sử dụng thuộc tính **Anchor**
 - Cho phép control phản ứng lại với thao tác resize của form
 - Control có thể thay đổi vị trí tương ứng với việc resize của form
 - Control cố định không thay đổi theo việc resize của form
 - Các trạng thái neo
 - **Left**: cố định theo biên trái
 - **Right**: cố định theo biên phải
 - **Top**: cố định theo biên trên
 - **Bottom**: cố định theo biên dưới

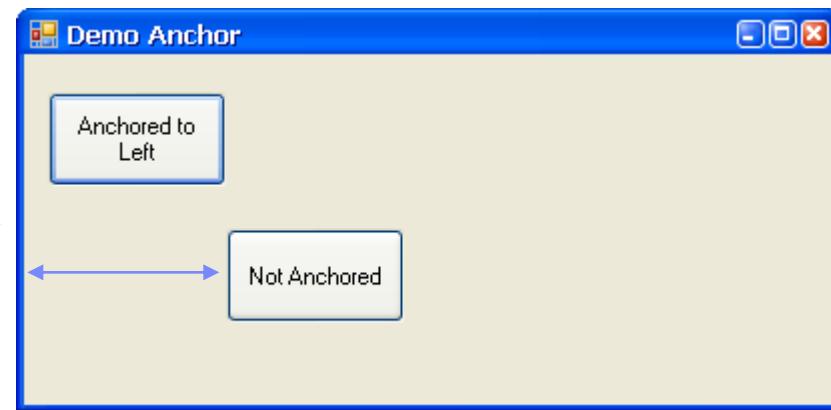
Control Layout - Anchor

Button được neo biên trái



Button tự do

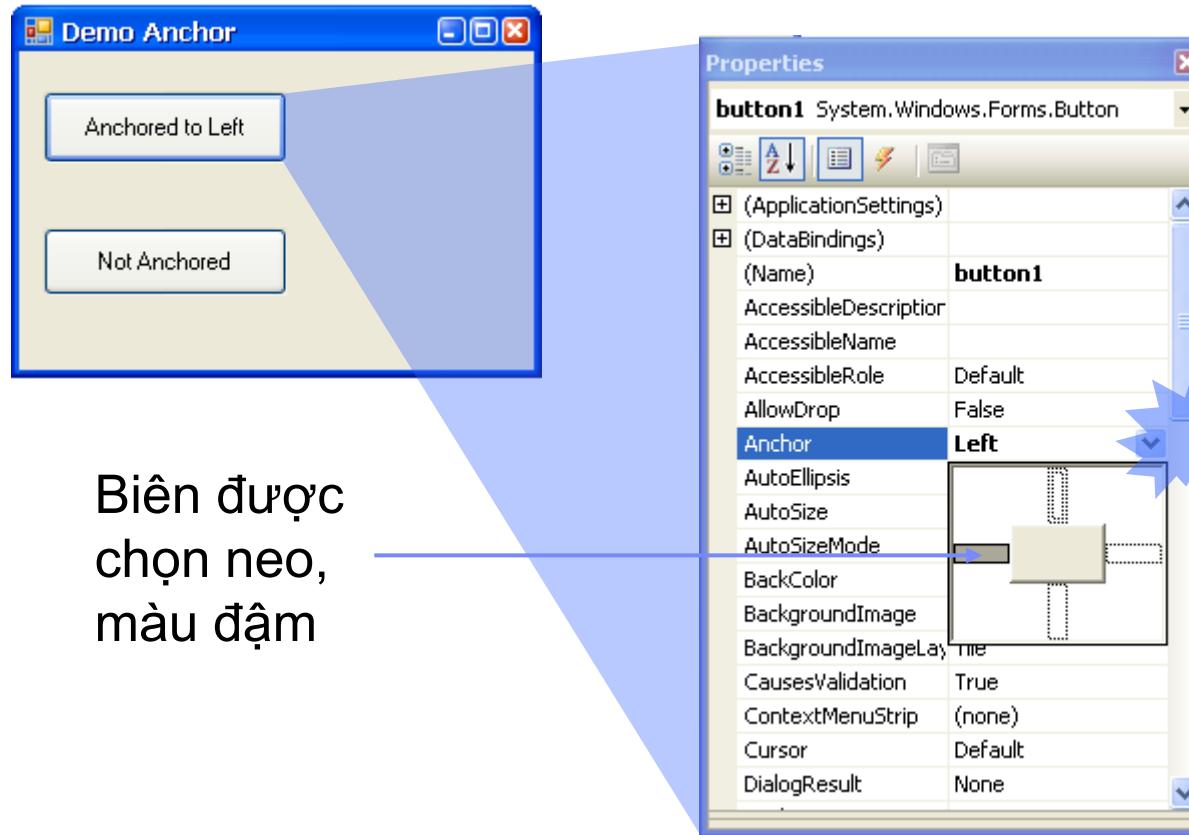
Vị trí tương đối với biên trái không đổi



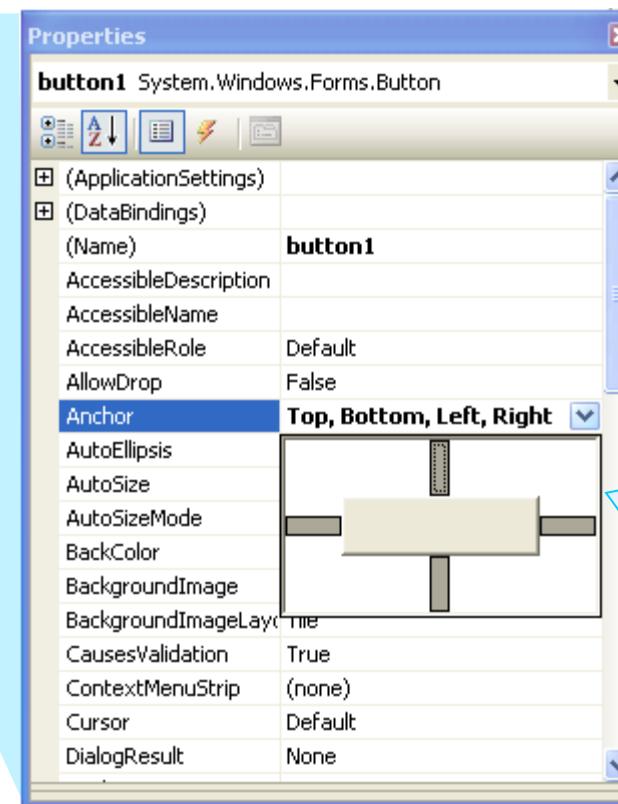
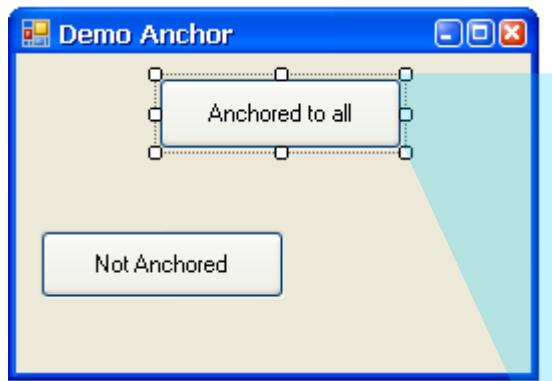
Di chuyển tương ứng theo kích thước mới

Control Layout - Anchor

Thiết lập Anchor cho control

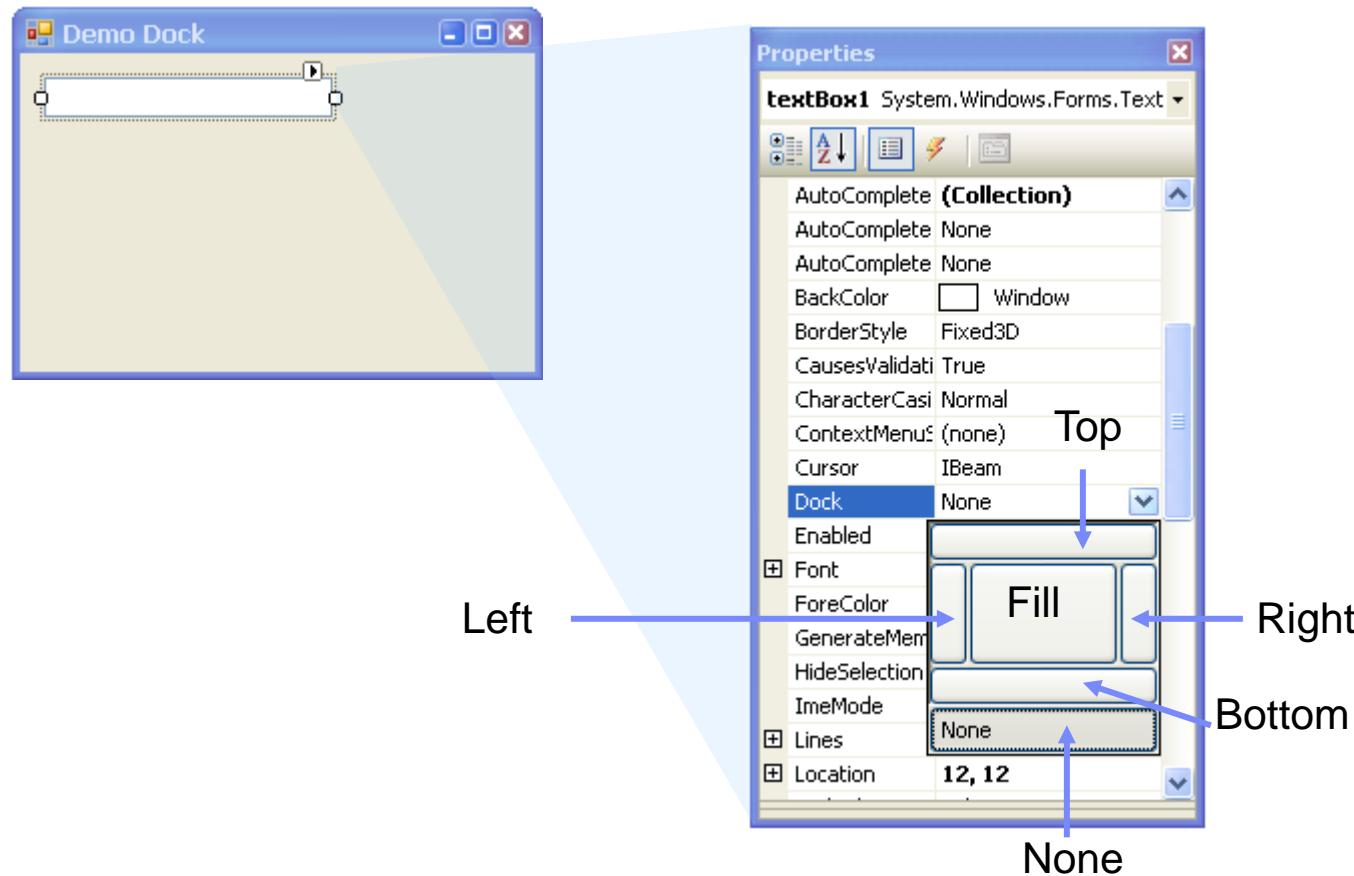


Control Layout - Anchor

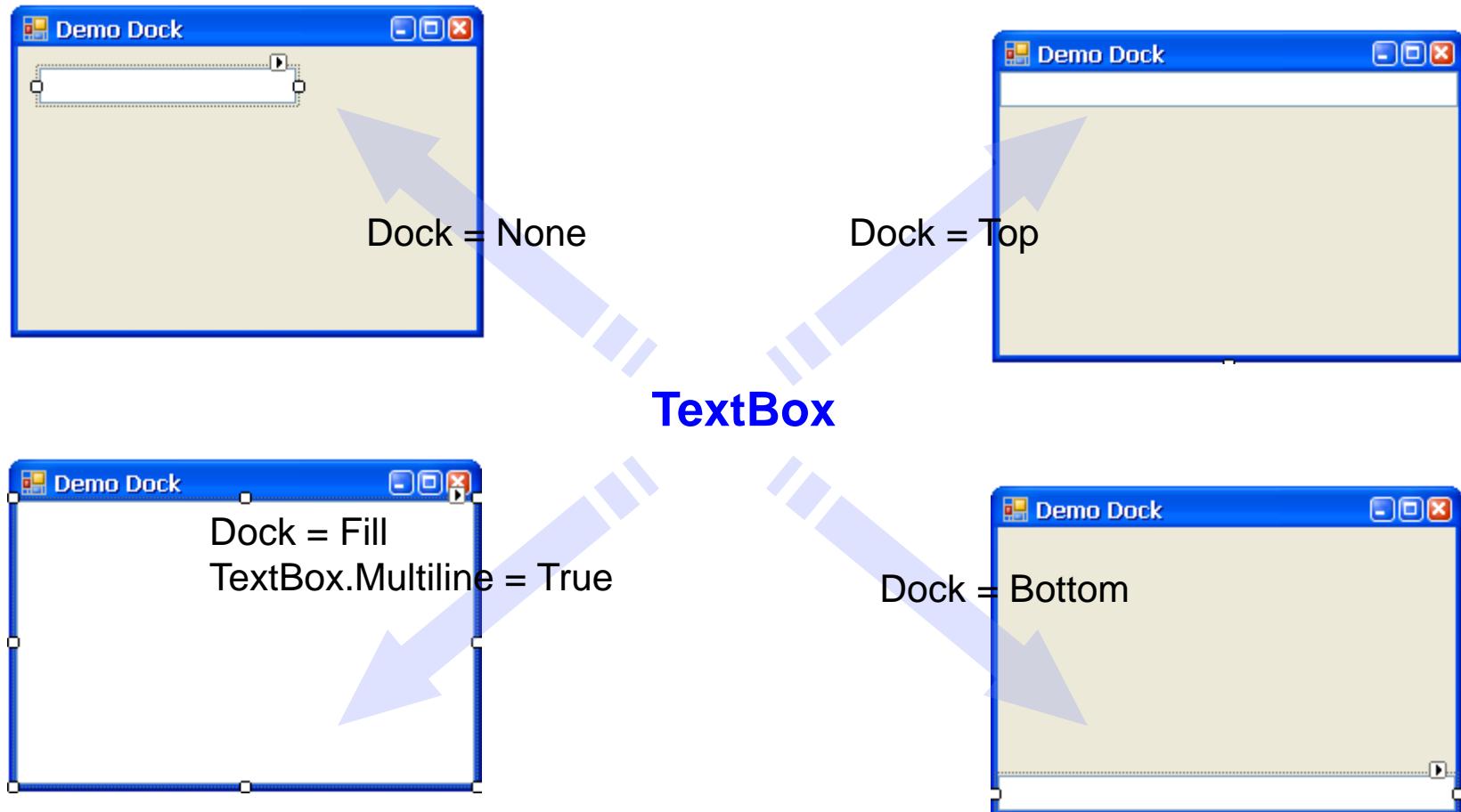


Neo theo
bốn phía

Control Layout - Docking



Control Layout - Docking



Các lớp cơ sở

- System.Windows.Forms.Control - chứa chức năng cơ bản của thao tác xử lý bàn phím và nhập từ chuột và xử lý tin nhắn window.
- System.Windows.Forms.ButtonBase - Lớp này hỗ trợ chức năng cơ bản của một nút
- System.Windows.Forms.TextBoxBase - cung cấp chức năng và thuộc tính thông thường cho các lớp thừa hưởng. Cả hai lớp TextBox và RichTextBox sử dụng chức năng cung cấp bởi TextBoxBase.
- System.Windows.Forms.ScrollableControl - quản lý sự phát sinh và hiển thị của các thanh cuộn đến người dùng để truy cập đến gốc của một hiển thị.
- System.Windows.Forms.ContainerControl - Lớp này quản lý chức năng yêu cầu cho một control để hành động
- System.Windows.Forms.Panel - có thể chứa các control thêm vào, nhưng khác với lớp ContainerControl, nó phân loại các control một cách đơn giản.
- System.Windows.Forms.Form - Tạo bất kỳ loại cửa sổ nào: standard, toolbox, borderless, modal dialog boxes và multi-document interfaces.
- System.Windows.Forms.UserControl - tạo một custom control để việc được dùng trong một nơi phức tạp trong một ứng dụng hay tổ chức

Standard Control

- Một đối tượng control kế thừa trực tiếp / gián tiếp từ **System.Windows.Forms.Control**
- Có các dạng control sau:
 - 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
 - Điều khiển container :
 - GroupBox, TabControl, Panel v.v....

Standard Control

- **Buttons:** Button, CheckBox, RadioButton
- **Time and Date:** DateTimePicker, MonthCalender
- **Labels and Pictures:** GroupBox, Label, LinkLabel, PictureBox
- **Text editing:** TextBox, RichTextbox, NumericUpDown, DomainUpDown
- **Lists and Data:** ListBox, ComboBox, ListView, TreeView, PropertyGrid, DataGridView
- **Position and ProgressBar:** HScrollBar , VScrollBar, TrackBar, ProgressBar
- **Layout:** Tab Control, Splitter, StatusBar, ToolBar



| Bài giảng môn học

Label, Textbox, Button

Lập trình Windows Form với C#

Label

- 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 lè text

Màu text

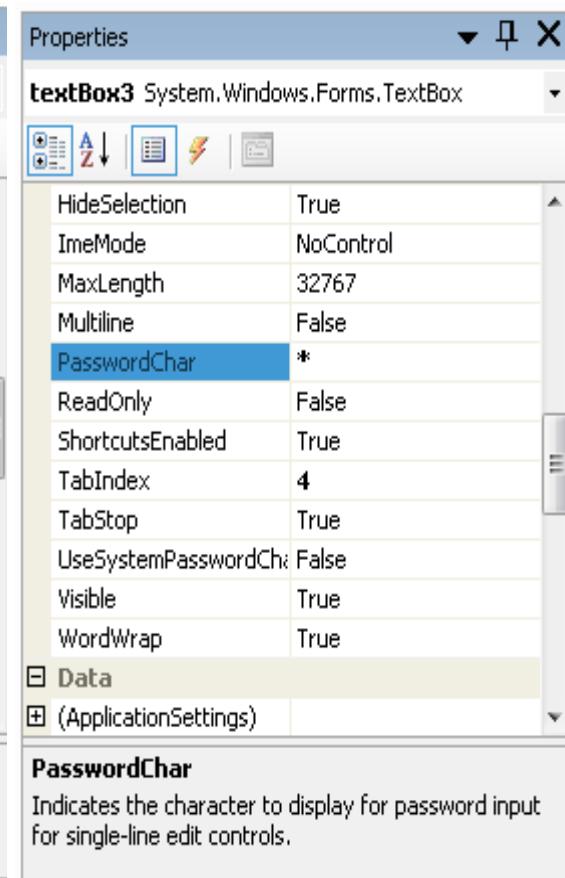
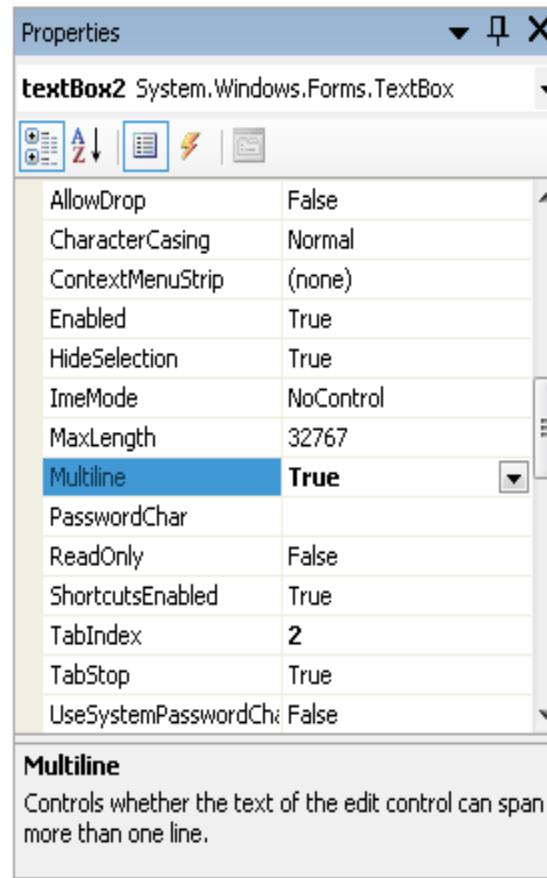
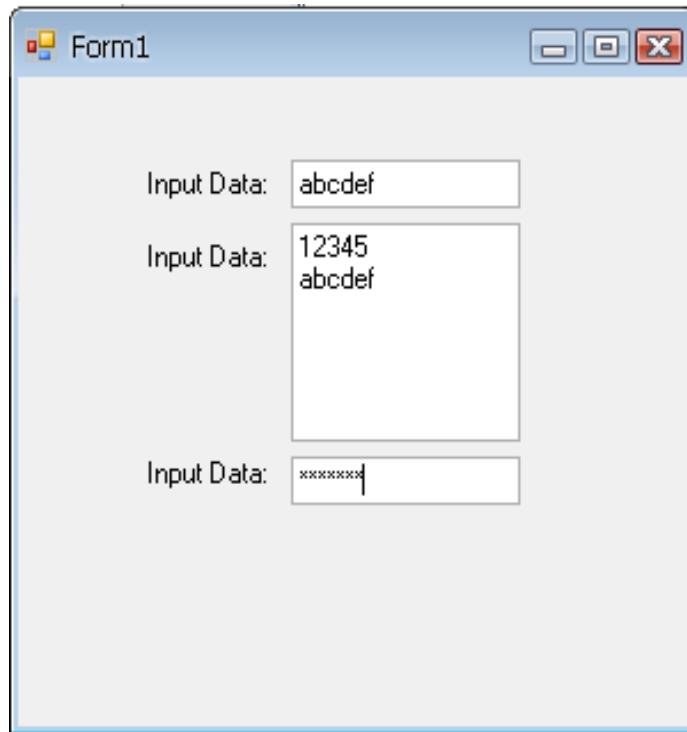
Trạng thái hiển thị

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

TextBox

- Cho phép nhập dữ liệu
- Hai dạng là TextBox và MaskedTextBox
- Các hình thức nhập của TextBox là :
PasswordChar và **Multiline**
- **MaskedTextBox**: có nhiều dạng có sẵn
Numeric, Phone Number, Date, Time v.v...
- **Lấy nội dung nhập** : **[ObjectName].Text**

Textbox



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

Các thuộc tính TextBox

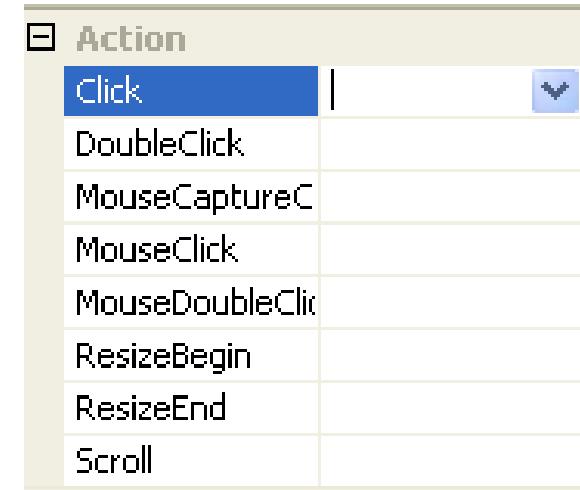
Tên thuộc tính	Ý nghĩa
Text	Lấy hoặc thiết lập giá trị của textbox. Đây là thuộc tính thể hiện dữ liệu mà tất cả các control đều có.
TextLength	Chiều dài tối đa của TextBox
TextAlign	Canh lề chữ trong TextBox
Multiline	Cho phép TextBox có thể nhiều hàng
PasswordChar	Kí tự hiện trong textbox dạng **
SelectedText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

Các sự kiện TextBox

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>

Button

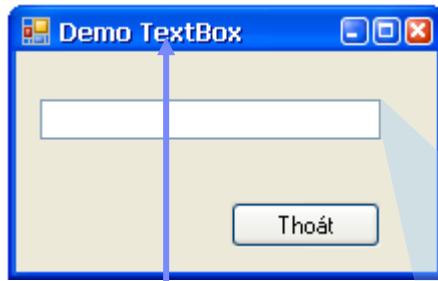
- Dùng chuột nhấn để thao tác
- Có thể show Icon trên button



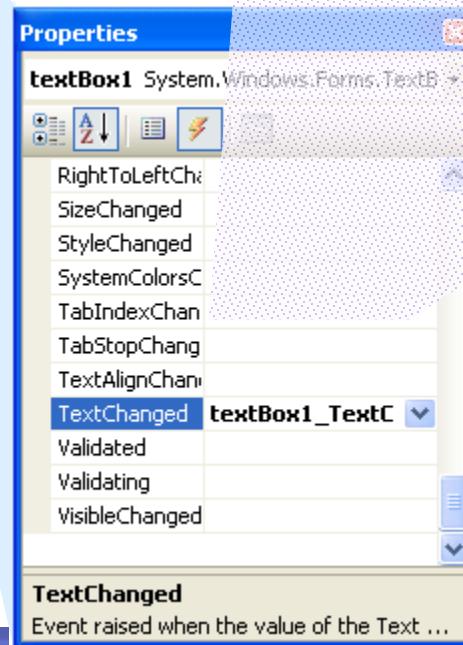
Button	
<i>Thuộc tính thường dùng</i>	
Text	Chuỗi hiển thị trên bề mặt button
<i>Event thường dùng</i>	
Click	Kích hoạt khi user kích vào button, khai báo mặc định khi người lập trình kích đúp vào button trong màn hình Design View của Form.

Demo Label, TextBox, Button

Demo TextBox



Double click vào textbox
để tạo event handler
cho event TextChanged

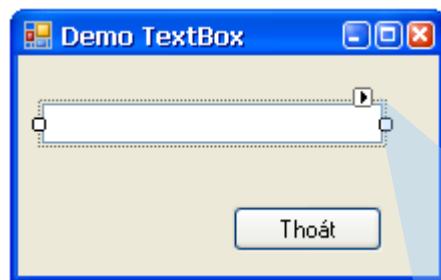


```
private void textBox1_TextChanged(object sender, EventArgs e)
{
    string text;
    text = ((TextBox)sender).Text;
    ((TextBox)sender).Text = text.ToUpper();
}
```

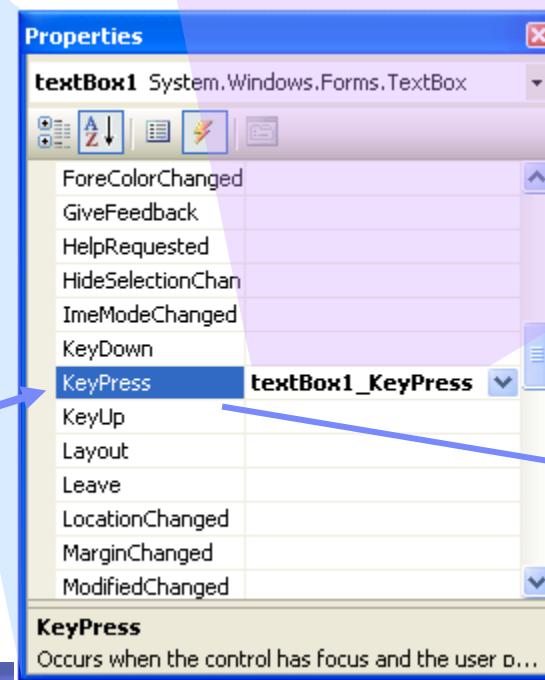
Chuyển thành chữ hoa

Demo Label, TextBox, Button

▪ Chỉ cho nhập số



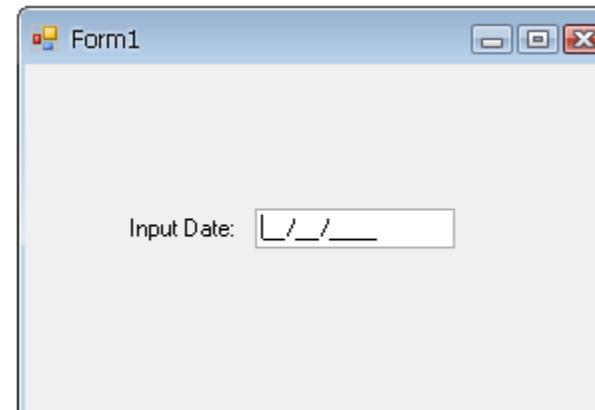
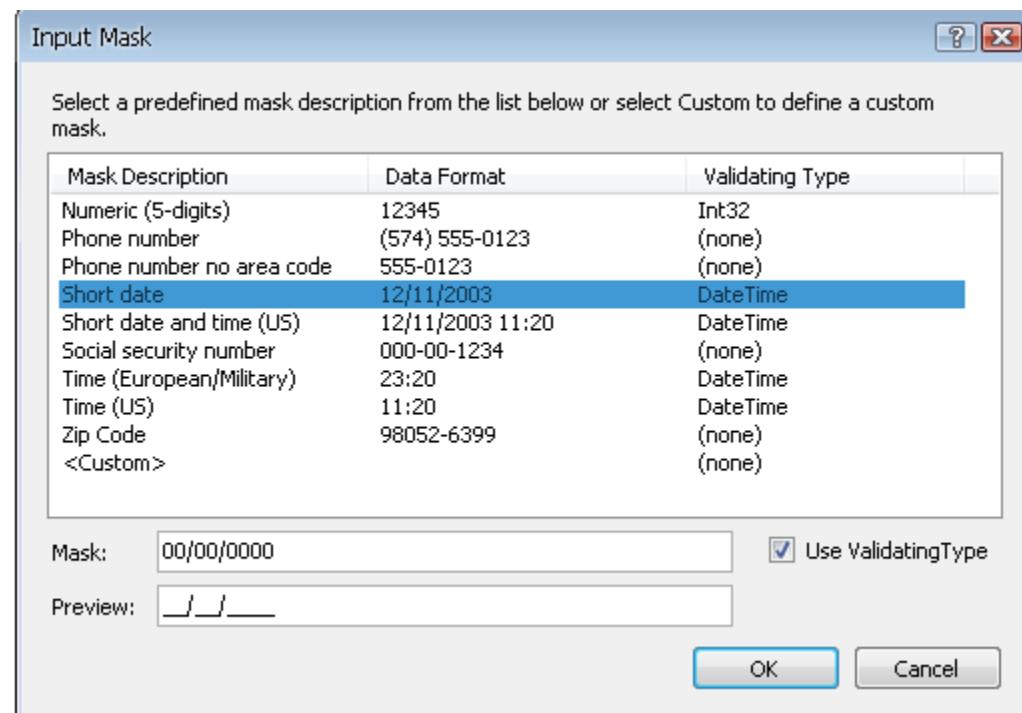
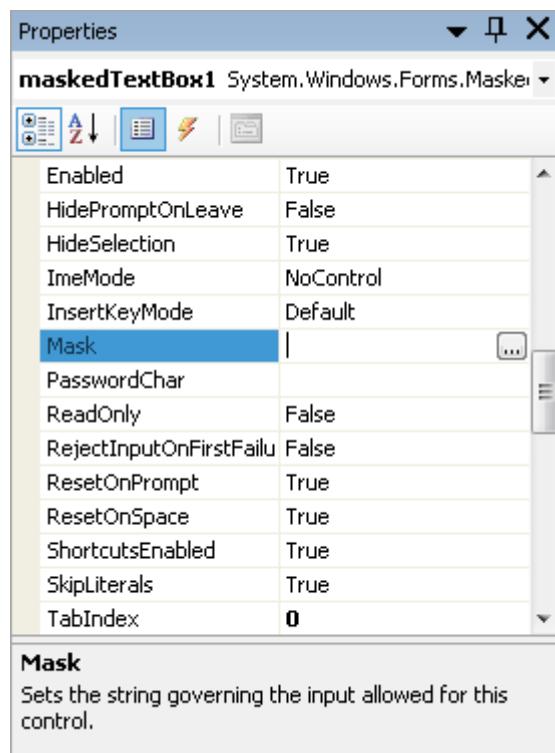
Sự kiện KeyPress



```
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar)) // không là ký tự số
        e.Handled = true; // đã xử lý sự kiện keypress
}
```

Sự kiện phát sinh khi
textbox nhận focus và
user nhấn 1 phím

MaskedTextBox



MaskEditBox - Thiết lập giá trị Mask

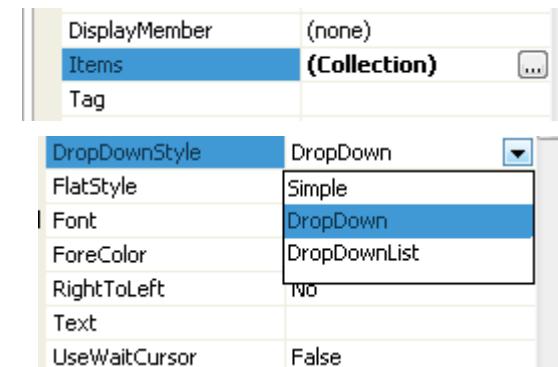
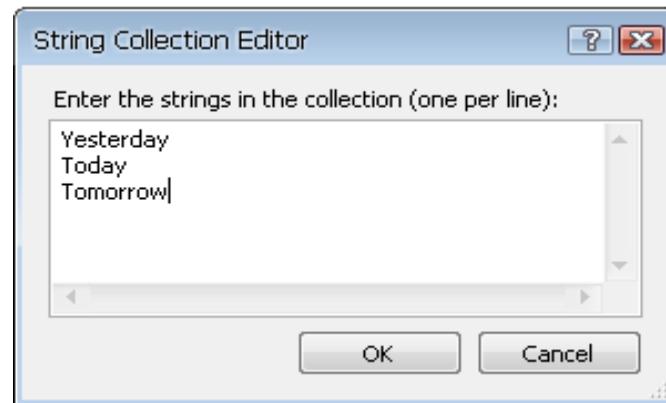
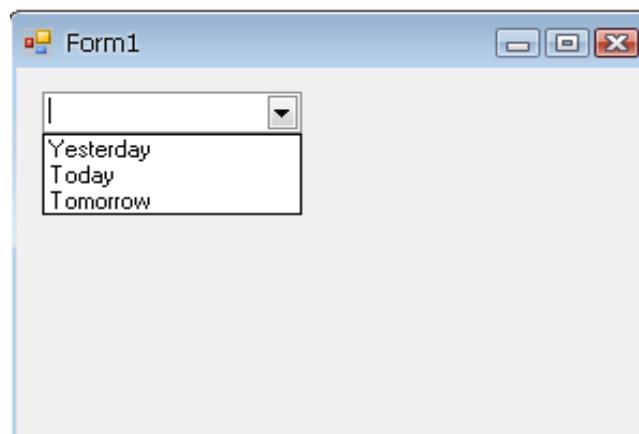
Thành phần mask	Ý nghĩa
0	Số. Yêu cầu bắt buộc phải nhập số từ 0-9
9	Số hoặc khoảng trắng (Optional)
#	Số hoặc khoảng trắng (Optional). Có thể nhập dấu + hoặc -
L	Kí tự [a..z] hoặc [A..Z] (Bắt buộc)
?	Kí tự [a..z] hoặc [A..Z] (Không bắt buộc)
,	Đơn vị phần ngàn (1,234)
.	Đơn vị phần lẻ (0.32)

Combobox, ListBox

- **Liệt kê danh sách phần tử cho phép người dùng chọn một hay nhiều**
- **Có các thuộc tính chung quan trọng sau:**
 - **DataSource**: chọn tập dữ liệu điền vào điều khiển (VD string [] , ArrayList là tập dữ liệu đưa vào)
 - **SelectedText**, **SelectValue**, **SelectedItem**, **SelectedIndex** để lấy giá trị hay đối tượng chọn
- **Thuộc tính riêng:**
 - Combobox: **DropDownStyle** (**DropDown**: cho phép nhập thêm chuỗi ; **DropDownList**: chỉ cho phép chọn chuỗi/item đã có)
 - ListBox: **SelectionMode** (True: cho phép chọn nhiều)

Combobox

Combobox: chọn Items để thêm dữ liệu



```

private void Form1_Load
    (object sender, EventArgs e)
{
    //Code Here
    string[] data = { "Yesterday", "Today", "Tomorrow" };
    comboBox1.DataSource = data;
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    MessageBox.Show(comboBox1.SelectedItem.ToString());
}

```

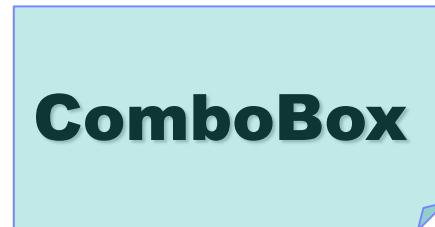
ComboBox

- Kết hợp TextBox với một danh sách dạng drop down
- Cho phép user kích chọn item trong danh sách drop down

Items

DropDownStyle

Text



Sorted

MaxDropDownItems

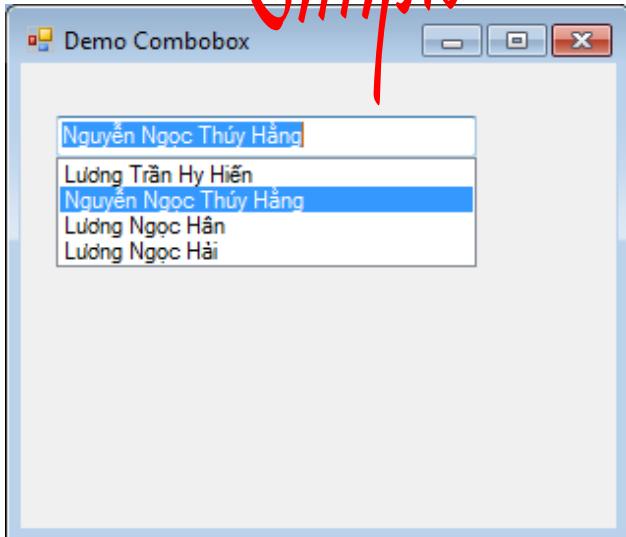
AutoCompleteMode

DropDownHeight

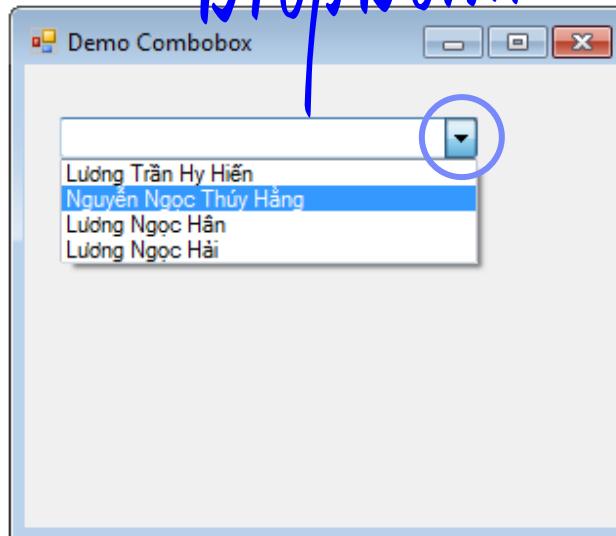
ComboBox

■ *DropDownStyle*

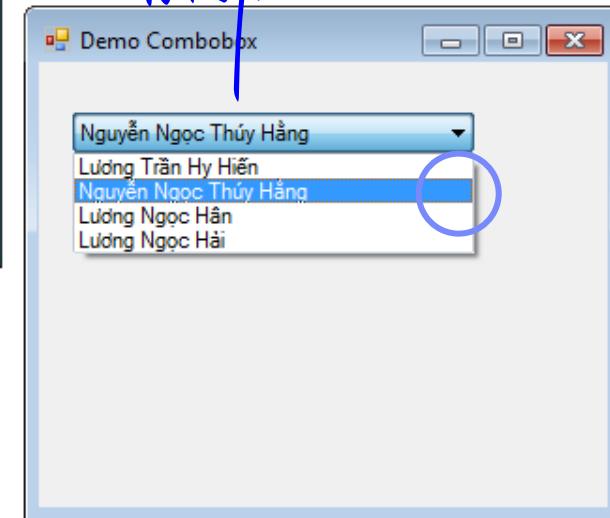
Simple



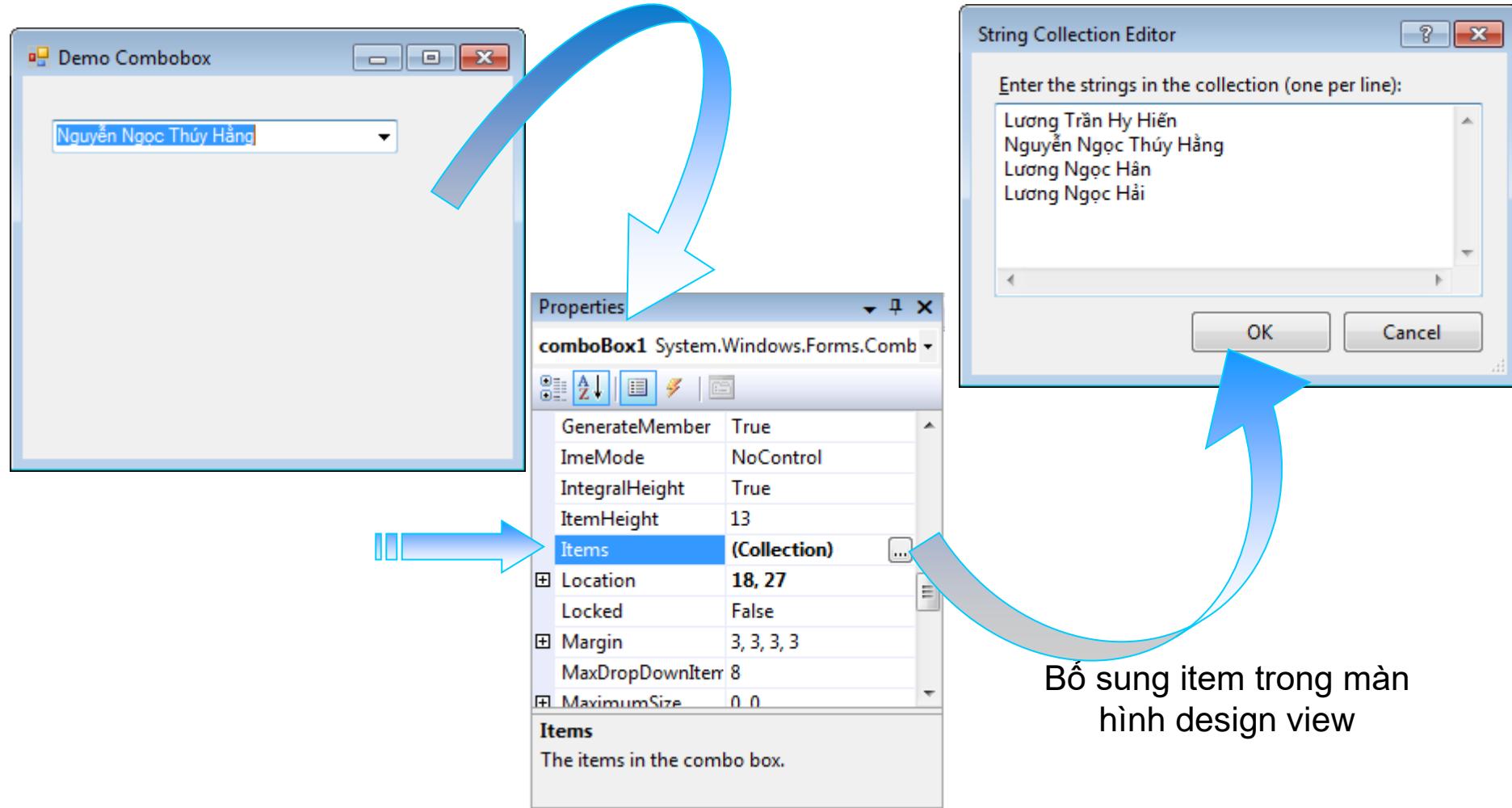
DropDown



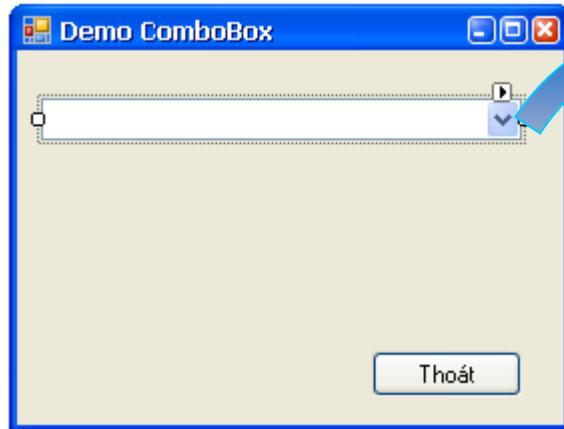
DropDownList



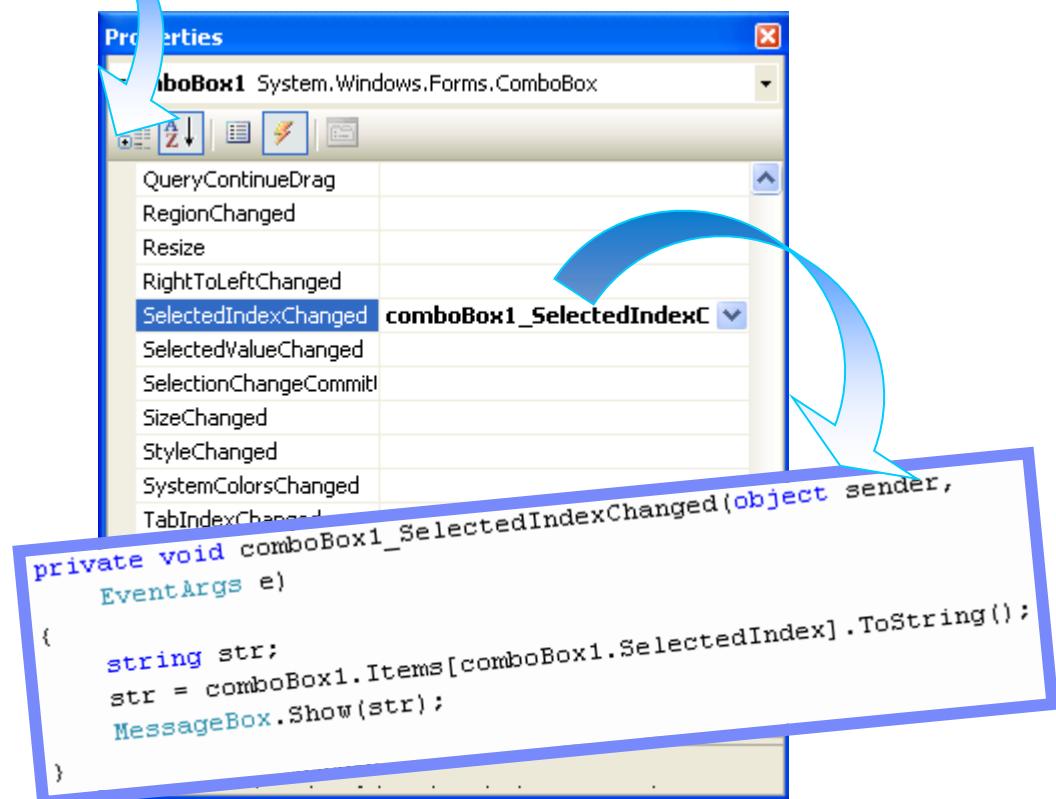
ComboBox



ComboBox

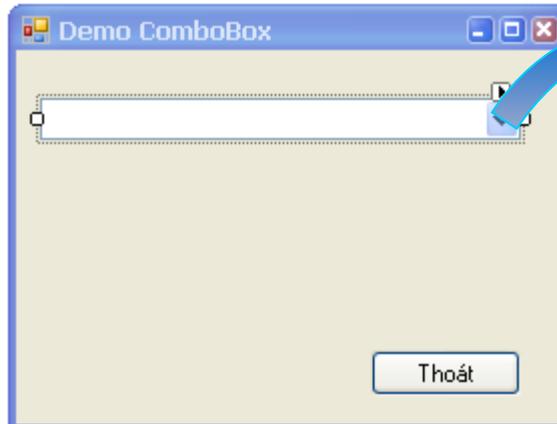


Mỗi khi kích chọn
một item ⇒ hiển thị
item được chọn
trên MessageBox



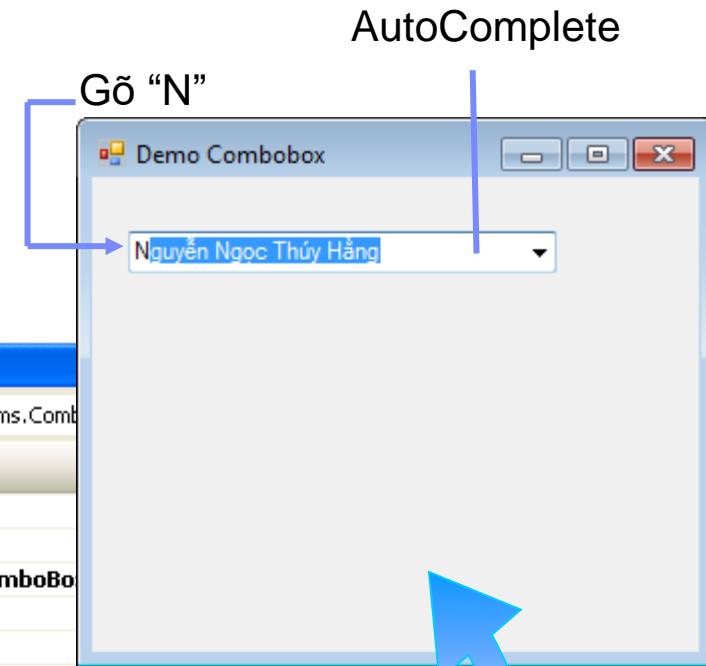
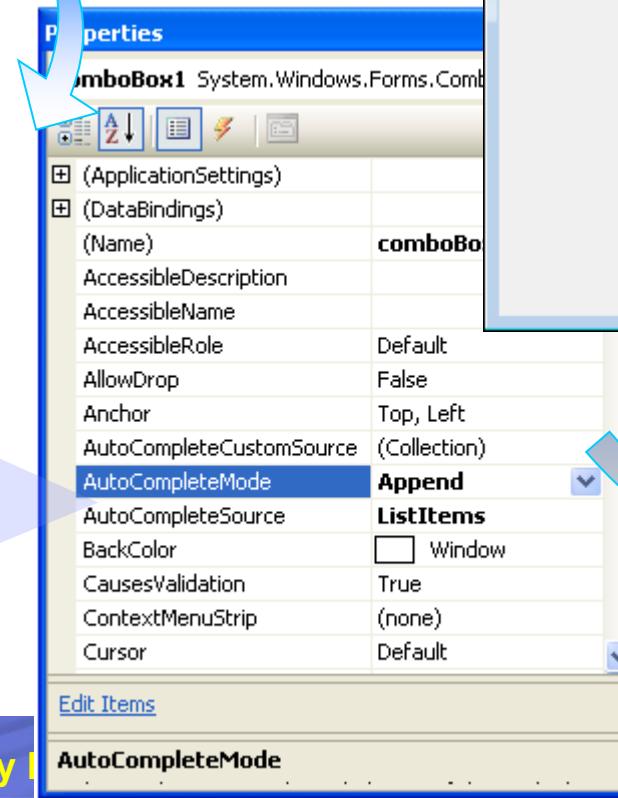
ComboBox

- Tính năng **AutoComplete**



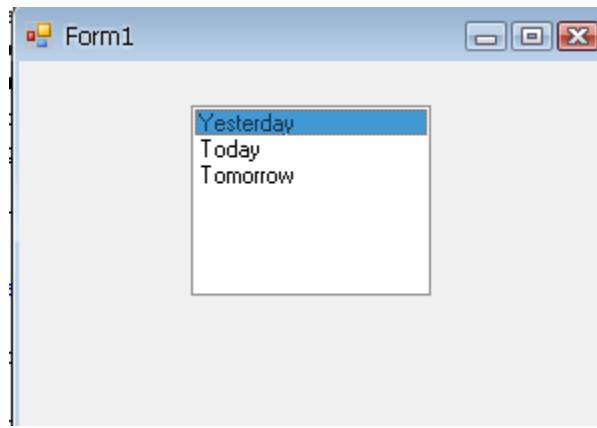
AutoCompleteMode

AutoCompleteSource



AutoComplete

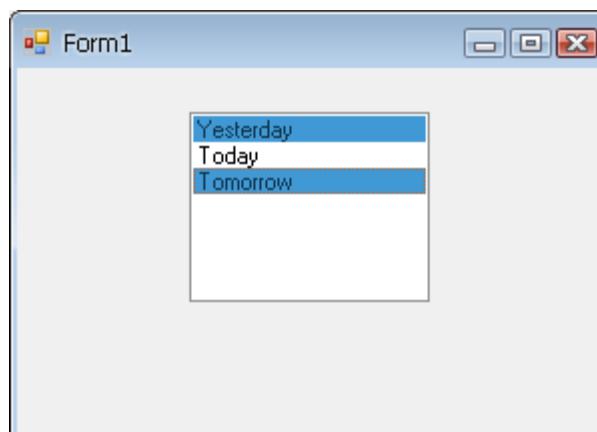
Listbox



DisplayMember	(none)
Items	(Collection)
Tag	
DropDownStyle	DropDown
FlatStyle	Simple
Font	DropDown
ForeColor	DropDownList
RightToLeft	Two
Text	
UseWaitCursor	False

SelectionMode	MultiSimple
Sorted	None
TabIndex	One
TabStop	MultiSimple
UseTabStops	MultiExtended
Visible	True

SelectionMode
Indicates if the list box is to be single-select, multi-select, or not selectable.



```
private void Form1_Load
    (object sender, EventArgs e)
{
    //Code Here
    string[] data = { "Yesterday", "Today", "Tomorrow" };
    listBox1.DataSource = data;
}
```

ListBox

■ ***Method & Event***

Method

ClearSelected

GetSelected

SetSelected

FindString

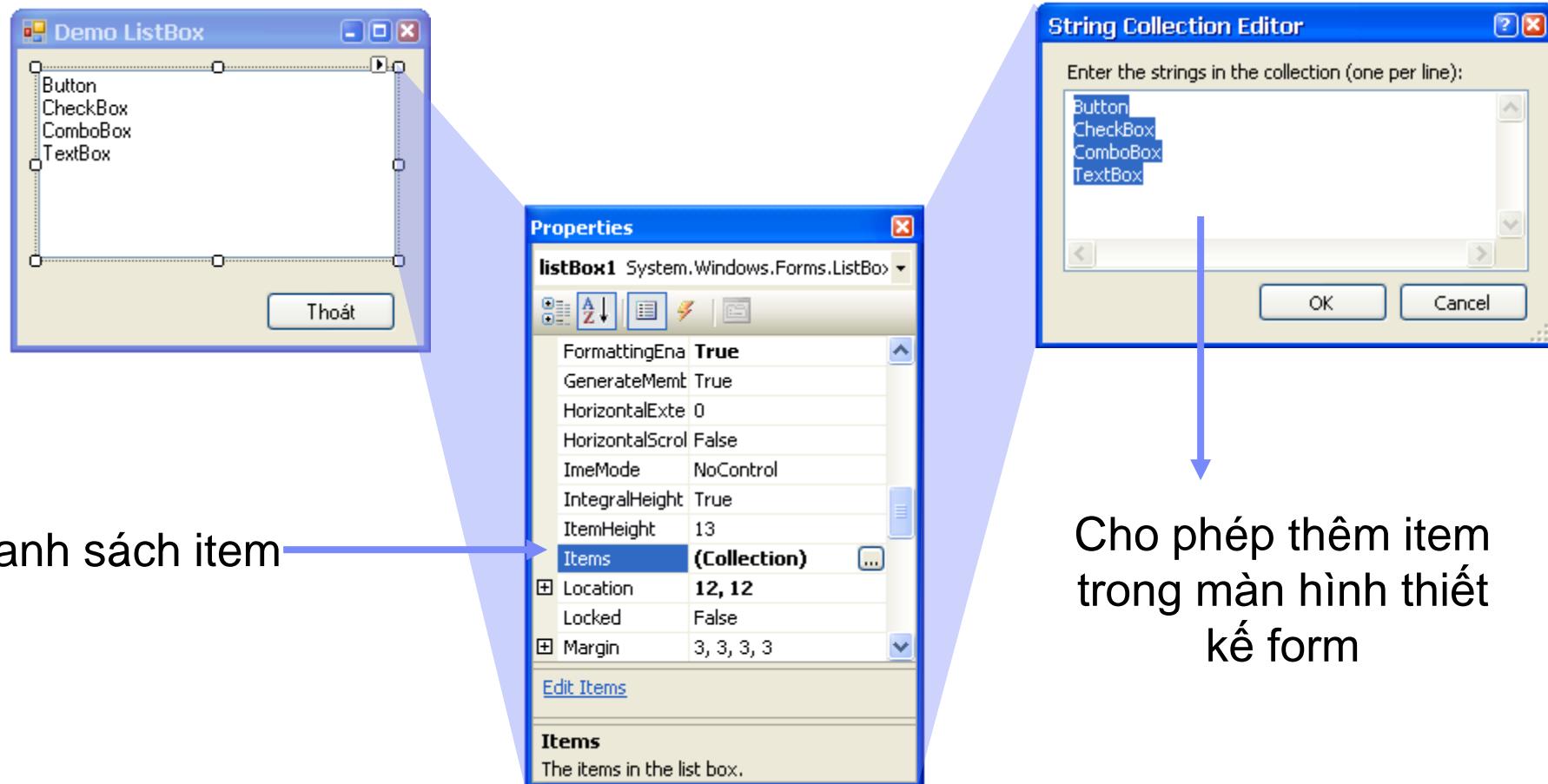


SelectedIndexChanged

SelectedValueChanged

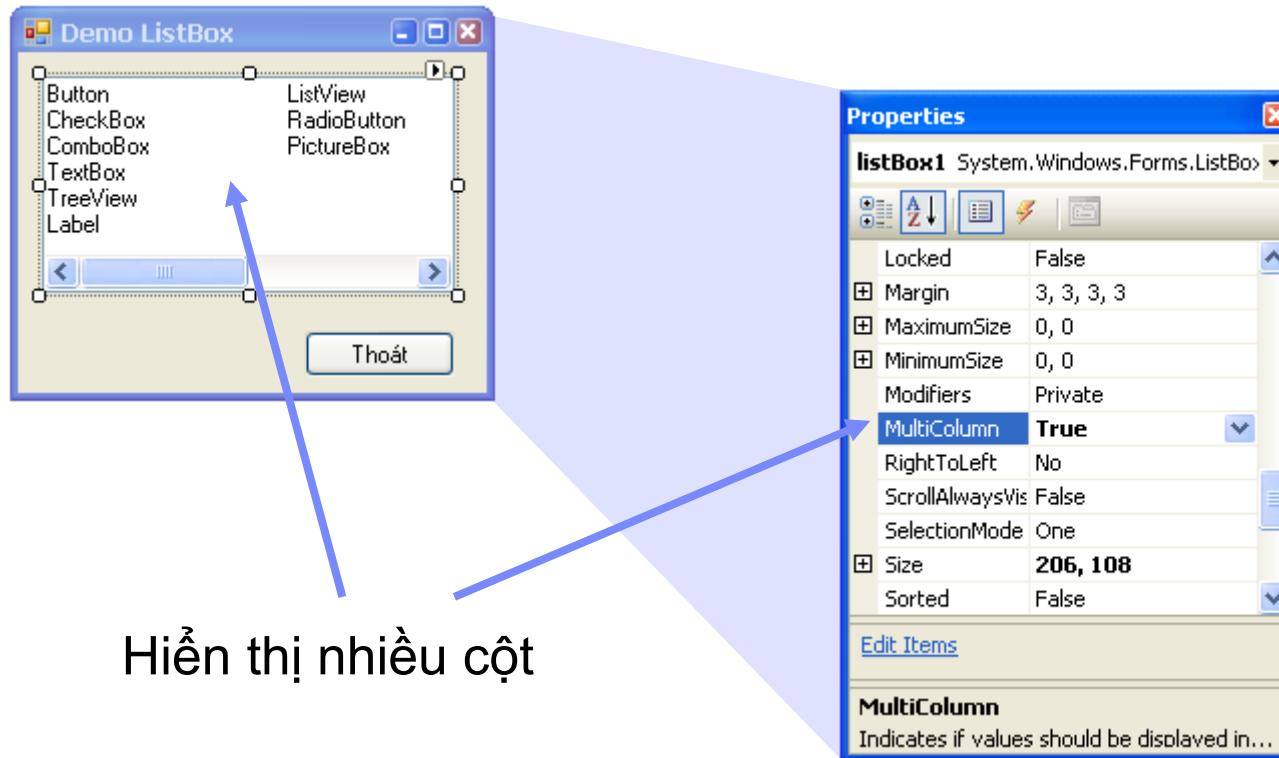
ListBox

- Thuộc tính **Items** cho phép thêm item vào ListBox



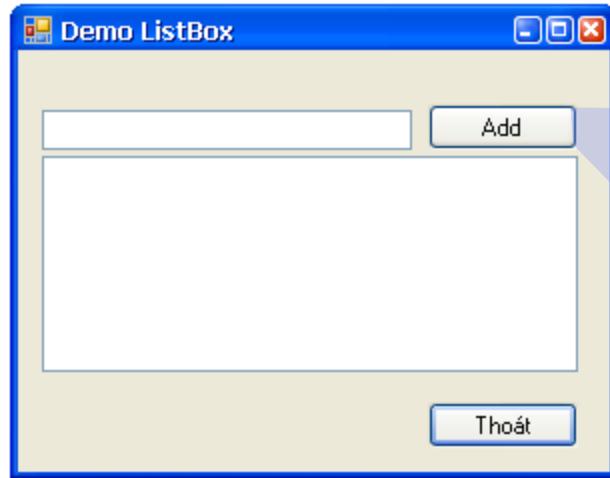
ListBox

■ ListBox hiển thị dạng Multi Column



ListBox

Demo *ListBox*

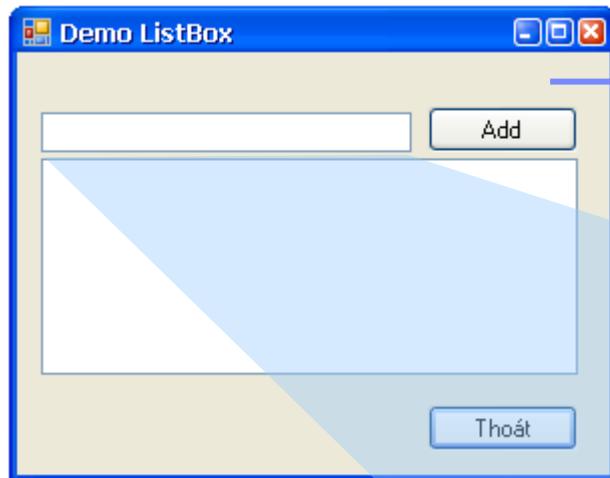


```
private void button2_Click(object sender, EventArgs e)
{
    if (listBox1.Items.IndexOf(textBox1.Text) > -1)
        listBox1.SelectedItem = textBox1.Text;
    else if (textBox1.Text.Length > 0)
        listBox1.Items.Add(textBox1.Text);
}
```

Kiểm tra xem chuỗi nhập có trong listbox?
- Nếu có: select item đó.
- Ngược lại: thêm chuỗi mới vào listbox.

ListBox

- Sự kiện ***SelectedIndexChanged***



Mỗi khi kích chọn vào item trong listbox ⇒ sẽ xóa item được chọn tương ứng

SelectedIndexChanged

```
private void listBox1_SelectedIndexChanged(object sender,
    EventArgs e)
{
    if (listBox1.SelectedIndex >= 0)
        listBox1.Items.RemoveAt(listBox1.SelectedIndex);
}
```



| Bài giảng môn học

CheckBox, RadioButton, CheckListBox

Lập trình Windows Form với C#

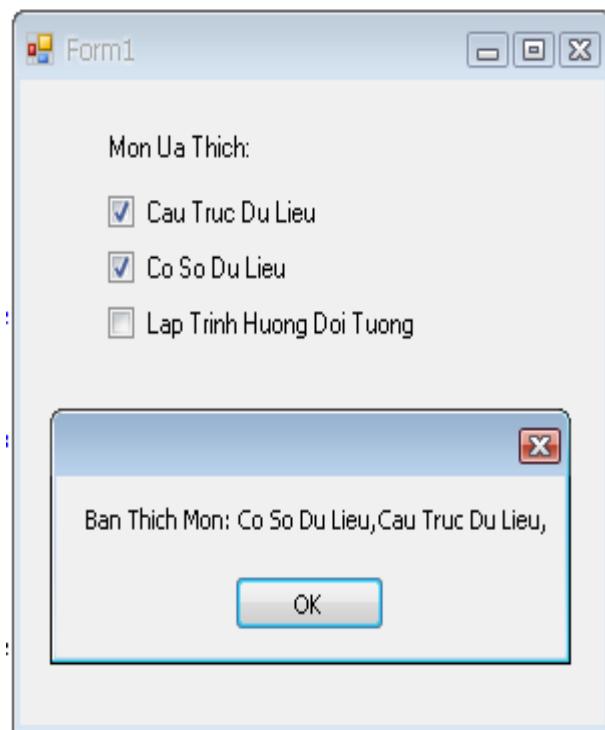
Điều khiển thông thường

- **CheckBox, RadioButton:**

- cho phép người dùng chọn một hay nhiều giá trị
- thuộc tính **Checked** (true/false) để kiểm tra xem có check chọn hay không
- thuộc tính **Text** (string) để biết chọn nội dung gì.

- **CheckListBox:** Kết hợp checkbox trong ListBox trong trường hợp có quá nhiều lựa chọn

CheckBox



```

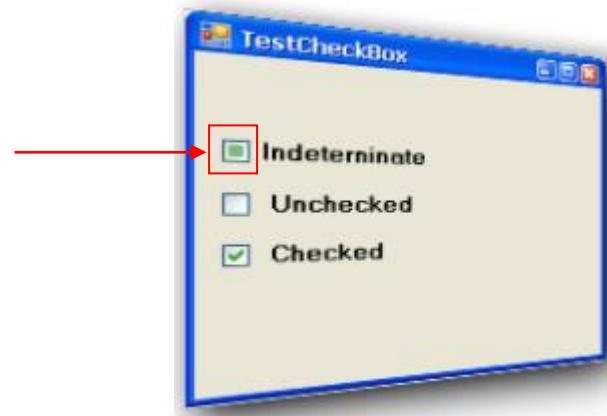
private void button1_Click(object sender, EventArgs e)
{
    string choose = "";
    foreach (Control chk in this.Controls)
    {
        if (chk is CheckBox)
        {
            if (((CheckBox)chk).Checked == true)
                choose += chk.Text + ",";
        }
    }
    MessageBox.Show("Ban Thich Mon: " + choose);
}

```

CheckBox

- **ThreeState = true** : cho phép thiết lập 3 trạng thái:
 - Checkstate = **Indeterminate**: không xác định
 - CheckState= **Checked**: chọn
 - CheckState= **Unchecked**: không chọn

Chưa chọn



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 (radioButton3.Checked) MessageBox.Show("You are Divorce");
}
```

RadioButton

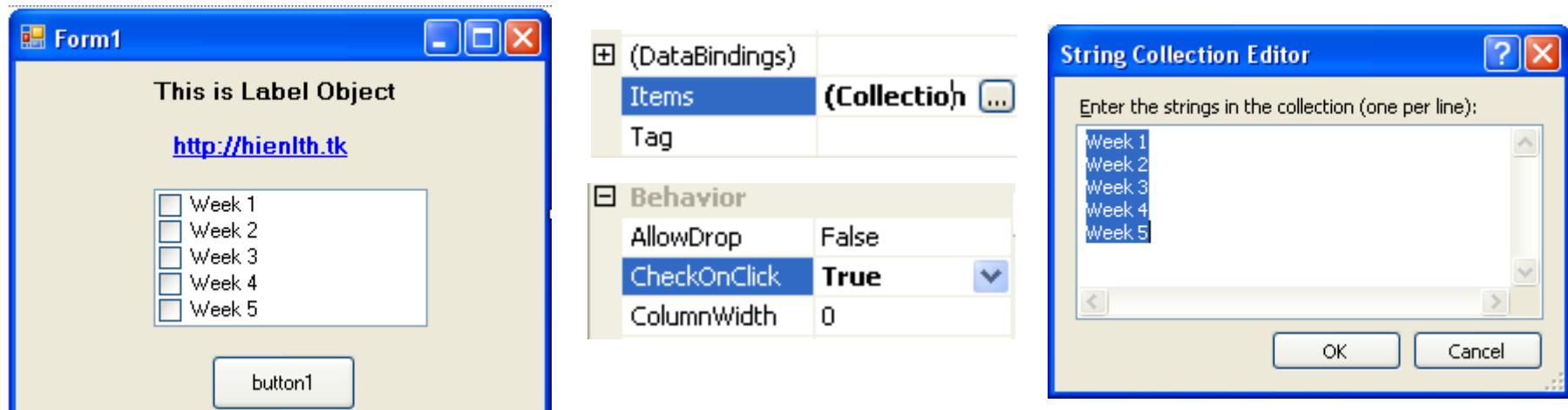
Nhóm RadioButton
thứ 1 chứa trong
GroupBox1



Nhóm RadioButton
thứ 2 chứa trong
GroupBox2

CheckListBox

- Phương thức Add cho phép thêm dữ liệu. Hoặc dùng Properties → Items



CheckListBox

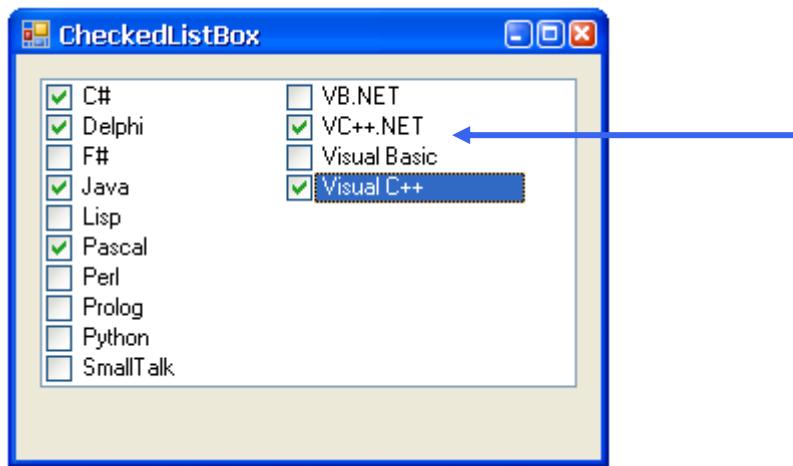
- Thuộc tính cơ bản: **Items**
- Sự kiện cơ bản: **SelectedIndexChanged**



```
private void button1_Click(object sender, EventArgs e)
{
    string strvalue = "";
    foreach (string str in checkedListBox1.CheckedItems)
        strvalue += str + ",";
    MessageBox.Show("Bạn đã chọn: " + strvalue);
}
```

CheckedListBox

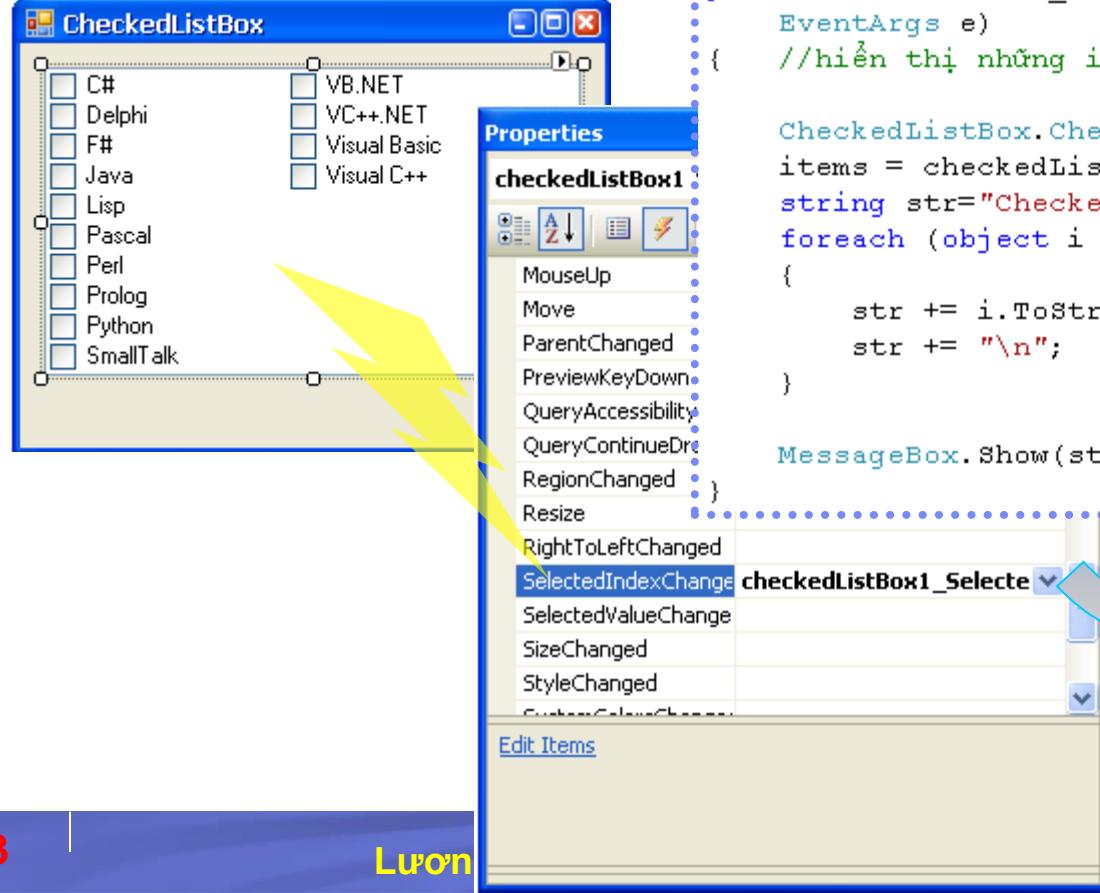
- ***MultiColumn = true***



Các item được tổ chức
theo nhiều cột

CheckedListBox

Sự kiện SelectedIndexChanged



```
private void chkLB1_SelectedIndexChanged(object sender,
EventArgs e)
{
    //hiển thị những item được check
    CheckedListBox.CheckedItemCollection items;
    items = checkedListBox1.CheckedItems;
    string str="Checked items: \n";
    foreach (object i in items)
    {
        str += i.ToString();
        str += "\n";
    }

    MessageBox.Show(str);
}
```



| Bài giảng môn học

GroupBox, Panel & Tab Control

Lập trình Windows Form với C#

GroupBox & Panel

■ **Bố trí controls trên GUI**

■ **GroupBox**

- Hiển thị một khung bao quanh một nhóm control
- Có thể hiển thị một tiêu đề
 - Thuộc tính Text
- Khi xóa một GroupBox thì các control chứa trong nó bị xóa theo
- Thiết lập giá trị của GroupBox sẽ ảnh hưởng đến các control mà nó chứa
- Lớp GroupBox kế thừa từ System.Windows.Forms.Control

■ **Panel**

- Chứa nhóm các control
- Không có caption
- Có thanh cuộn (scrollbar)
 - Xem nhiều control khi kích thước panel giới hạn

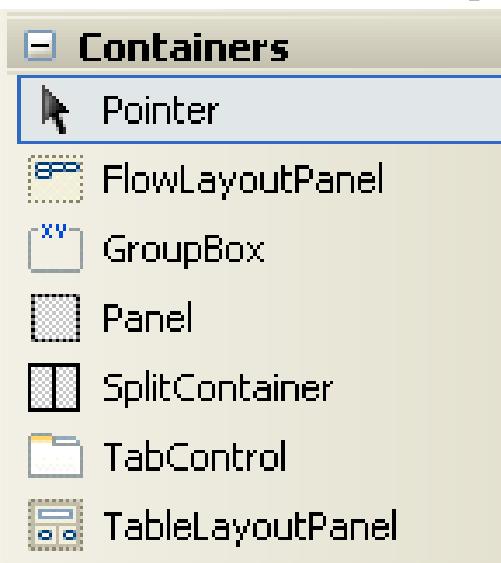
GroupBox & Panel

GroupBox	Mô tả
<i>Thuộc tính thường dùng</i>	
Controls	Danh sách control chứa trong GroupBox.
Text	Caption của GroupBox

Panel	
<i>Thuộc tính thường dùng</i>	
AutoScroll	Xuất hiện khi panel quá nhỏ để hiển thị hết các control, mặc định là false
BorderStyle	Biên của panel, mặc định là None, các tham số khác như Fixed3D, FixedSingle
Controls	Danh sách control chứa trong panel

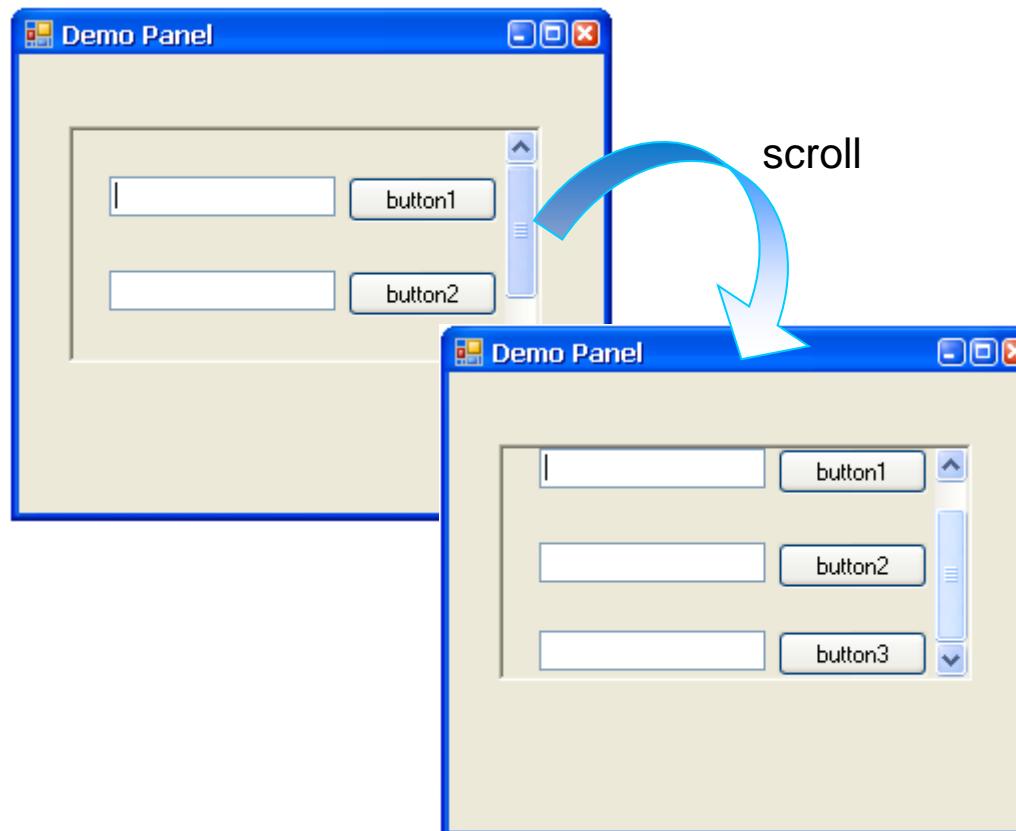
GroupBox & Panel

▪ Minh họa *GroupBox*



GroupBox & Panel

- Minh họa *Panel*

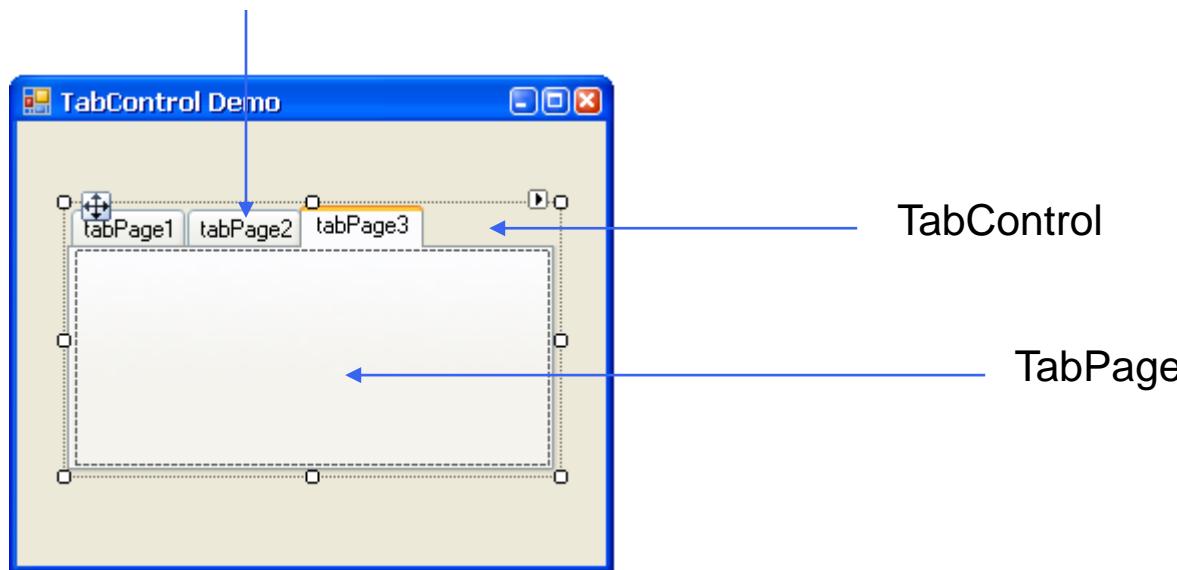


TabControl

- **Dạng container chứa các control khác**
- **Cho phép thể hiện nhiều page trên một form duy nhất**
- **Mỗi page chứa các control tương tự như group control khác.**
 - Mỗi page có tag chứa tên của page
 - Kích vào các tag để chuyển qua lại giữa các page
- **Ý nghĩa:**
 - Cho phép thể hiện nhiều control trên một form
 - Các control có cùng nhóm chức năng sẽ được tổ chức trong một tab (page)

TabControl

- **TabControl có thuộc tính TabPages**
 - Chứa các đối tượng TabPage



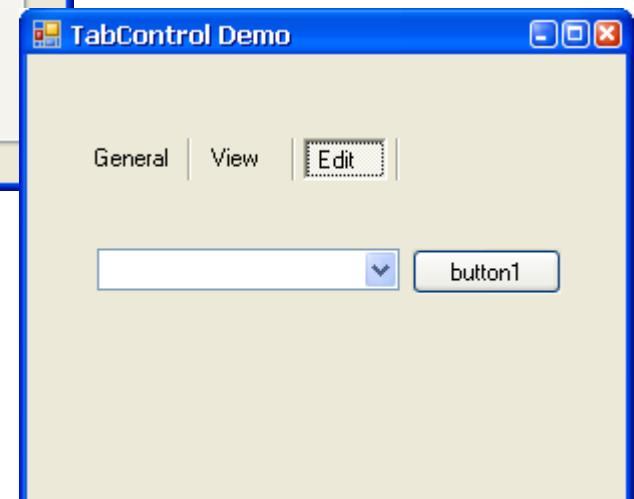
TabControl

- Thuộc tính **Appearance**
Normal

Buttons



FlatButton



TabControl

- Thuộc tính, phương thức & sự kiện thường dùng

Properties

TabPage

TabCount

SelectedTab

Multiline

SelectedIndex



Method

SelectTab

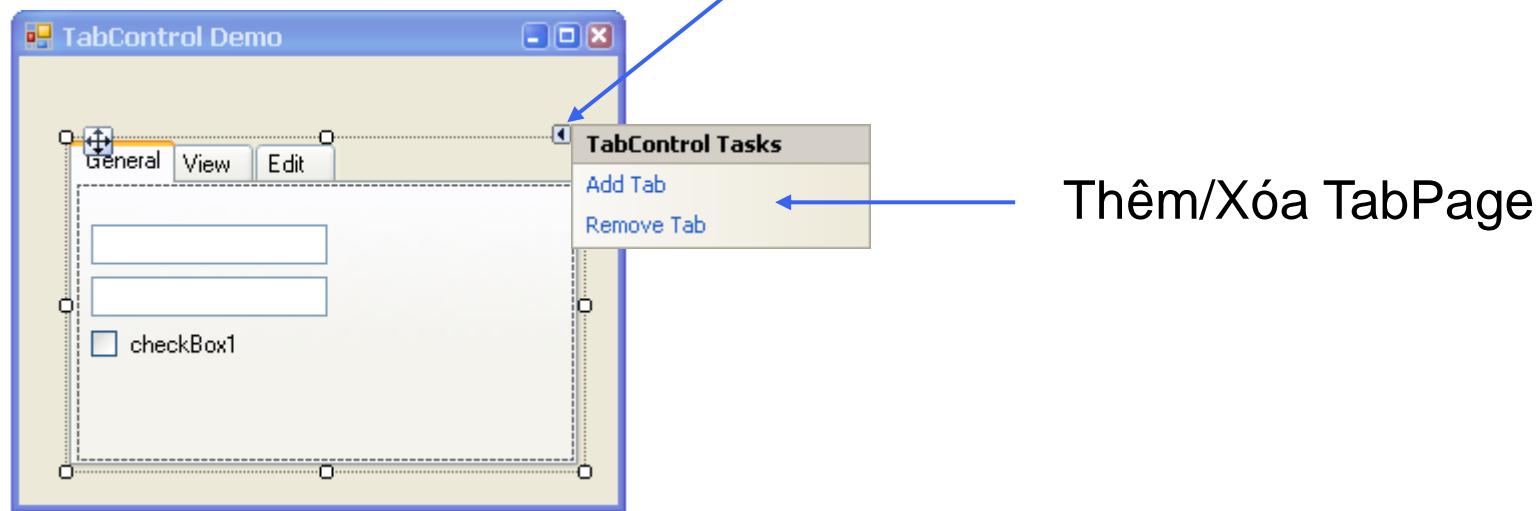
DeselectTab

Event

SelectedIndexChanged

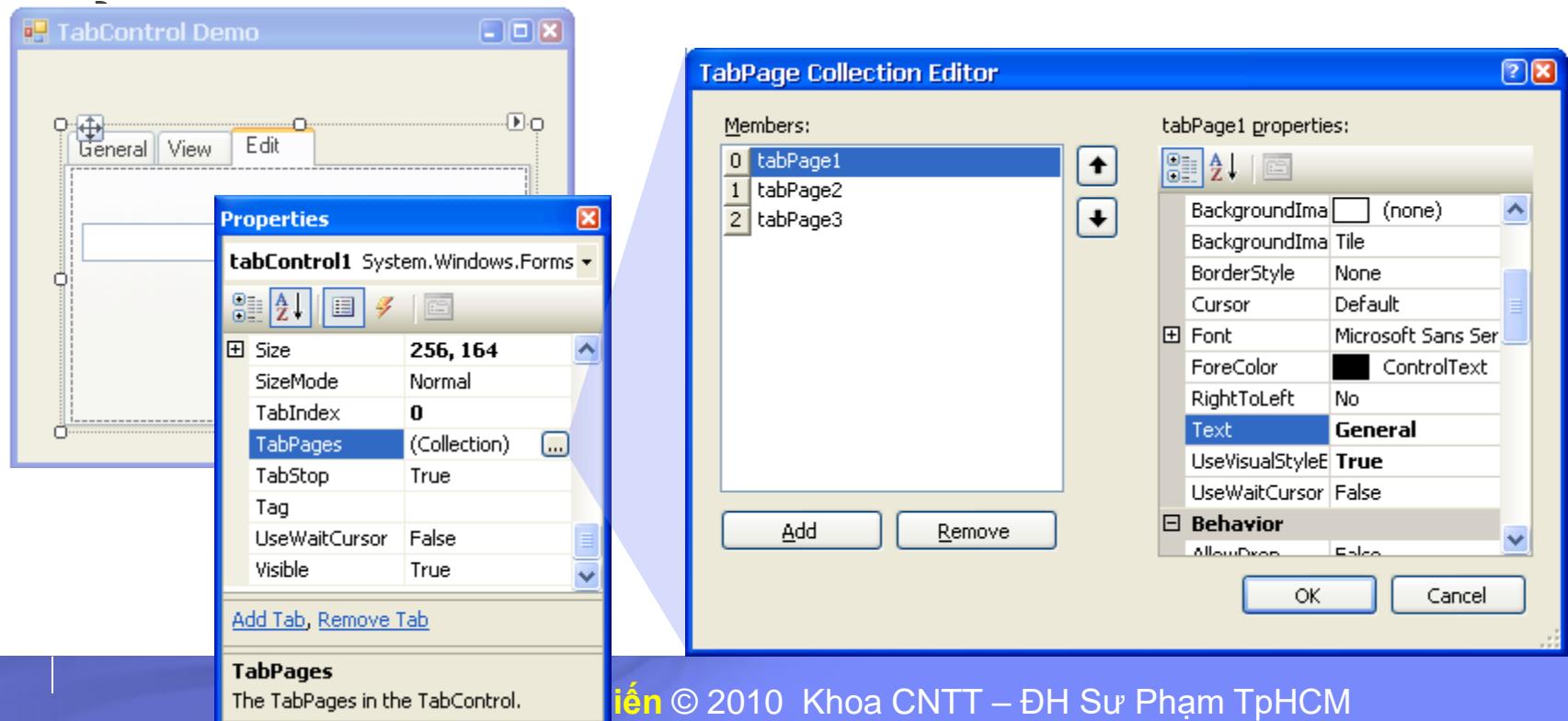
TabControl

- Thêm/Xóa TabPage



TabControl

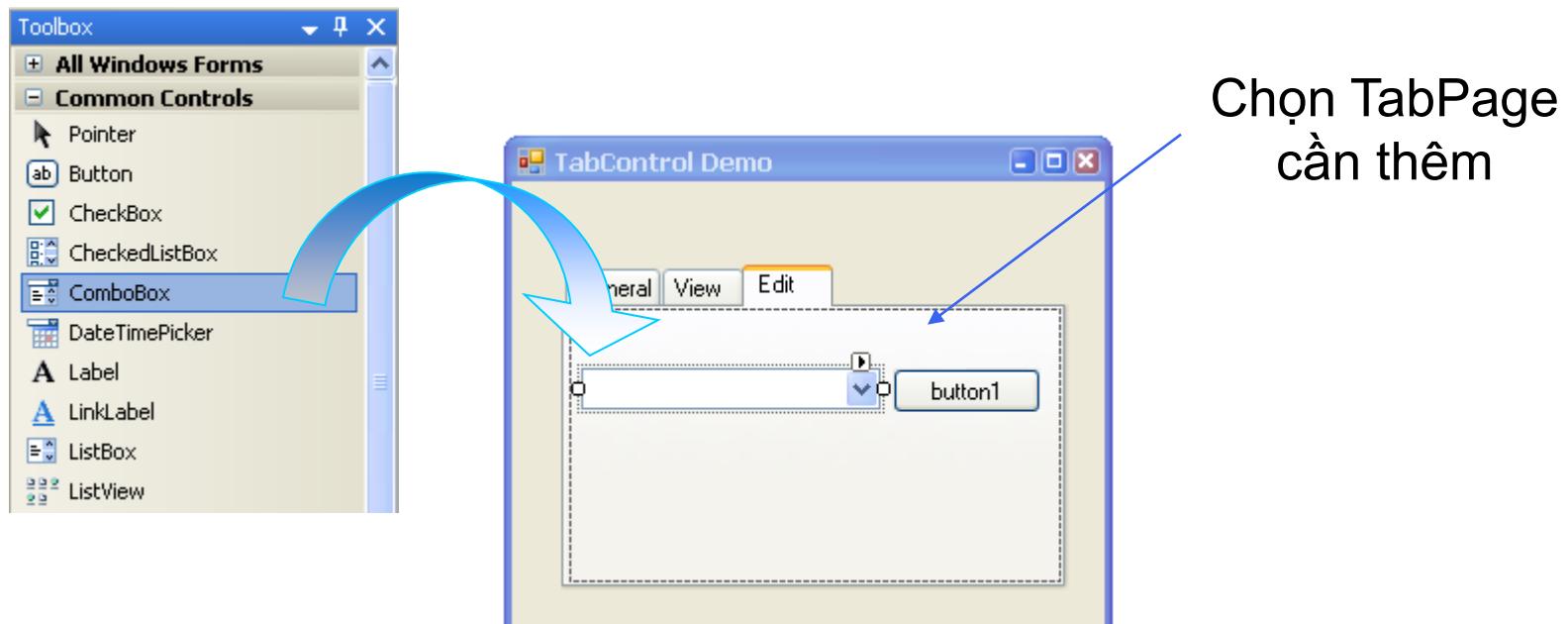
- **Chỉnh sửa các TabPage**
 - Chọn thuộc tính *TabPage*s của *TabControl*
 - Sử dụng màn hình *TabPage Collection Editor* để chỉnh



TabControl

■ **Bổ sung Control vào TabControl**

- Chọn TabPage cần thêm control
- Kéo control từ ToolBox thả vào TabPage đã chọn



TabControl

- Sử dụng code để thêm các TabPage vào TabControl

```
private void AddTabControl()
{
    TabControl tabControl1 = new TabControl();
   TabPage tabPageGeneral = new TabPage("General");
   TabPage tabPageView = new TabPage("View");

    tabControl1.TabPages.Add(tabPageGeneral);
    tabControl1.TabPages.Add(tabPageView);

    tabControl1.Location = new Point(20, 20);

    this.Controls.Add(tabControl1);
}
```



| Bài giảng môn học

PictureBox & ImageList

Lập trình Windows Form với C#

PictureBox

- **Sử dụng để hiển thị ảnh dạng bitmap, metafile, icon, JPEG, GIF.**
- **Sử dụng thuộc tính Image để thiết lập ảnh lúc design hoặc runtime.**
- **Các thuộc tính**
 - Image: ảnh cần hiển thị
 - SizeMode:
 - Normal
 - StretchImage
 - AutoSize
 - CenterImage
 - Zoom

PictureBox



5 pictureBox với
cácSizeMode
tương ứng

ImageList

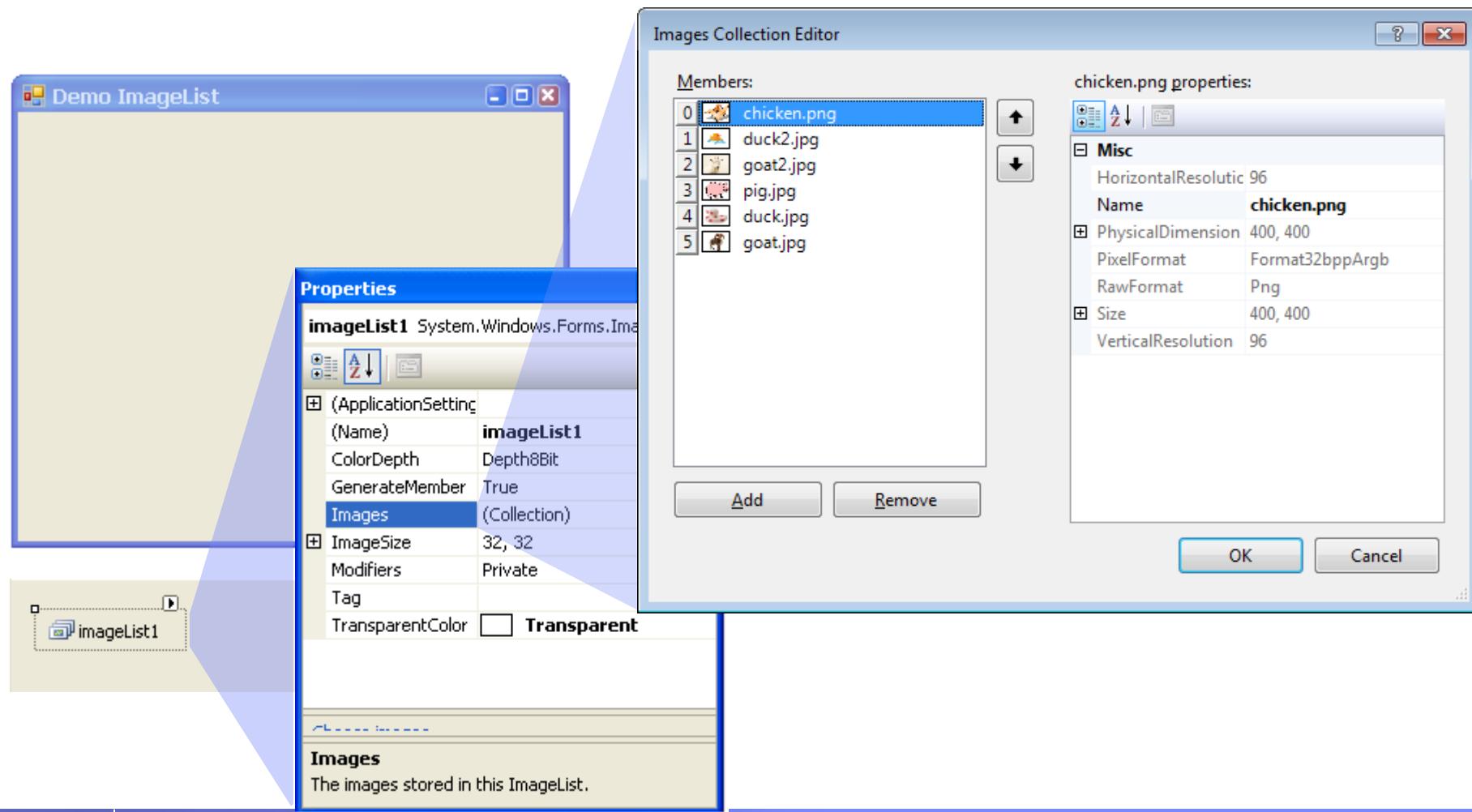
- **Cung cấp tập hợp những đối tượng image cho các control khác sử dụng**
 - ListView
 - TreeView
- **Các thuộc tính thường dùng**
 - **ColorDepth**: độ sâu của màu
 - **Images**: trả về ImageList.ImageCollection
 - **ImageSize**: kích thước ảnh
 - **TransparentColor**: xác định màu là transparent

ImageList

■ Các bước sử dụng ImageList

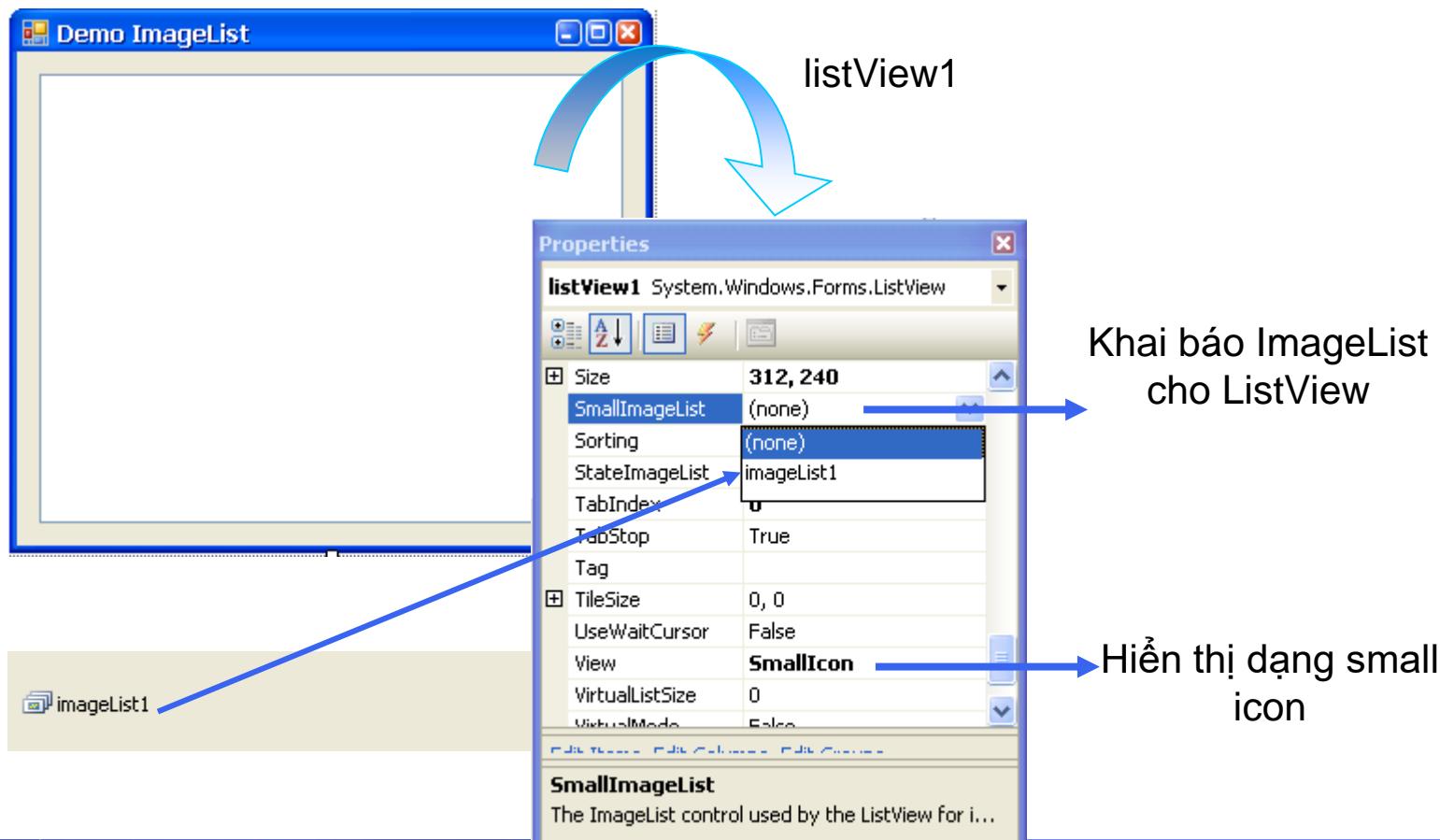
- Kéo control ImageList từ ToolBox thả vào Form
- Thiết lập kích thước của các ảnh: ImageSize
- Bổ sung các ảnh vào ImageList qua thuộc tính Images
- Sử dụng ImageList cho các control
 - Khai báo nguồn image là image list vừa tạo cho control
 - Thường là thuộc tính ImageList
 - Thiết lập các item/node với các ImageIndex tương ứng
 - Việc thiết lập có thể ở màn hình design view hoặc code view

ImageList



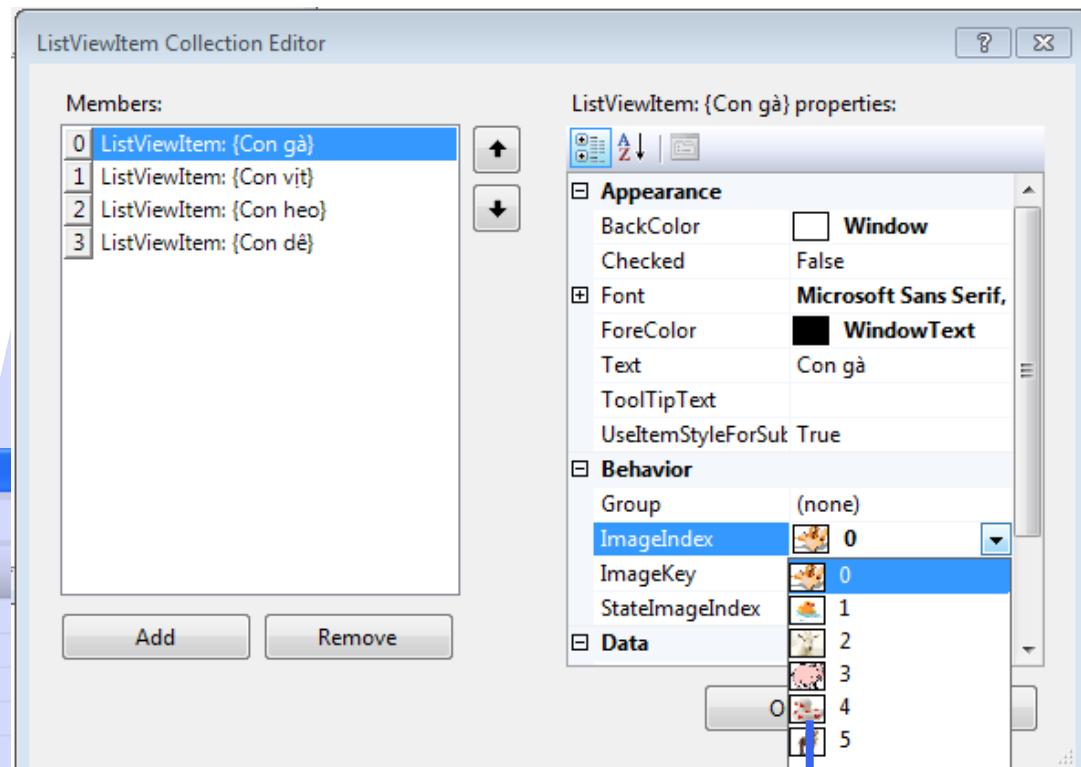
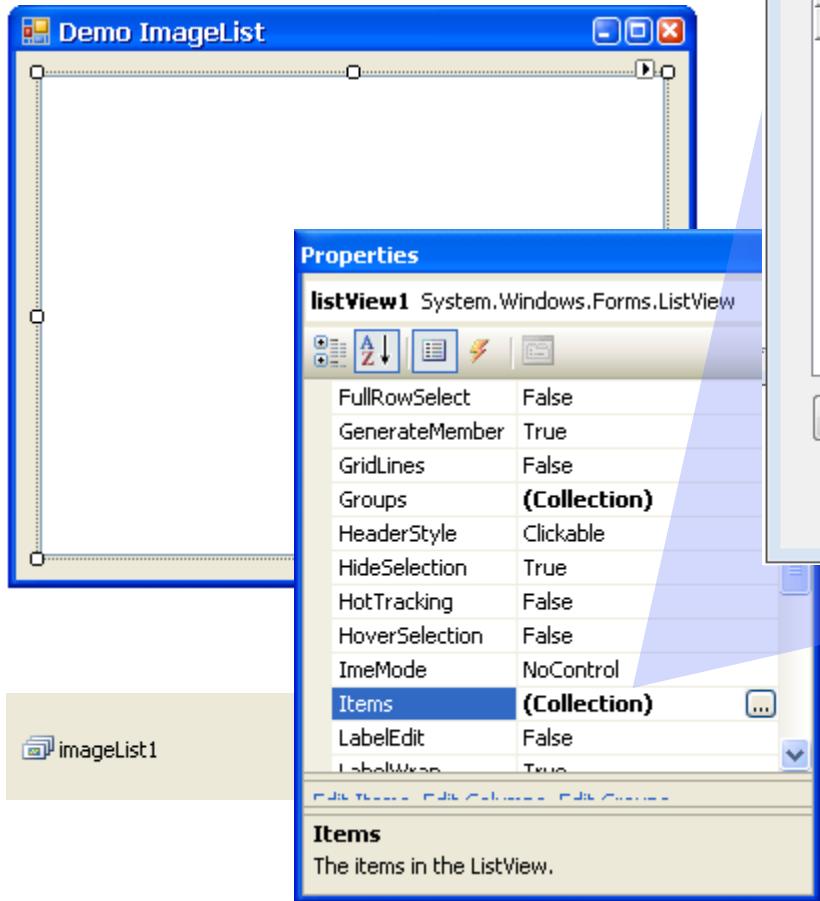
ImageList

Sử dụng ImageList trong ListView



ImageList

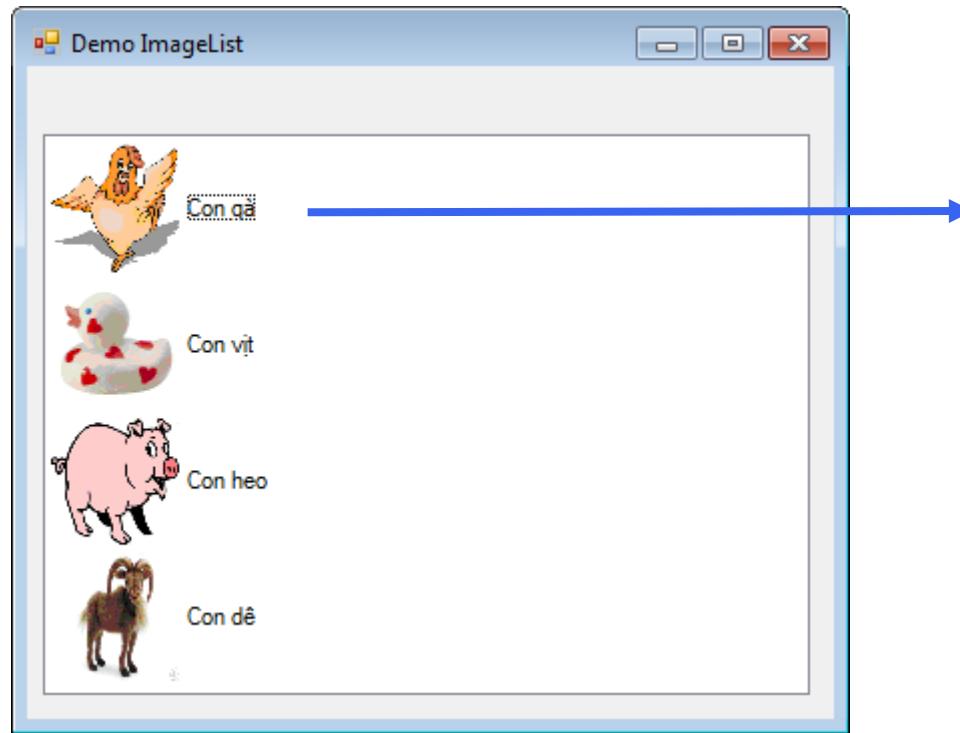
■ Thêm Item



Khai báo image cho
item qua ImageIndex

ImageList

Demo



Mỗi item sẽ có ảnh
theo đúng thứ tự
ImageIndex được khai
báo trong ImageList



| Bài giảng môn học

NumericUpDown & DomainUpDown

Lập trình Windows Form với C#

NumericUpDown

- Cho phép user chọn các giá trị trong khoảng xác định thông qua
 - Nút up & down
 - Nhập trực tiếp giá trị
- Các thuộc tính
 - Minimum
 - Maximum
 - Value
 - Increment
- Sự kiện
 - ValueChanged
- Phương thức
 - DownButton
 - UpButton

NumericUpDown

■ Đoạn code thêm control NumericUpDown

```
public void AddNumericUpDown() {
    NumericUpDown numUpDn = new NumericUpDown();
    numUpDn.Location = new Point(50, 50);
    numUpDn.Size = new Size(100, 25);

    numUpDn.Hexadecimal = true;          // hiển thị dạng hexa
    numUpDn.Minimum = 0;                 // giá trị nhỏ nhất
    numUpDn.Maximum = 255;               // giá trị lớn nhất
    numUpDn.Value = 0xFF;                // giá trị khởi tạo
    numUpDn.Increment = 1;               // bước tăng/giảm

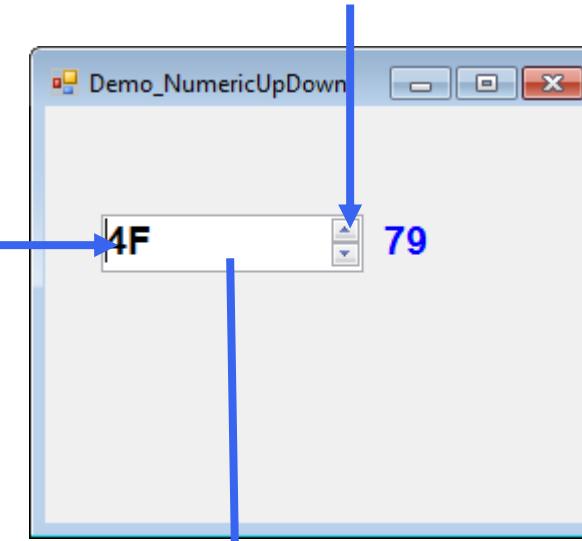
    Controls.Add(numUpDn); // thêm control vào ds control của form
}
```

NumericUpDown

Demo

Hiển thị giá trị
Hexa

Tăng giảm giá trị



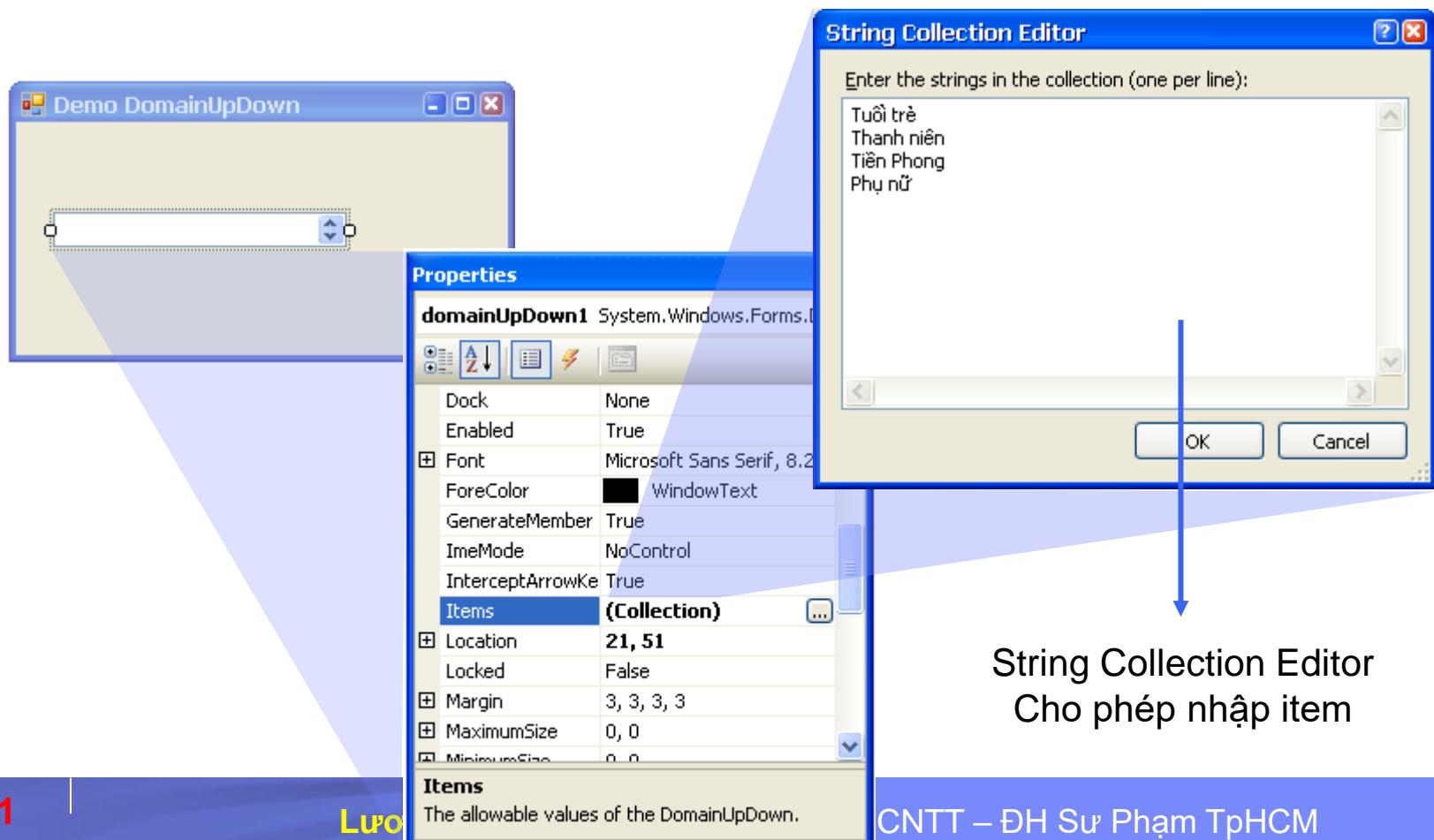
Nhập trực tiếp giá trị

DomainUpDown

- Cho phép user chọn item trong số danh sách item thông qua
 - Button Up & Down
 - Nhập từ bàn phím
- Properties
 - **Items**: danh sách item
 - **ReadOnly**: true chỉ cho phép thay đổi giá trị qua Up & Down
 - **SelectedIndex**: chỉ mục của item đang chọn
 - **SelectedItem**: item đang được chọn
 - **Sorted**: sắp danh sách item
 - **Text**: text đang hiển thị trên DomainUpDown.
- Event
 - **SelectedItemChanged**

DomainUpDown

Nhập item cho DomainUpDown



DatePicker

- Cho phép chọn ngày trong khoảng xác định thông qua giao diện đồ họa dạng calendar
- Kết hợp ComboBox và MonthCalendar
- Properties
 - Format: định dạng hiển thị
 - long, short, time, custom
 - CustomFormat:
 - dd: hiển thị 2 con số của ngày
 - MM: hiển thị 2 con số của tháng
 - yyyy: hiển thị 4 con số của năm
 - ...*(xem thêm MSDN Online)*
 - MaxDate: giá trị ngày lớn nhất
 - MinDate: giá trị ngày nhỏ nhất
 - Value: giá trị ngày hiện tại đang chọn

DatePicker – Custom Format

Format String	Description
ddd	Display three character day of week abbreviation
dddd	Display full day-of week name
hh	Display two-digit hour in 12 hour format
HH	Display two-digit hour in 24 hour format.
mm	Display two-digit minute
MM	Display three-character month abbreviation
MMM	Display full month name
ss	Display two-digit seconds
t	Display the first letter of the AM/PM abbreviation
tt	Display two-letter AM/PM abbreviation
yy	Display last two-digit of the year
yyyy	Display full year name

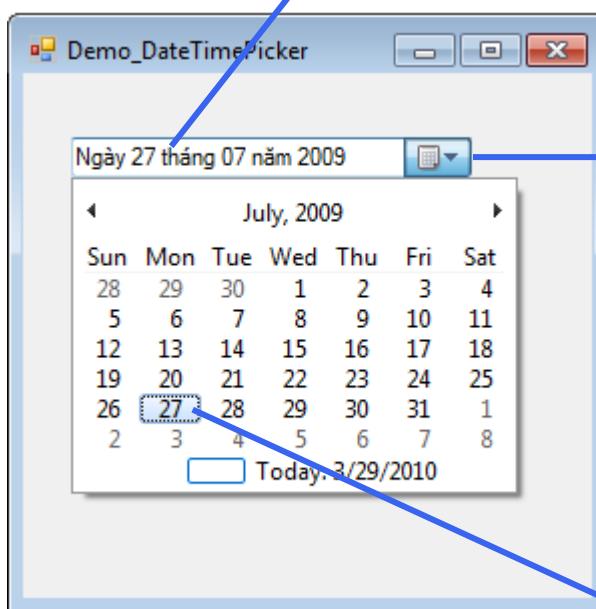
DatePicker

```
private void AddDatePicker() {  
    DateTimePicker DTPicker = new DateTimePicker();  
  
    DTPicker.Location = new Point(40, 80);  
    DTPicker.Size = new Size(160, 20);  
  
    DTPicker.DropDownAlign = LeftRightAlignment.Right;  
  
    DTPicker.Value = DateTime.Now;  
    DTPicker.Format = DateTimePickerFormat.Custom;  
    DTPicker.CustomFormat = "'Ngày' dd 'tháng' MM 'năm' yyyy";  
  
    this.Controls.Add(DTPicker);  
}
```

DatePicker

Demo

Định dạng xuất: 'Ngày' dd 'tháng' MM 'năm' yyyy



Kích drop down
để hiện thị hộp
chọn ngày

Chọn ngày trong
khoảng cho trước

MonthCalendar

- Cho phép user chọn một ngày trong tháng hoặc nhiều ngày với ngày bắt đầu và ngày kết thúc.
- Một số thuộc tính thông dụng
 - MaxDate, MinDate
 - SelectionStart: ngày bắt đầu chọn
 - SelectionEnd: ngày kết thúc
- Sinh viên tự tìm hiểu thêm...



RichTextBox

- **Chức năng mở rộng từ TextBox, có thể hiển thị text dạng rich text format (RTF)**
- **Các text có thể có các font chữ và màu sắc khác nhau.**
- **Đoạn text có thể được canh lề**
- **Có thể chứa các ảnh**
- **Ứng dụng WordPad là dạng RichTextBox**

- ***Sinh viên tự tìm hiểu thêm...***

RichTextBox control

Property

Description

Font	Specifies or retrieves the font of the text displayed in the control.
ScrollBars	Specifies or retrieves the type of scroll bars for the control.
SelectedText	Specifies or retrieves the selected text in the control.
SelectionFont	Specifies or retrieves the font of the selected text or at the insertion point.
SelectionLength	Specifies or retrieves the number of characters selected in the control.
Text	Specifies or retrieves the text in the control.
WordWrap	Determines whether the control automatically wraps the words towards the beginning of the next line whenever necessary.

RichTextBox control

Method	Description
AppendText	Adds text to the current text of a RichTextBox control.
Copy	Copies the selected text from the control to the clipboard.
Paste	Pastes the contents of the clipboard into the control.
Redo	Performs the last operation again that was undone in the control.
Undo	Undoes the last edit operation.

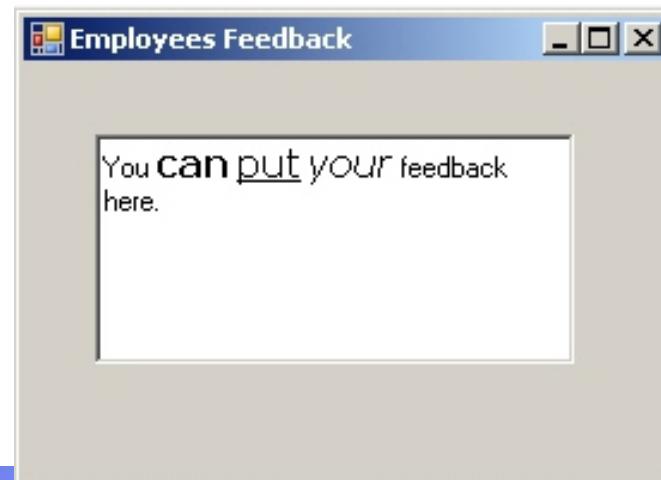
Event	Description
Click	Occurs when the control is clicked.
HScroll	Occurs when the user clicks the horizontal scroll bar of the control.
VScroll	Occurs when the user clicks the vertical scroll bar of the control.

RichTextBox control

The following code demonstrates the creation of a RichTextBox control.

```
RichTextBox rtbFeedback = new RichTextBox();
rtbFeedback.SelectedText = " This is a sample ";
rtbFeedback.SelectionColor = Color.Red;
rtbFeedback.SelectedText = " RichTextBox control ";
rtbFeedback.WordWrap = false;

private void rtbFeedback_TextChanged(object sender, EventArgs e)
{
    rtbFeedback.WordWrap = true;
}
```

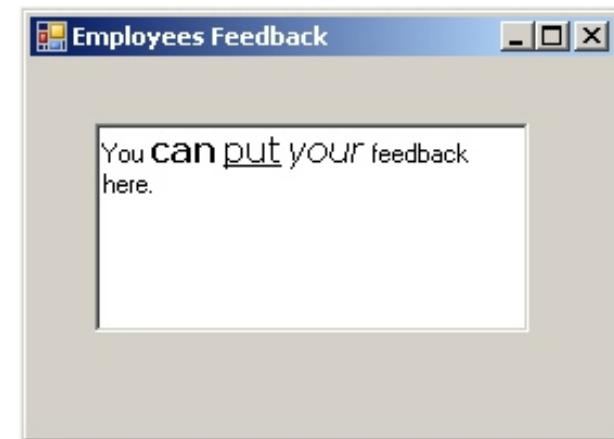


Font and FontStyle classes

- You can specify the font, font style and font size for the text using the **SelectionFont** property and **Font** class (exists in the **System.Drawing** namespace)
- The **FontStyle** enumeration is used to specify of the font such as **Bold, Italics and Underline**

The following code demonstrates how to apply a font, font style, and font size to the text in the RichTextBox control.

```
RichTextBox rtbFeedback = new RichTextBox();
rtbFeedback.Text = "You can put your feedback here.";
rtbFeedback.Select(4, 3);
rtbFeedback.SelectionFont = new Font("Verdana", 12,
FontStyle.Bold);
rtbFeedback.Select(8, 3);
rtbFeedback.SelectionFont = new Font("Verdana", 12,
FontStyle.Underline);
rtbFeedback.Select(12, 4);
rtbFeedback.SelectionFont = new Font("Verdana", 12,
FontStyle.Italic);
```



RichTextBox

Menu Strip, Tool Strip



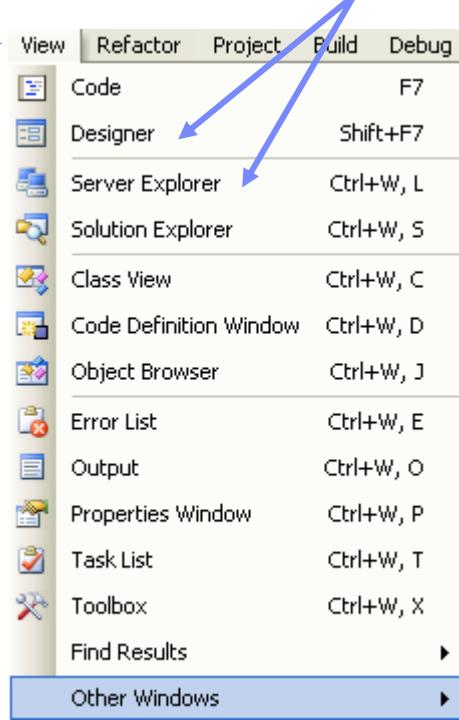
Lập trình
với C#

Menu Strip

- **Menu cung cấp nhóm lệnh có quan hệ với nhau cho các ứng dụng Windows**

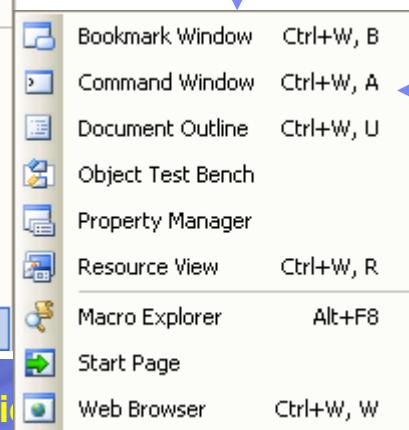
Menu item

Menu



Namespace: **System.Windows.Forms**

Submenu

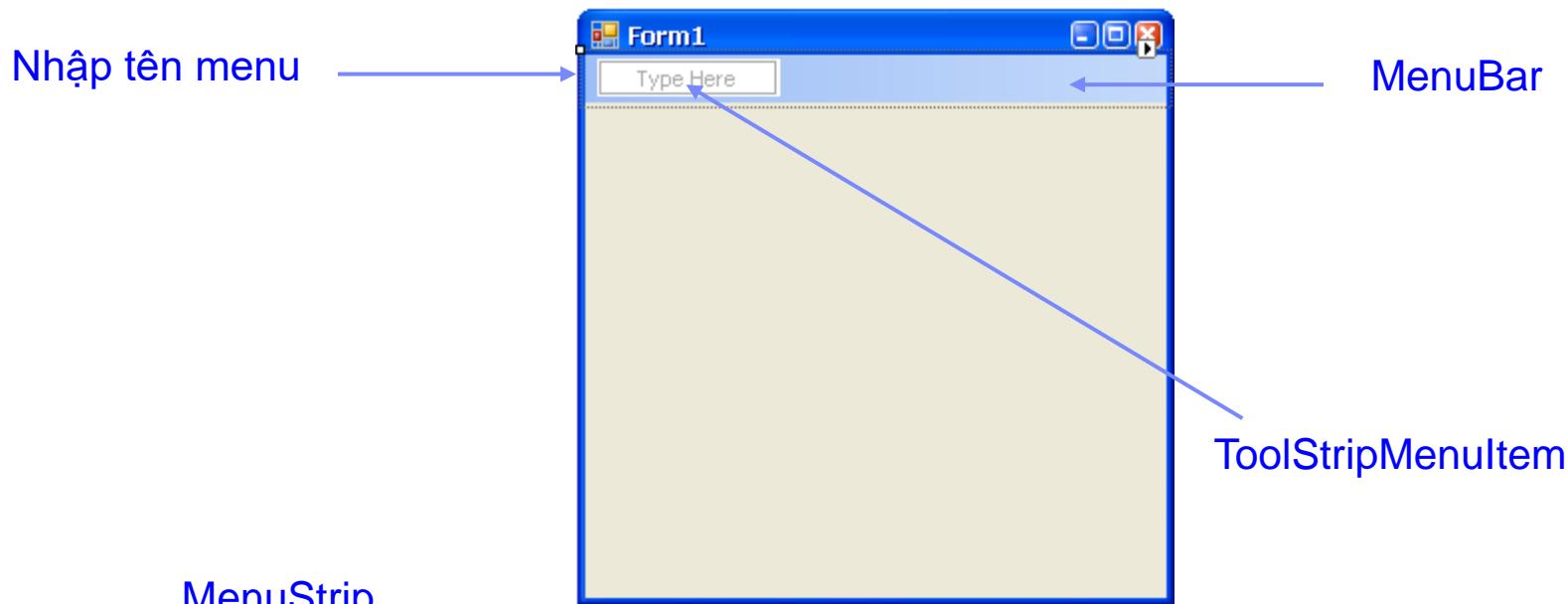


Shortcut key

MenuStrip

■ Cách tạo menu

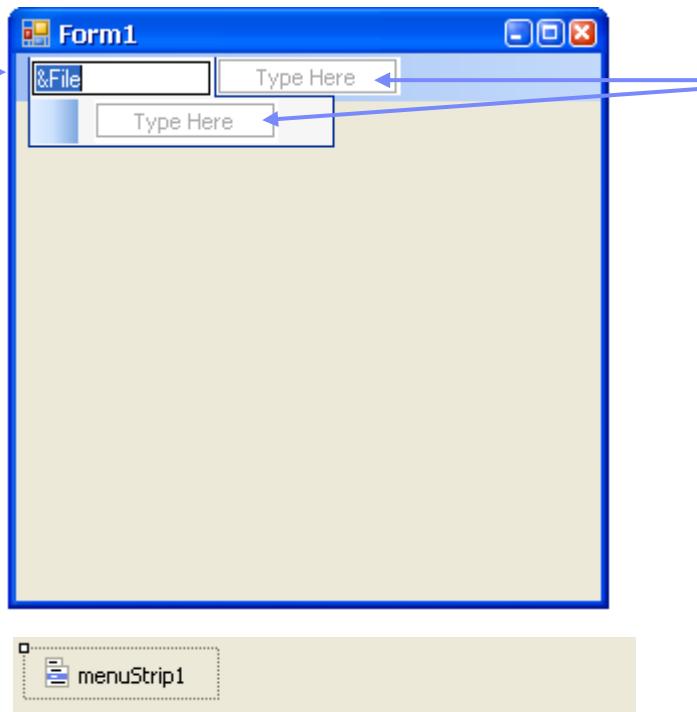
- Trong ToolBox kéo control MenuStrip thả vào form
- Thanh menuBar xuất hiện trên cùng của form
- Trong màn hình design dễ dàng tạo lập các menu item



MenuStrip

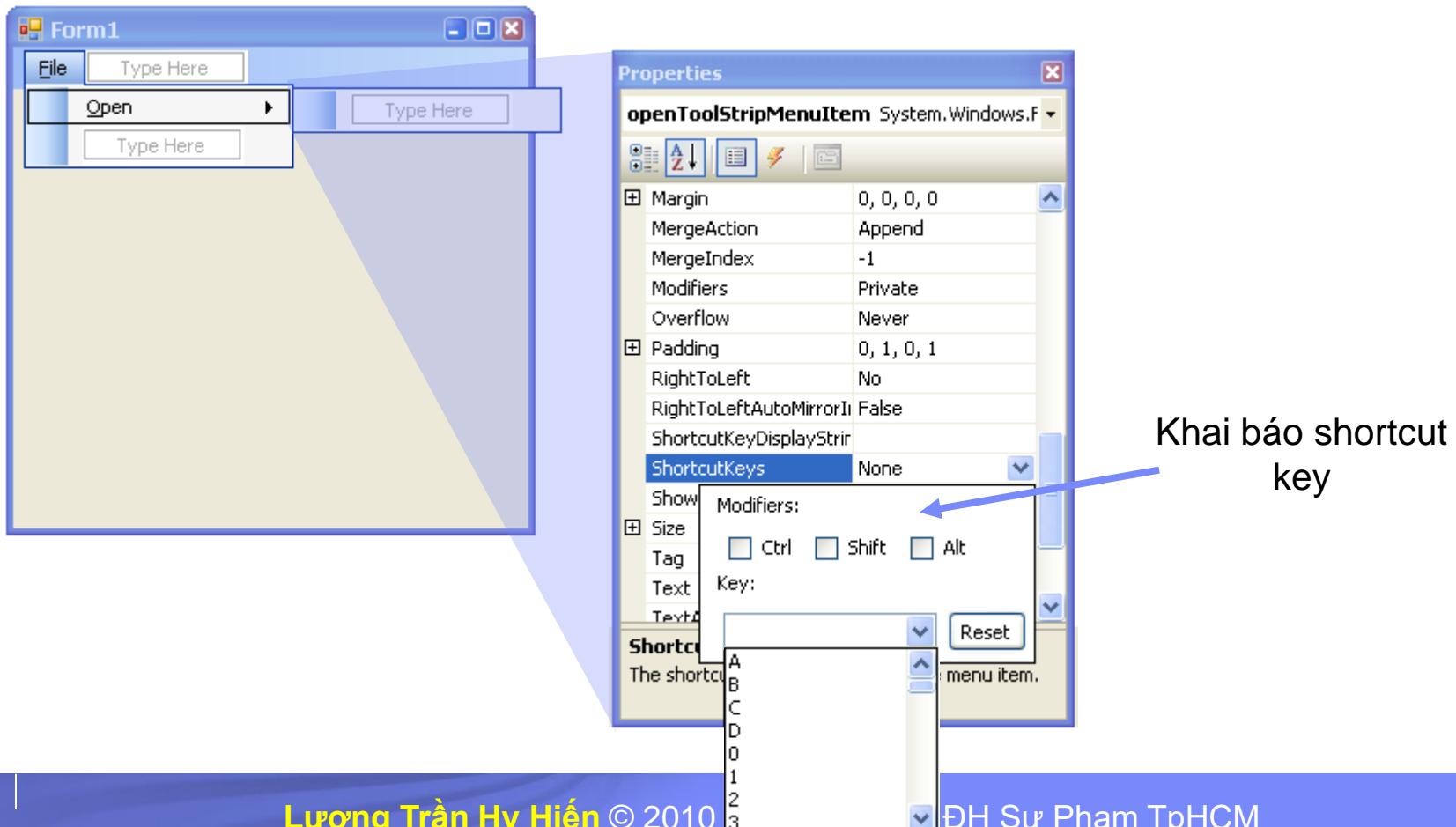
Đặt ký tự &
trước ký tự
làm phím tắt

Tạo menu item
mới bằng cách
nhập vào textbox



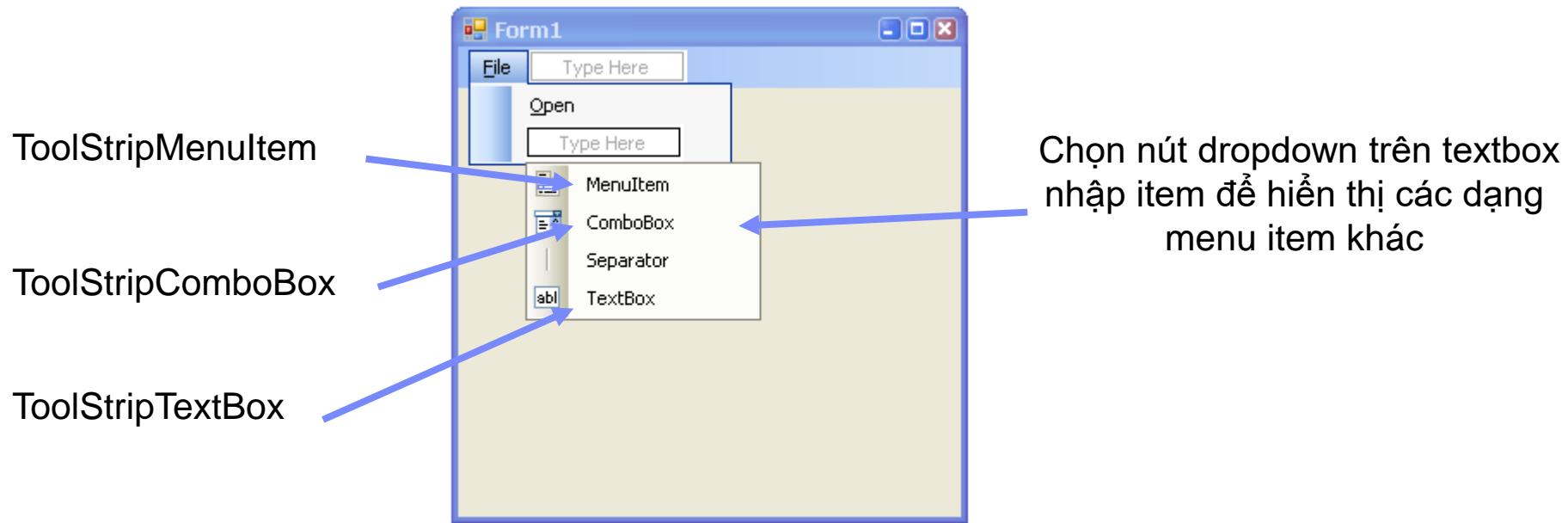
Menu

- Thiết lập Shortcut Key cho menu item



MenuStrip

- Menu item có thể là TextBox hoặc ComboBox



MenuStrip

Thuộc tính của MenuStrip, ToolStripMenuItem

MenuStrip

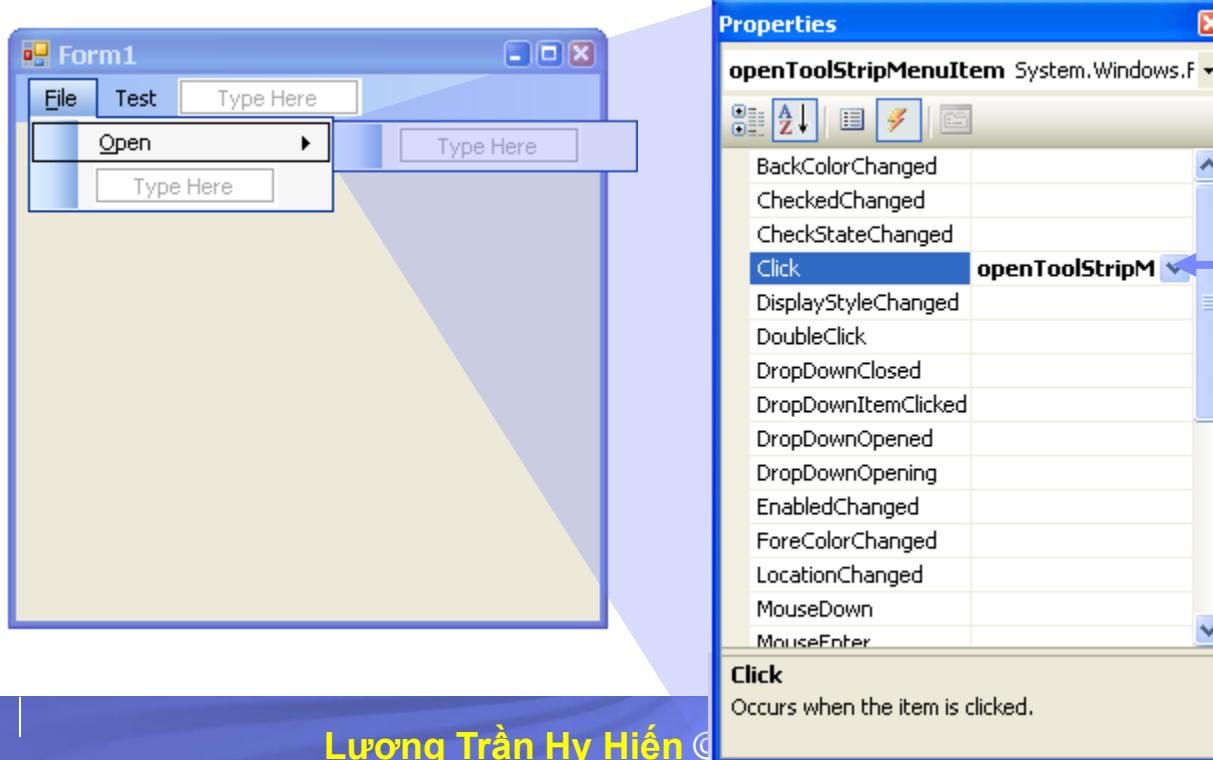
Items	Chứa những top menu item
MdiWindowListItem	Chọn top menu item hiển thị tên các cửa sổ con

ToolStripMenuItem

Checked	Xác định trạng thái check của menu item
Index	Chỉ mục menu item trong menu cha
DropDownItems	Chứa những menu item con
ShortcutKeys	Phím tắt
Text	Tiêu đề menu item
ShowShortcutKeys	Xác định trạng thái hiện thị phím tắt bên cạnh menu item

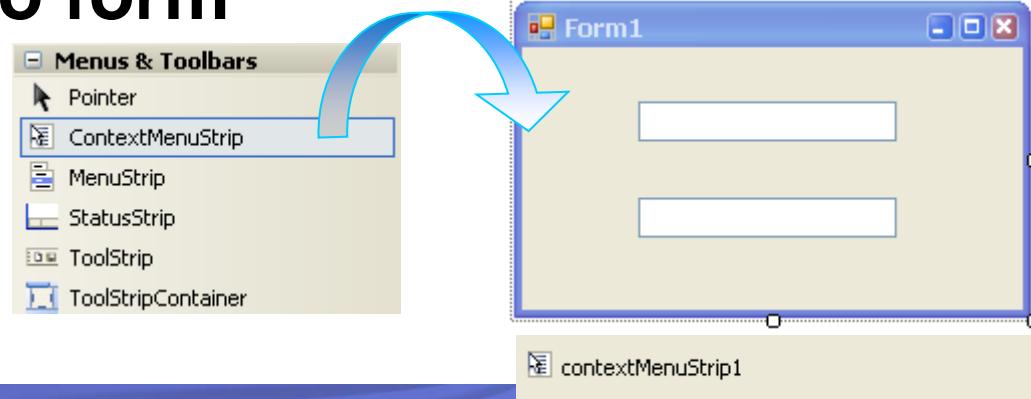
MenuStrip

- Sự kiện cho Menu Item thường dùng là Click
- Trong màn hình Design View kích đúp vào menu item VS sẽ tự động tạo trình xử lý cho menu item



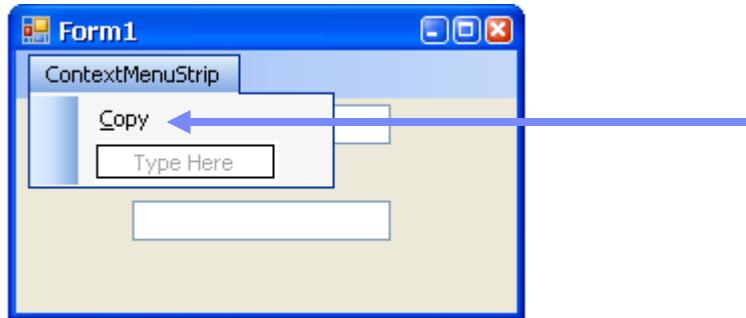
Context Menu

- Xuất hiện khi user kích chuột phải
- Thông thường menu này xuất hiện tùy thuộc vào đối tượng trong vùng kích chuột phải.
- Trong ToolBox kéo ContextMenuStrip thả vào form



Context Menu

- Kích vào ContextMenuStrip để soạn thảo các menuitem
- ContextMenuStrip tạm thời thể hiện trên cùng của form
 - Khi run thì sẽ không hiển thị cho đến khi được gọi



Soạn thảo Context Menu
tương tự như Menu bình
thường

Context Menu

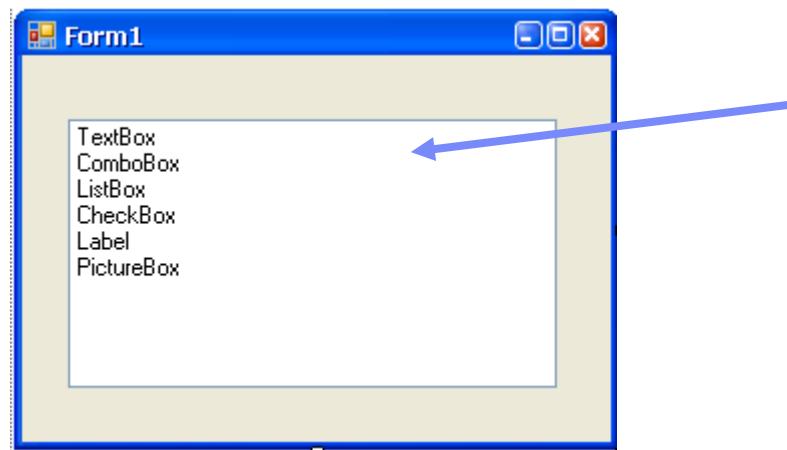
- Khai báo sử dụng Context Menu
- Mỗi control đều có property là:

ContextMenuStrip

- Khai báo thuộc tính này với ContextMenuStrip
 - Khi đó user kích chuột phải lên control thì sẽ hiển thị context Menu đã cài đặt sẵn
- Khai báo trình xử lý sự kiện Click cho ContextMenu
 - Kích đúp vào menu item của Context Menu để tạo
 - Hoặc trong cửa sổ Properties -> Event kích đúp vào sự kiện Click.

Context Menu

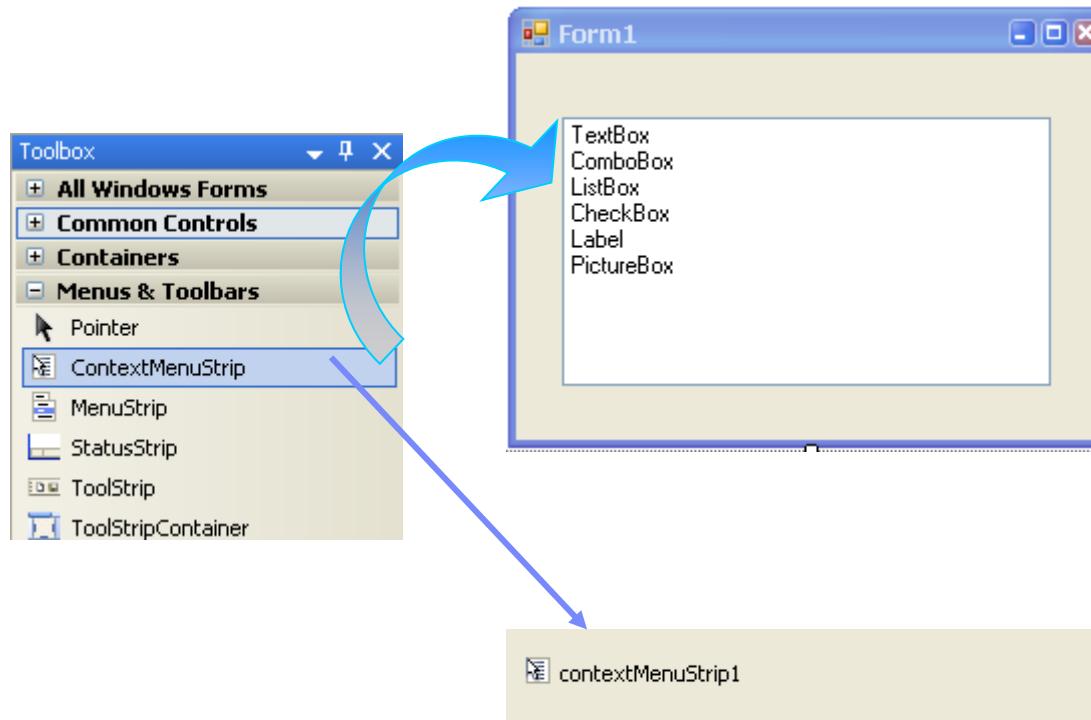
- **Demo: tạo context Menu hiển thị trong ListBox có menu item Remove, cho phép xóa item đang được chọn.**
- **Tạo Form có mô tả như sau**



ListBox hiển thị các item

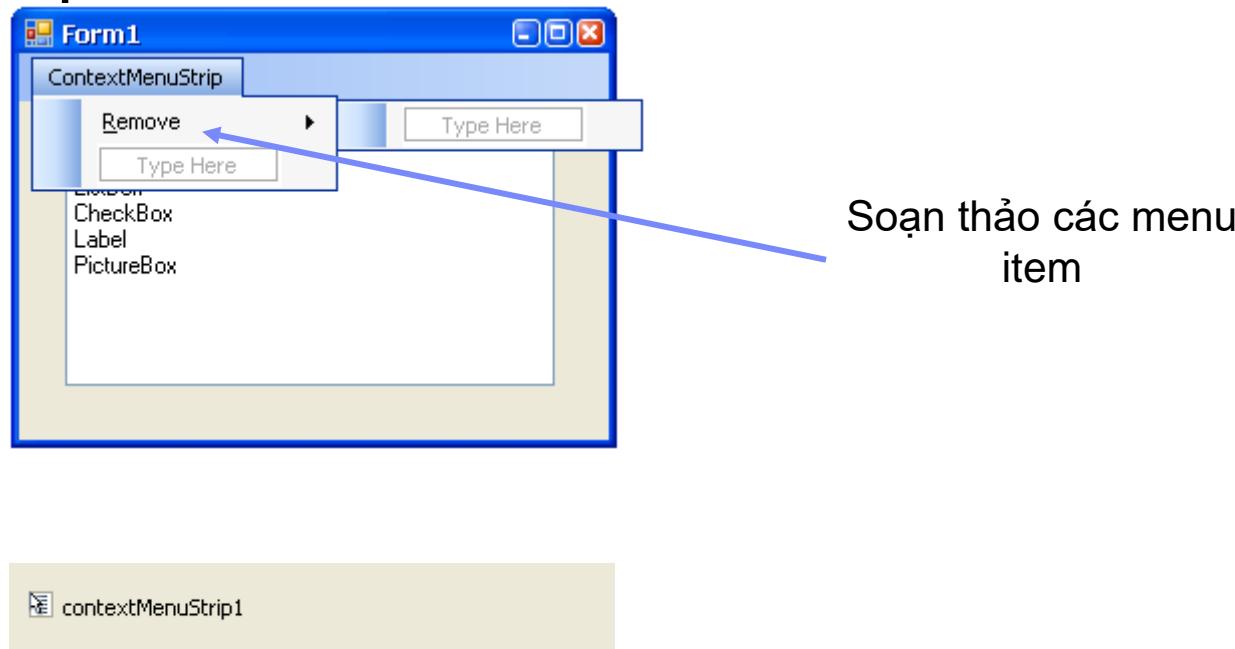
Context Menu

- Kéo ContextMenuStrip thả vào Form



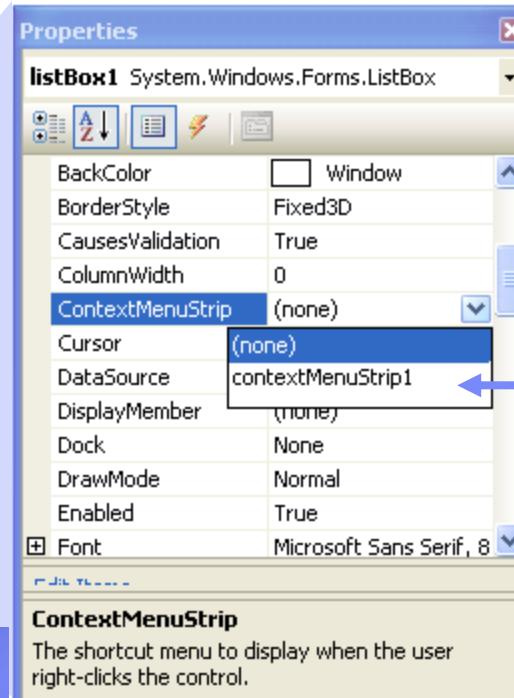
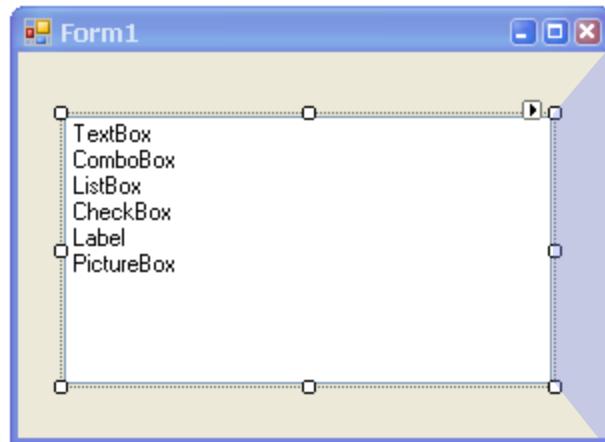
Context Menu

- Kích vào ContextMenuStrip để thiết kế menu
 - Tạo một menu item “Remove” như hình mô tả



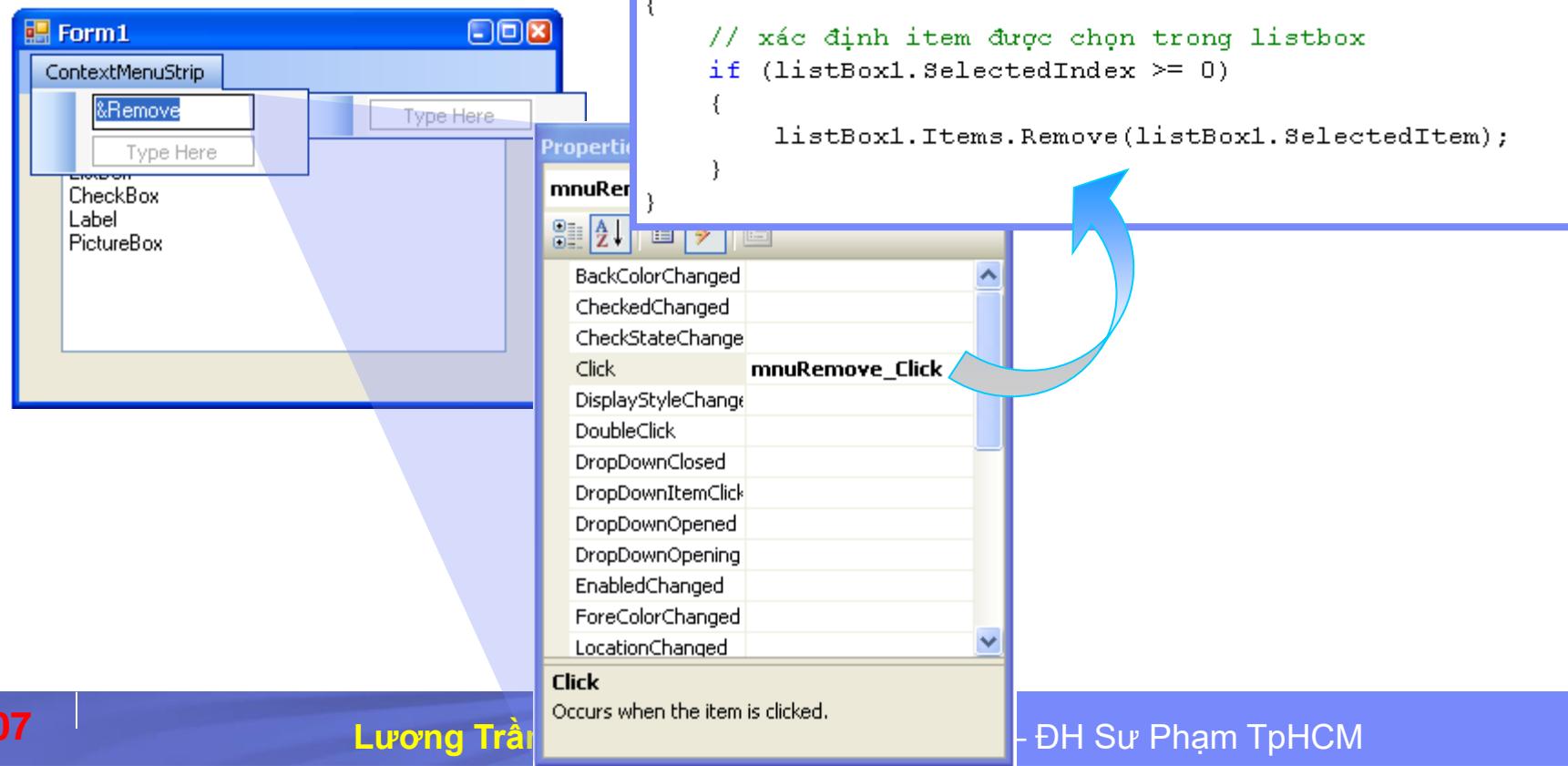
Context Menu

- Liên kết ContextMenu với ListBox
 - Trong cửa sổ properties của ListBox
 - Khai báo thuộc tính ContextMenuStrip = ContextMenuStrip1



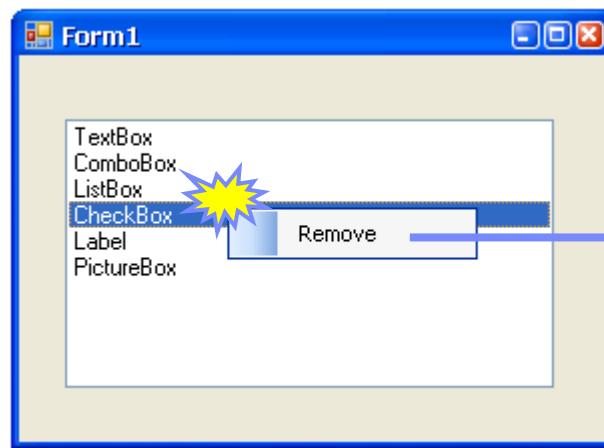
Context Menu

- Khai báo trình xử lý sự kiện Click cho menu item



Context Menu

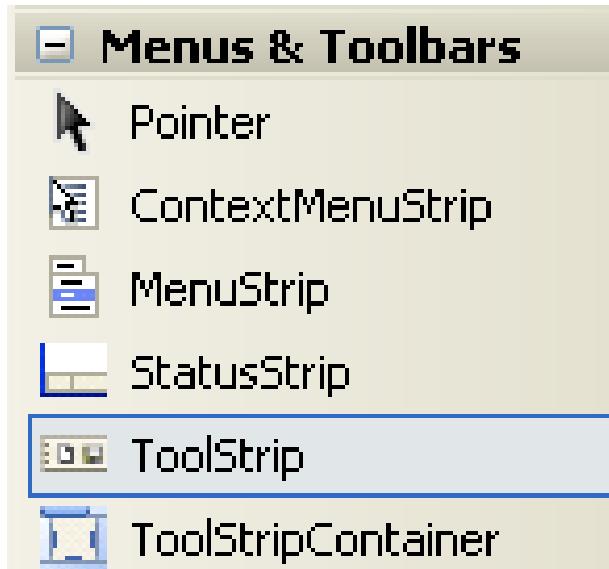
■ Demo



Context Menu hiển thị khi user kích chuột phải lên ListBox

ToolStrip

- Namespace: **System.Windows.Forms**
- Dạng ToolBar chứa nhiều các controls khác nhau
- Xuất hiện ngay dưới thanh Menu



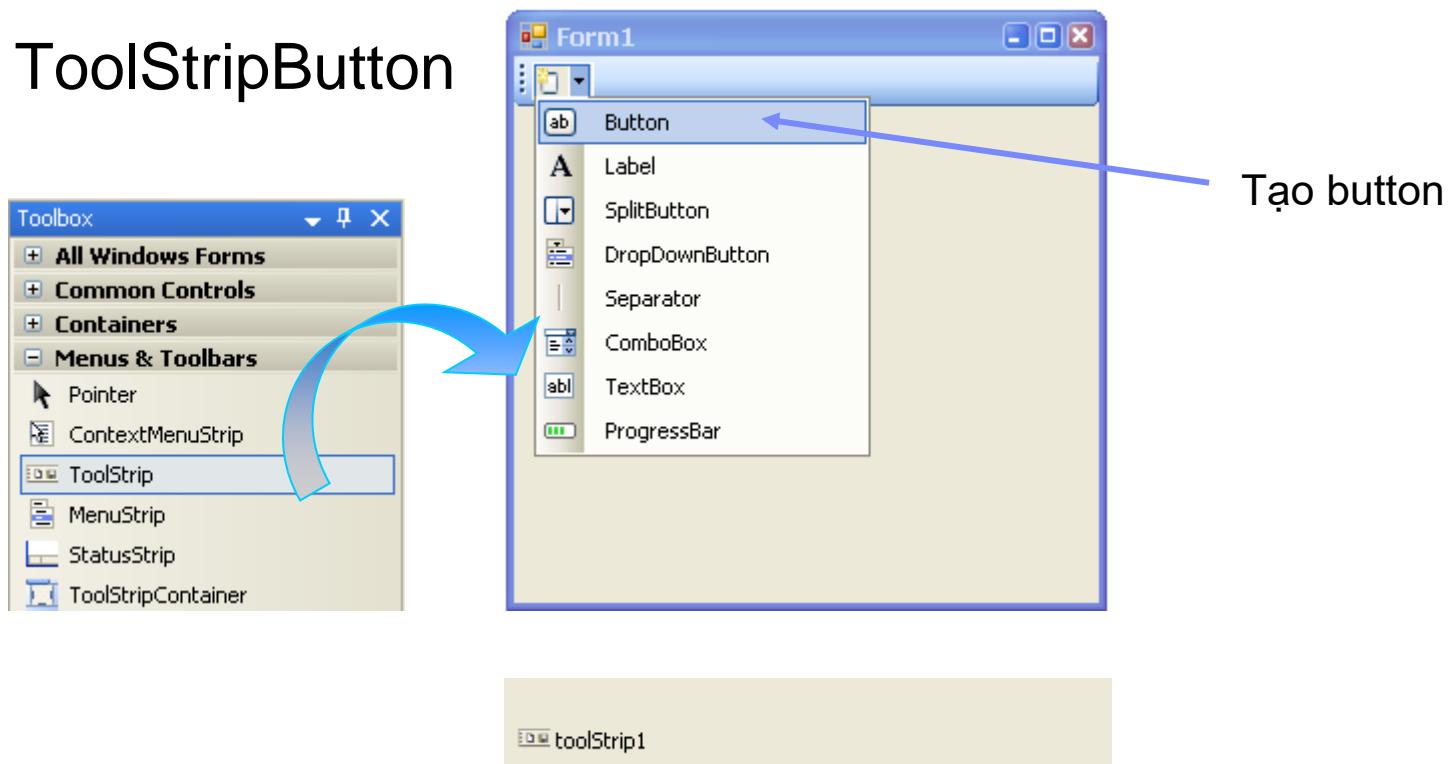
ToolStrip
Version 2.0.0.0 from Microsoft Corporation
.NET Component

Provides toolbars and other user interface elements that support many appearance options, and that support overflow and run-time item reordering.

ToolStrip

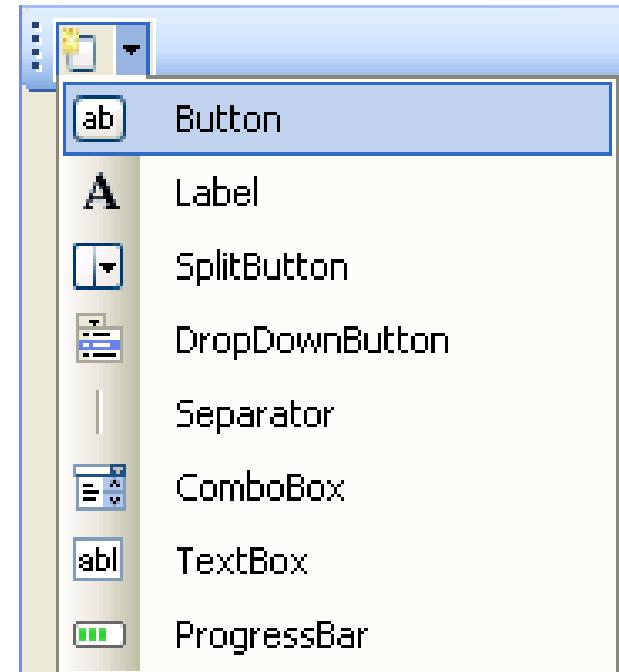
■ Cách tạo button trên ToolStrip

- Kéo ToolStrip thả vào form
- Add ToolStripButton



ToolStrip

- **ToolStripMenuItem là 1 Menu Item của Menu Strip**
 - Một số thuộc tính cơ bản
 - Text
 - Image
 - Checked
 - DropDownItems
 - Một số sự kiện cơ bản
 - Click

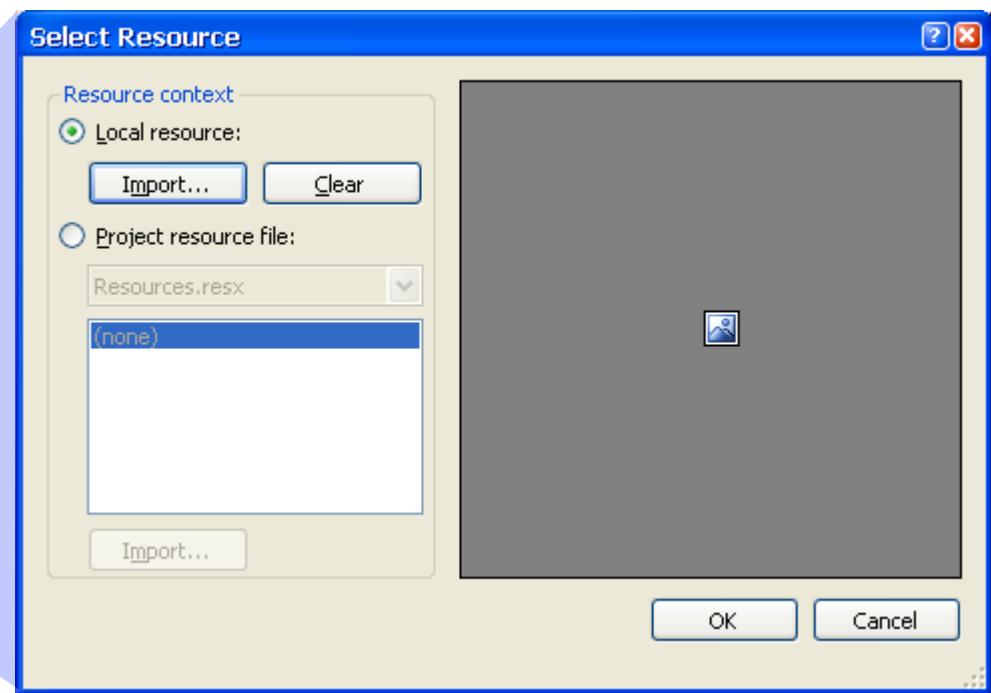
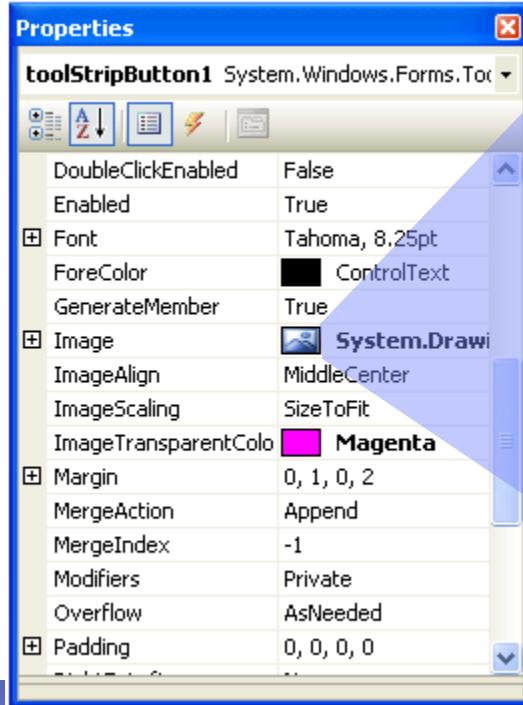


ToolStrip

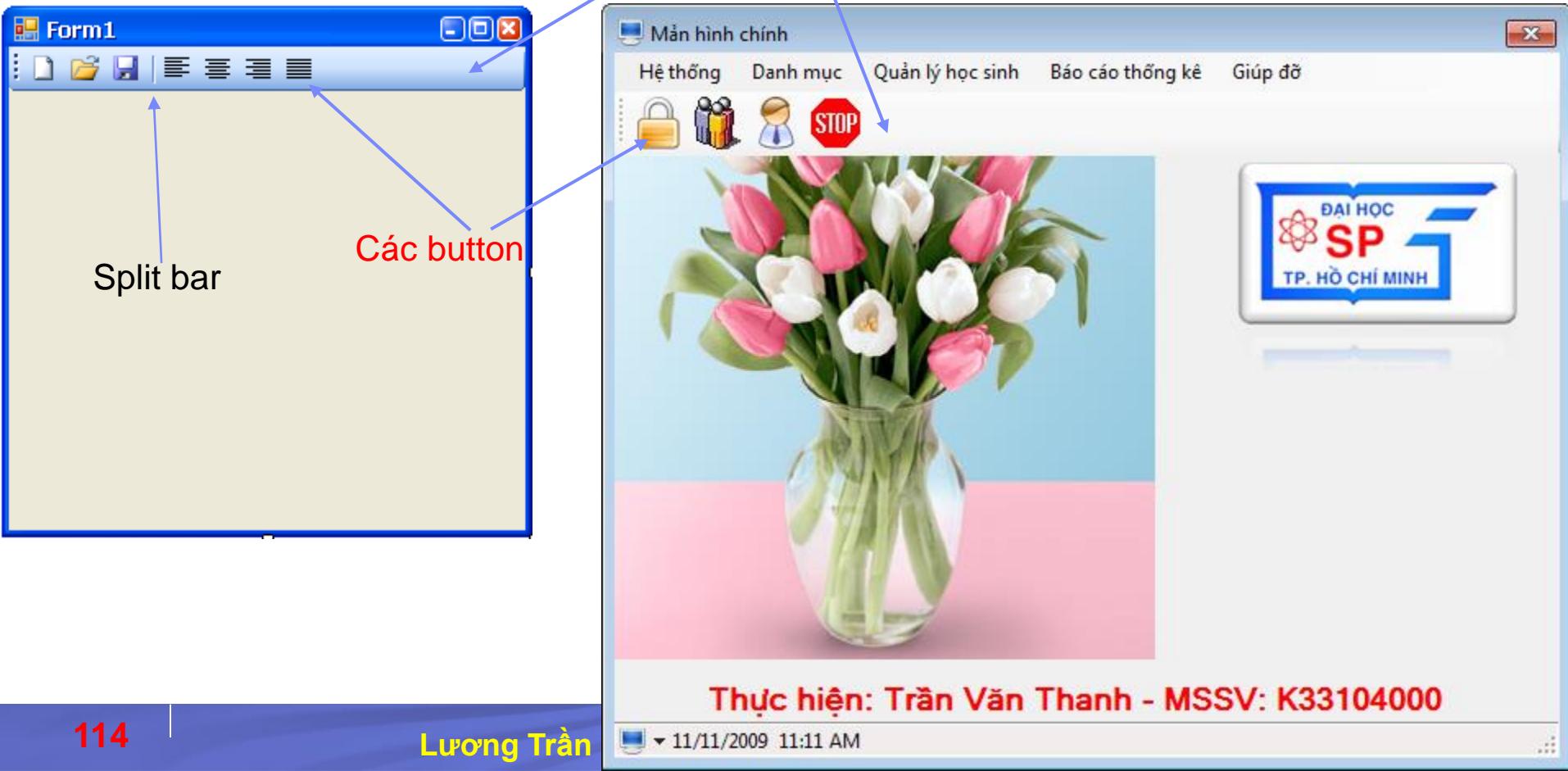
- **Mỗi button có thể là**
 - ToolStripButton
 - ToolStripSplitButton
 - ToolStripDropDownButton
 - ToolStripComboBox
 - ToolStripTextBox
 - ToolStripSeperator
 - ToolStripProgressBar
- **Thuộc tính cơ bản**
 - Items

ToolStrip

- **Bổ sung image cho button**
 - Sử dụng thuộc tính Images để thiết lập



ToolStrip



ToolStrip

- **Khai báo trình xử lý sự kiện Click cho ToolStripButton**
 - Khai báo tương tự như các button
 - Kích đúp vào button trong Design View
 - Thông thường các button là các chức năng thường sử dụng chứa trong menu
 - VD: các button New, Open, Save...
 - Do đó có thể khai báo cùng trình xử lý sự kiện cho các button và menu item cùng chức năng.
 - VD: Menu item “Open” cùng trình xử lý với ToolStripButton “Open”



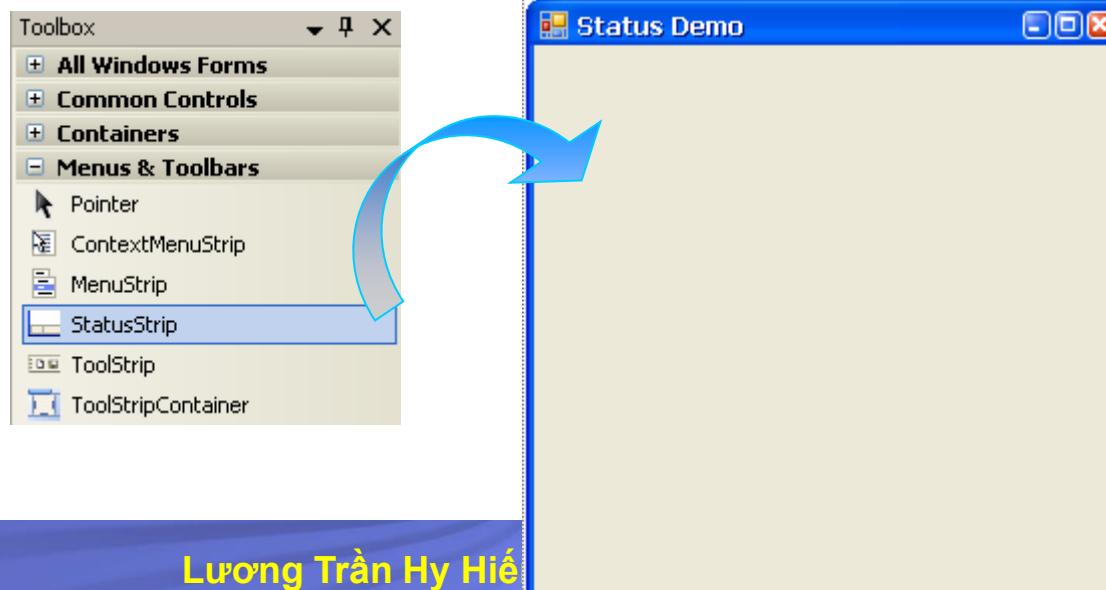
| Bài giảng môn học

StatusStrip

Lập trình Windows Form với C#

StatusStrip

- **Hiển thị thông tin trạng thái của ứng dụng**
- **Nằm bên dưới cùng của Form.**
- **Các lớp liên quan**
 - StatusStrip: là container chứa control khác
 - ToolStripStatusLabel: control có thể add vào StatusStrip

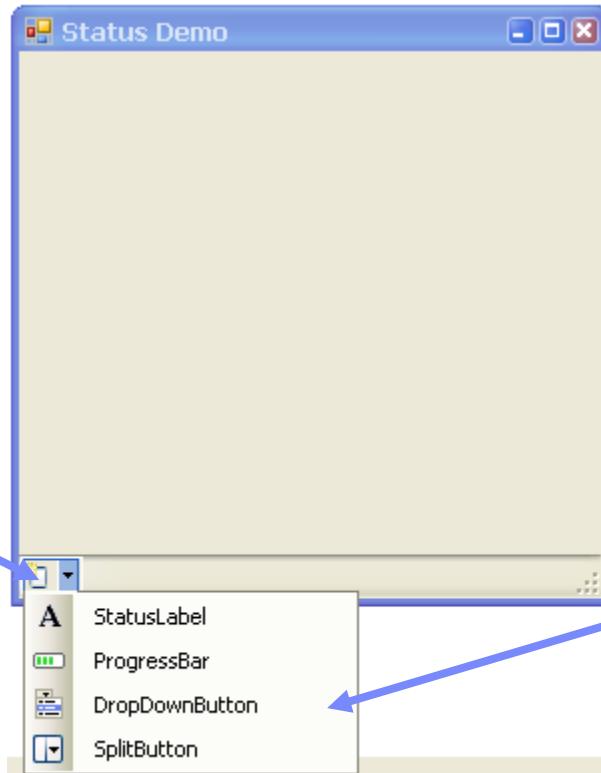


StatusStrip

■ Tạo các item cho StatusStrip

Tạo các item cho
StatusStrip

Các kiểu control cho
StatusStrip



StatusStrip

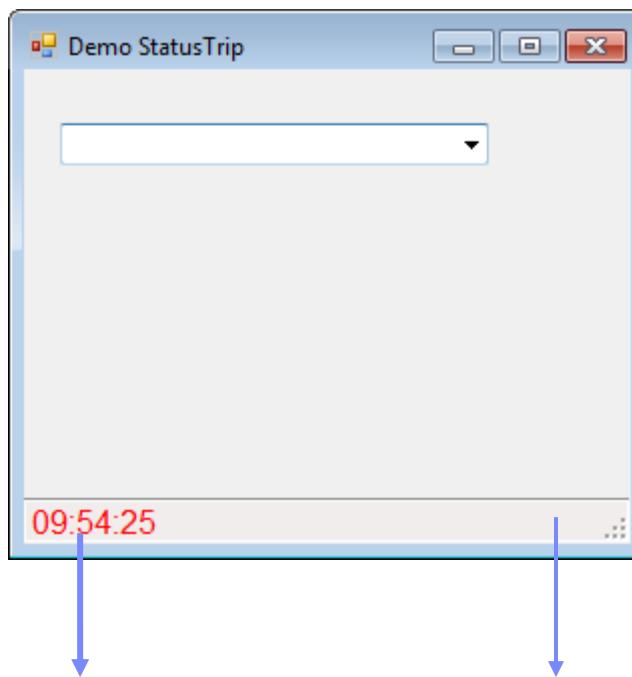
- **Ví dụ tạo sự kiện Tick của Timer cứ mỗi giây kích hoạt và hiển thị giờ trên StatusStrip**
 - Kéo Timer thả vào Form

```
private void timer1_Tick(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text =
        DateTime.Now.ToString("hh:mm:ss");
}
```

Item dạng Label của StatusStrip

StatusStrip

■ Demo

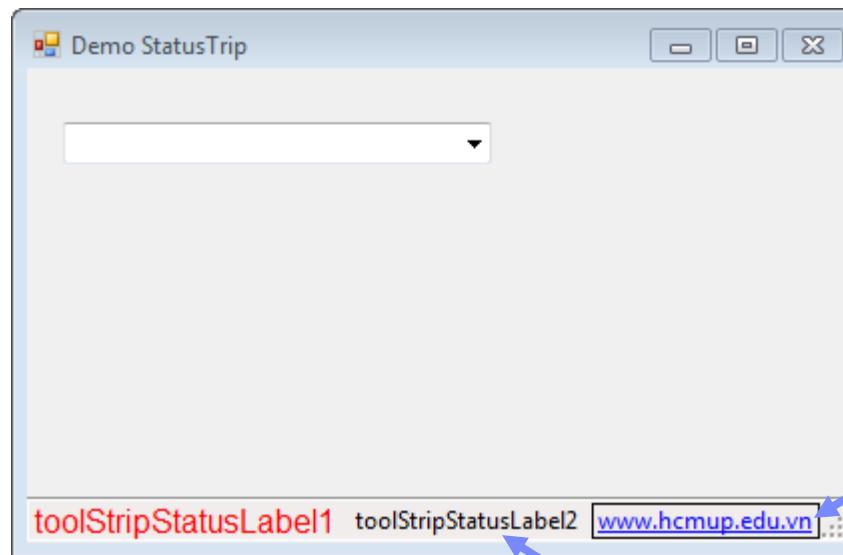


ToolStripStatusLabel

StatusStrip

StatusStrip

- VD: bổ sung hiển thị một hyperlink trên StatusStrip và open hyperlink khi user kích chuột



Label hiển thị hyperlink
Thuộc tính **isLink = true**

Thiết lập với thuộc tính **spring = true**, **Text = “ ”**

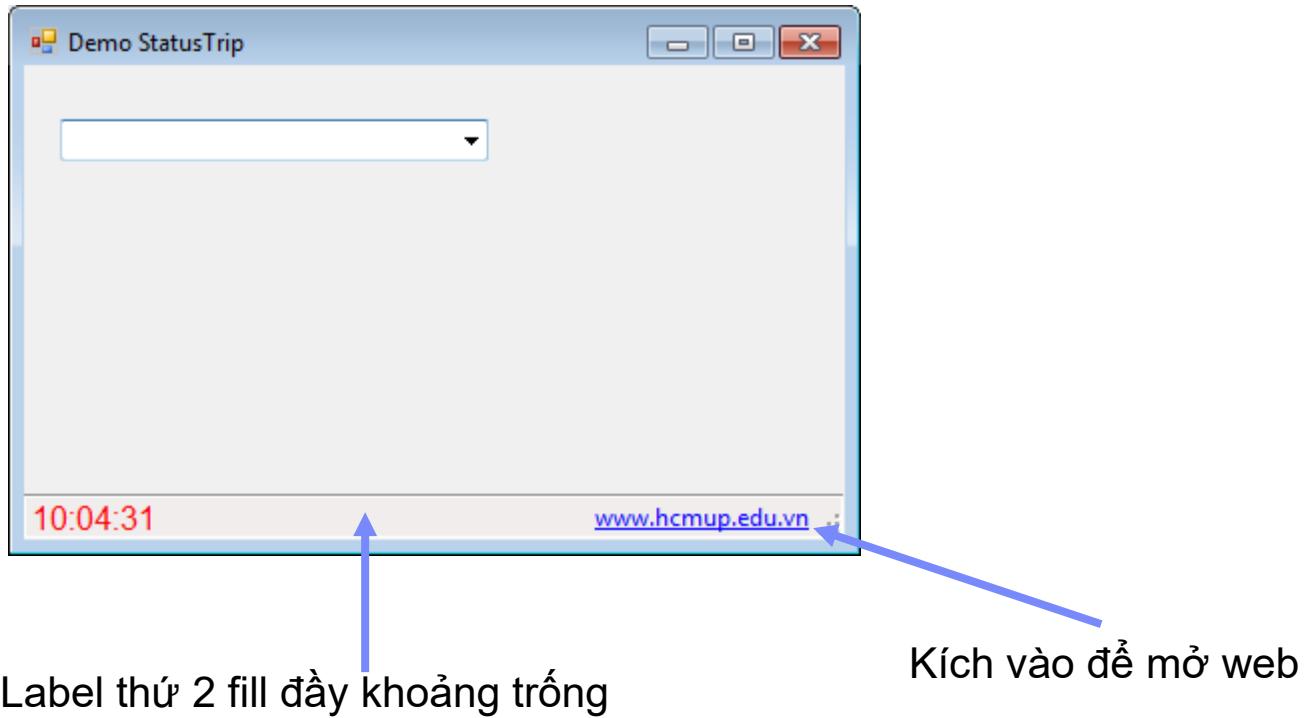
StatusStrip

- **Tạo trình xử lý sự kiện khi user kích vào item Label thứ 3 chứa hyperlink**
 - Trong màn hình design kích đúp vào item thứ 3, VS.NET sẽ phát sinh ra trình xử lý sự kiện

```
private void toolStripStatusLabel3_Click(object sender, EventArgs e)
{
    //Sử dụng Process để mở IE
    //Sử dụng namespace: System.Diagnostics
    ProcessStartInfo startInfo = new ProcessStartInfo();
    startInfo.FileName = "iexplore.exe";
    startInfo.Arguments = "http://hcmup.edu.vn";
    Process.Start(startInfo);
}
```

StatusStrip

- Demo





| Bài giảng môn học

Advanced Controls

| www.hienlth.info

Lập trình Windows Form với C#

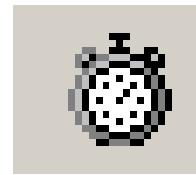
Timer

- Bộ định thời gian, thiết lập một khoảng thời gian xác định (interval) và khi hết khoảng thời gian đó Timer sẽ phát sinh sự kiện tick.



Methods

Start



Properties

Enabled

Stop

Events

Interval



Tick

Timer

- **Multitasking**
- **Quản lý và thông báo các trạng thái**
- **Autosave**
- **Demo version**
- **Game**

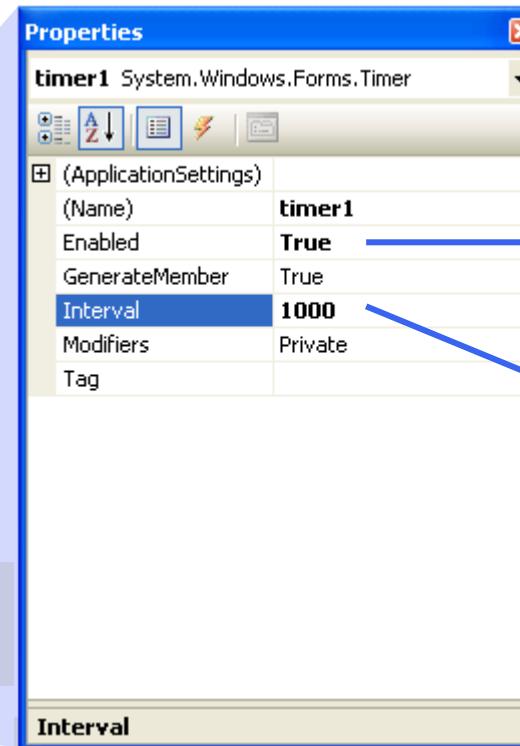
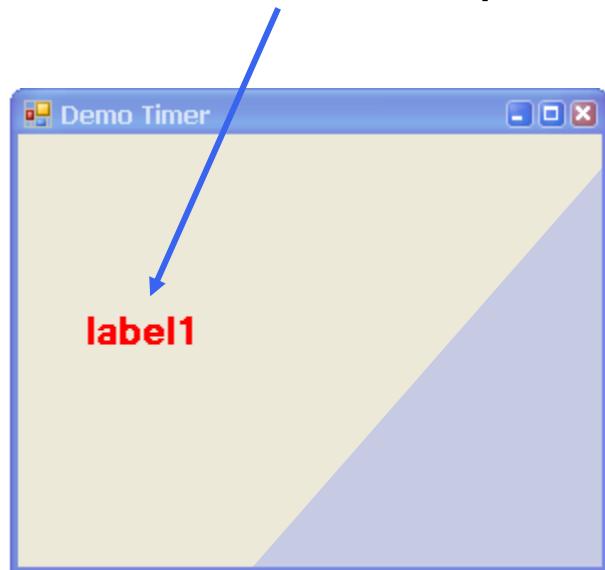
Xử lý sự kiện Timer

- Theo lý thuyết thông điệp thời gian do Windows cung cấp là chính xác đến mili giây nhưng thực tế không hoàn toàn như vậy.
- Sự chính xác còn phụ thuộc vào đồng hồ của hệ thống và các hoạt động hiện thời của chương trình.

Timer

- **Hiển thị giờ hệ thống**

Hiển thị thời gian

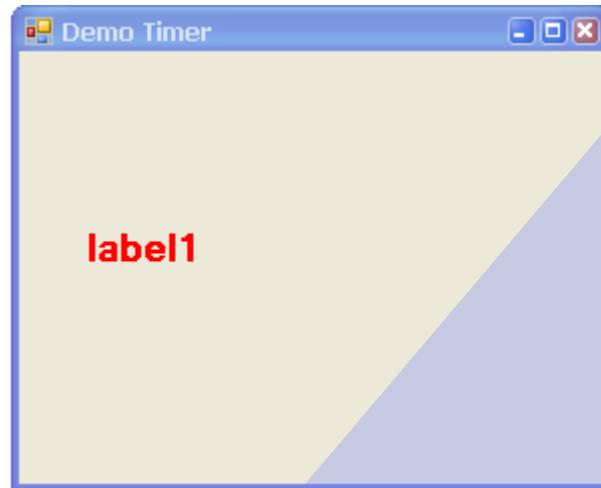


Enable sự kiện Tick

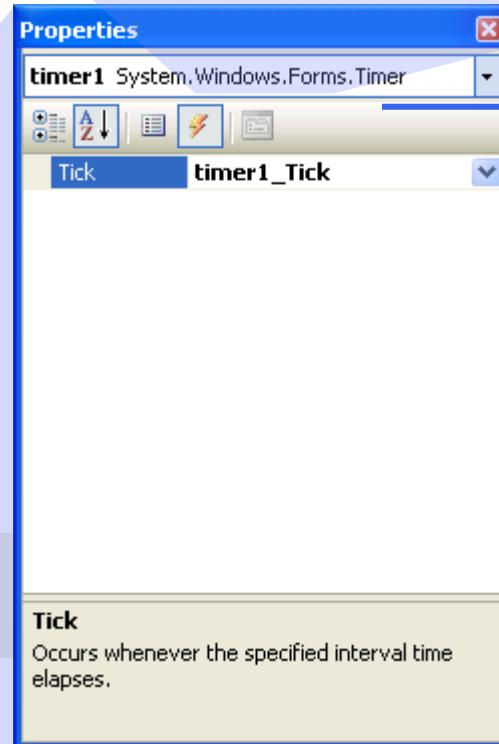
Khoảng thời gian chờ giữa 2 lần gọi Tick

Timer

- Sự kiện Tick



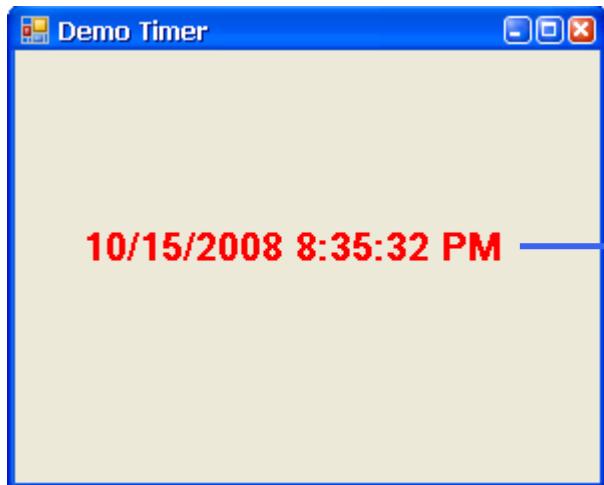
```
private void timer1_Tick(object sender, EventArgs e)
{
    // Lấy thời gian hệ thống
    DateTime now = DateTime.Now;
    label1.Text = now.ToString();
}
```



Khai báo trình xử lý sự kiện Tick

Timer

- **Demo**



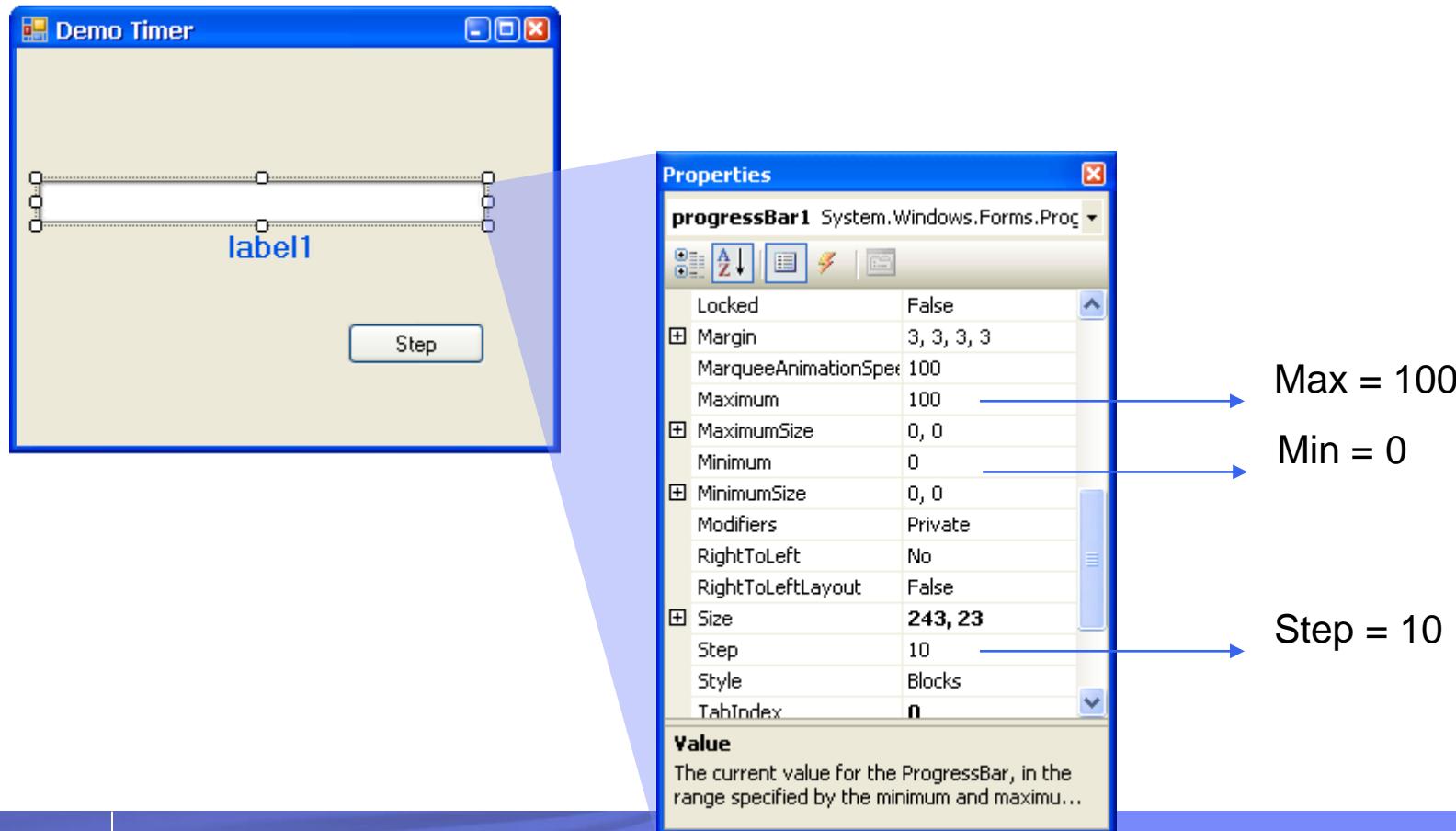
Mỗi giây sự kiện Tick
phát sinh. Trình xử lý
của Tick sẽ lấy giờ
hệ thống và hiển thị
lên Label

ProgressBar

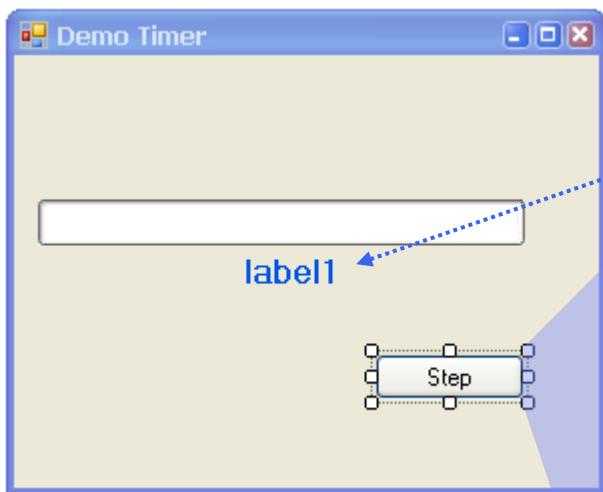
- **Hiển thị tiến độ thực hiện của một công việc nào đó**
- **Các thuộc tính**
 - **Minimum**: giá trị nhỏ nhất
 - **Maximum**: giá trị lớn nhất
 - **Step**: số bước tăng khi gọi hàm PerformStep
 - **Value**: giá trị hiện tại
 - **Style**: kiểu của progress bar
- **Phương thức**
 - **PerformStep()**: tăng thêm step
 - **Increment(int value)**: tăng vị trí hiện tại của tiến độ với giá trị xác định

ProgressBar

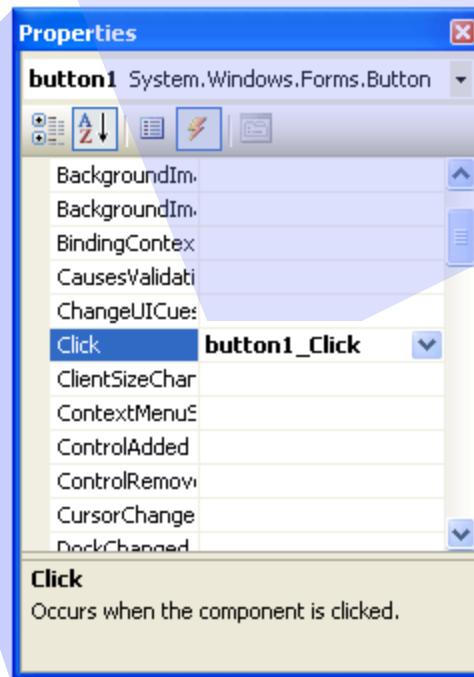
- Khai báo thanh tiến độ 0-100, step = 10



ProgressBar

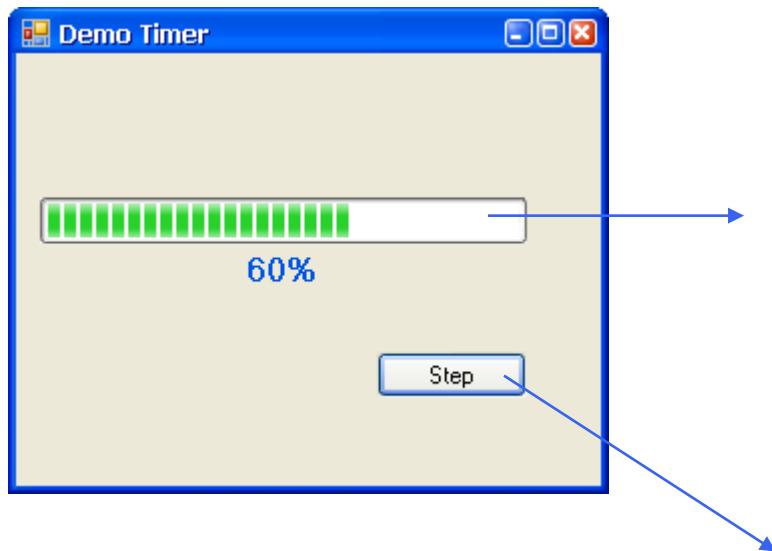


```
private void button1_Click(object sender, EventArgs e)
{
    // tăng thêm một bước: value +=step
    progressBar1.PerformStep();
    // hiển thị tiến độ % lên label
    label1.Text = progressBar1.Value.ToString() + "%";
}
```



ProgressBar

■ Demo



Thể hiện trực
quan tiến độ

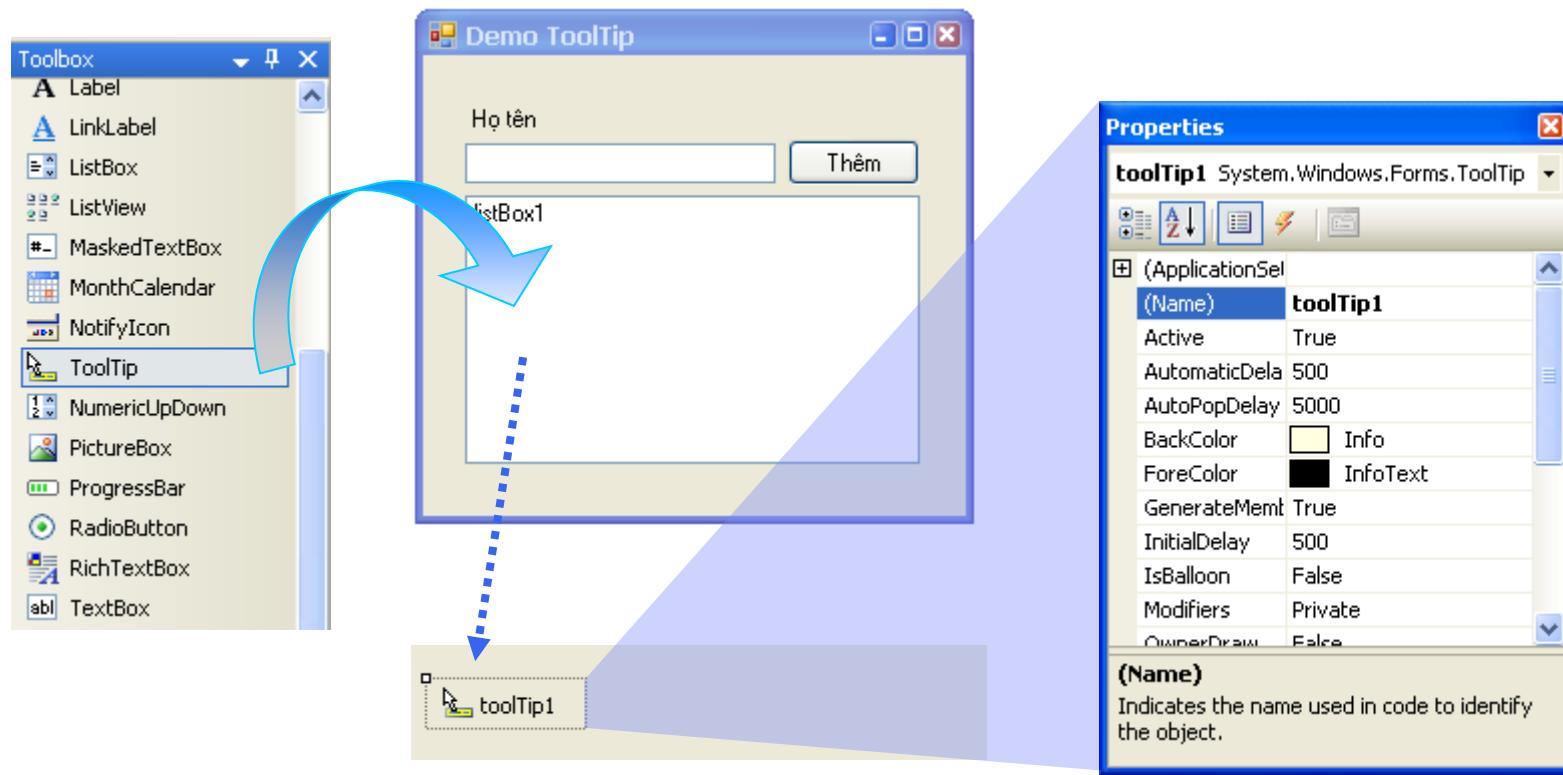
Tăng tiến độ theo step
và cập nhật lại % hoàn
thành lên label

ToolTip

- **Cung cấp chức năng hiển thị một khung text nhỏ khi user di chuyển chuột vào control bất kỳ**
- **Khung text chứa nội dung mô tả ý nghĩa của control**
- **Cách sử dụng**
 - Từ ToolBox kéo ToolTip thả vào form
 - Kích chọn control muốn thêm tooltip
 - Trong cửa sổ Properties của control sẽ có thuộc tính ToolTip. Thêm text vào thuộc tính này để hiển thị khi tooltip xuất hiện.

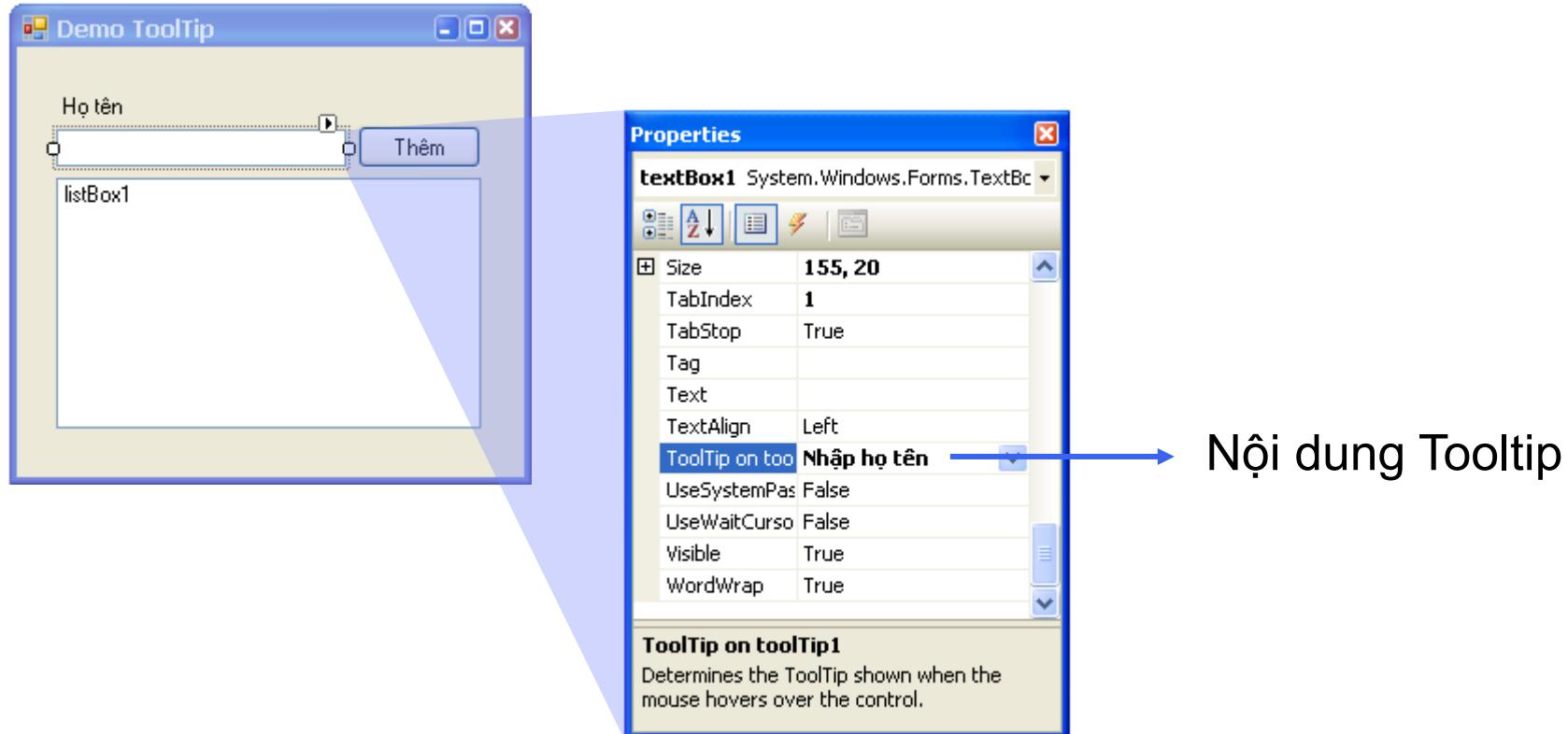
ToolTip

■ Tạo ToolTip



ToolTip

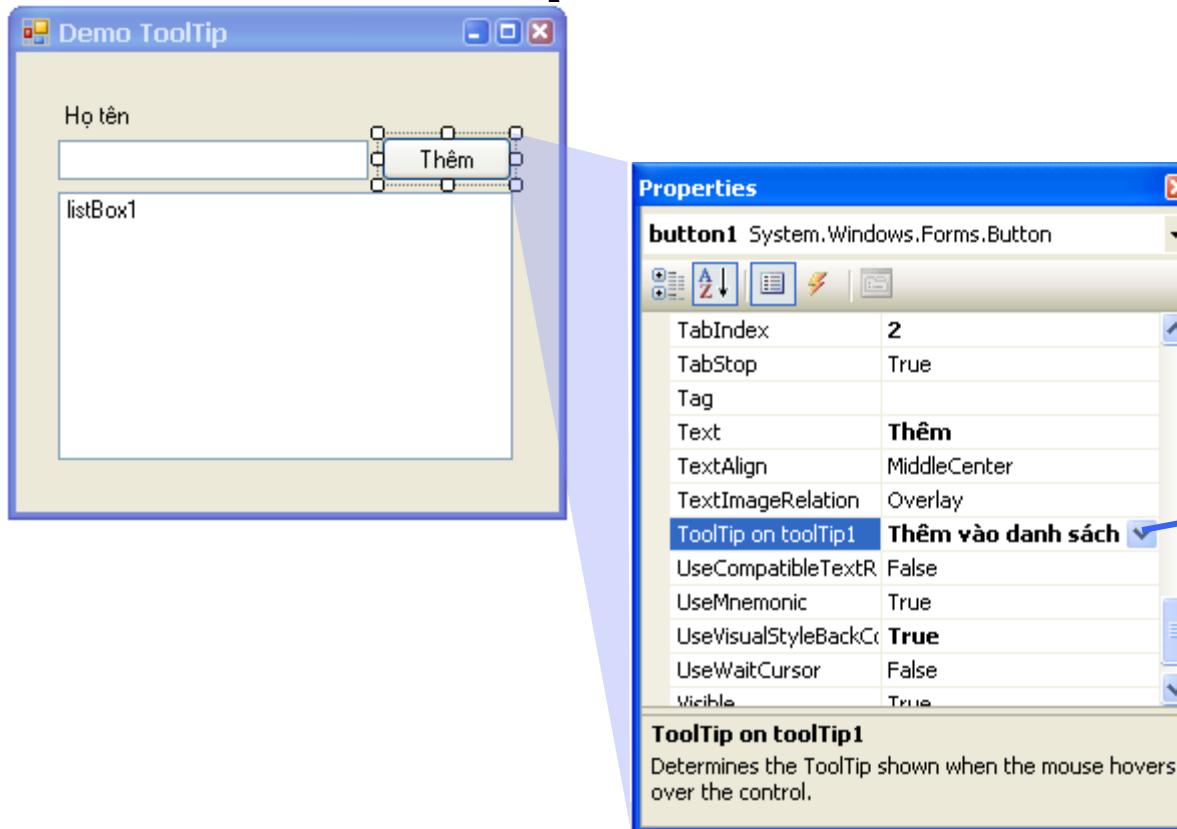
- Khai báo Tooltip cho textbox trong Design View



Nội dung Tooltip

ToolTip

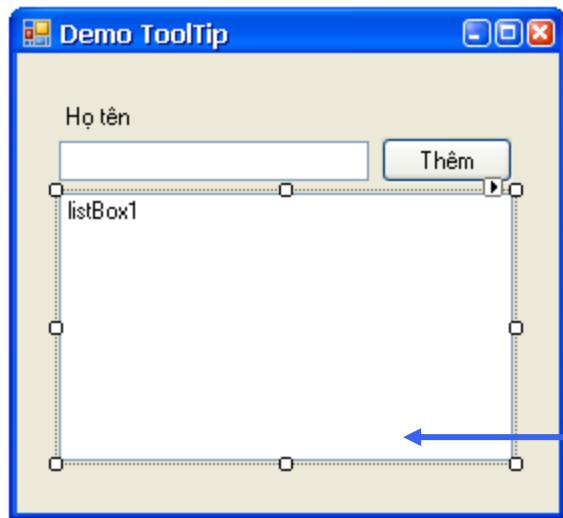
■ Khai báo tooltip cho button



Nhập nội dung Tooltip
cần hiển thị

ToolTip

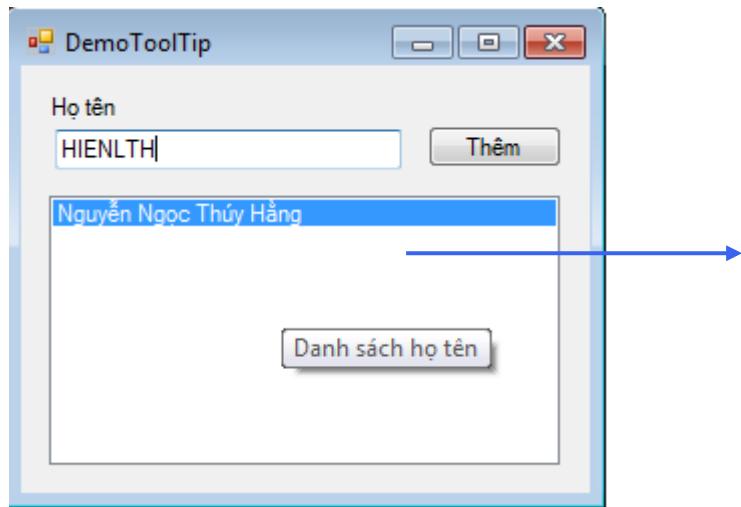
- Khai báo tooltip cho listbox bằng code



```
private void Form1_Load(object sender, EventArgs e)
{
    // thiết lập tooltip cho listBox1
    toolTip1.SetToolTip(listBox1, "Danh sách họ tên");
}
```

ToolTip

- **Demo**



ToolTip xuất hiện khi
user di chuyển chuột
vào vùng control

TrackBar

- Cho phép user thiết lập giá trị trong khoảng cố định cho trước
- Thao tác qua thiết bị chuột hoặc bàn phím Properties

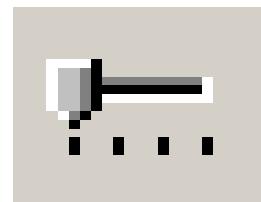
Minimum

Maximum

TickFrequency

TickStyle

Value



ValueChanged
Scroll

Methods

SetRange

TrackBar

```
public void AddTrackBar() {  
    TrackBar tb1 = new TrackBar();  
    tb1.Location = new Point(10, 10);  
    tb1.Size = new Size(250, 50);  
  
    tb1.Minimum = 0;           → Thiết lập khoảng: 0 - 100  
    tb1.Maximum = 100;  
  
    tb1.SmallChange = 1;       → Số vị trí di chuyển khi dùng  
    tb1.LargeChange = 5;       → phím mũi tên  
    tb1.TickStyle = TickStyle.BottomRight;  → Số vị trí di chuyển  
                                         khi dùng phím  
                                         Page  
                                         Kiểu stick ở bên  
                                         dưới/bên phải track  
  
    tb1.TickFrequency = 10;  
    tb1.Value = 10;  
    Controls.Add(tb1);  
}
```

Tạo thể hiện

Thiết lập khoảng: 0 - 100

Số vị trí di chuyển
khi dùng phím mũi tên

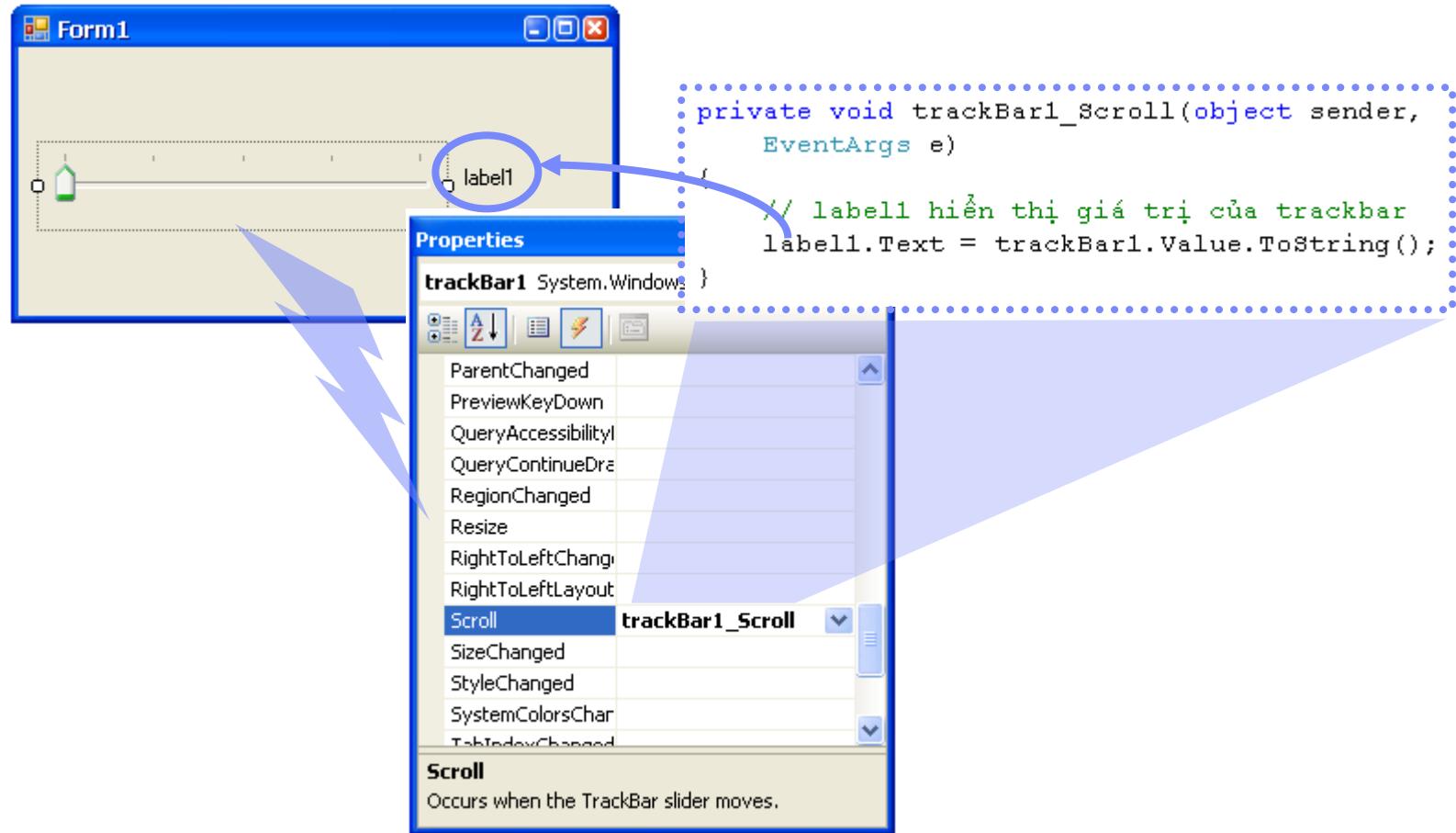
Số vị trí di chuyển
khi dùng phím

Page
Kiểu stick ở bên
dưới/bên phải track

Số khoảng cách
giữa các tick mark

TrackBar

- Bổ sung Label hiển thị giá trị của TrackBar

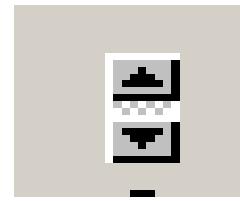
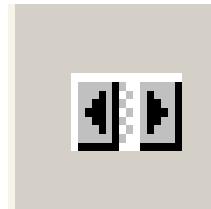


ScrollBar controls

Properties

HScrollBar and VScrollBar

SmallChange



Value

Events

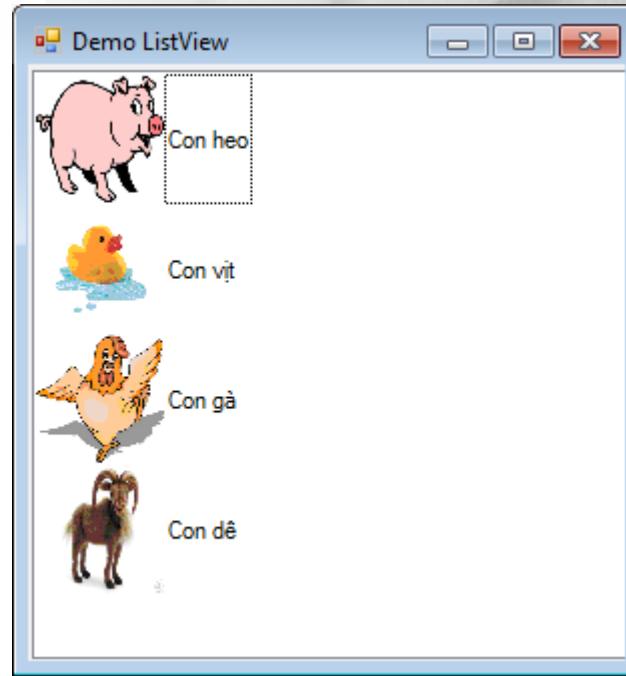
LargeChange

Scroll

ValueChanged

Bài giảng môn học

ListView



Lập trình Windows Form với C#

List View

- **Dạng control phổ biến hiện thị một danh sách item**
 - Các item có thể có các item con gọi là subitem
- **Windows Explorer hiển thị thông tin thư mục, tập tin...**
 - Có thể hiển thị thông tin theo nhiều dạng thông qua thuộc tính View
 - Xem dạng chi tiết thông tin
 - Xem dạng icon nhỏ
 - Xem dạng icon lớn
 - Xem dạng tóm tắt
 - ...
- **Lớp ListView dẫn xuất từ System.Windows.Forms.Control**

List View

■ *Properties*

Columns

Items

Sorting

ListView

GridLines

MultiSelect

View

SmallImageList

LargeImageList

FullRowSelect

List View

- Các dạng thể hiện của *ListView*

Details

Small Icons

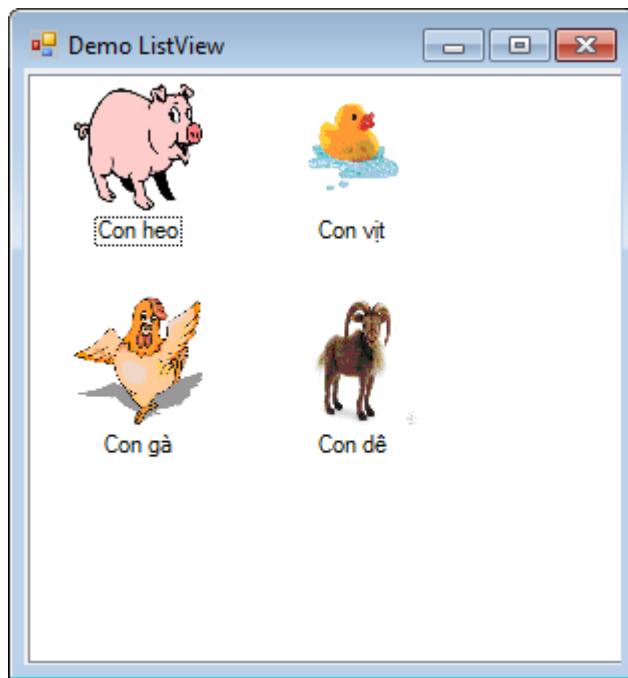
ListView

List

Large Icons

Tile

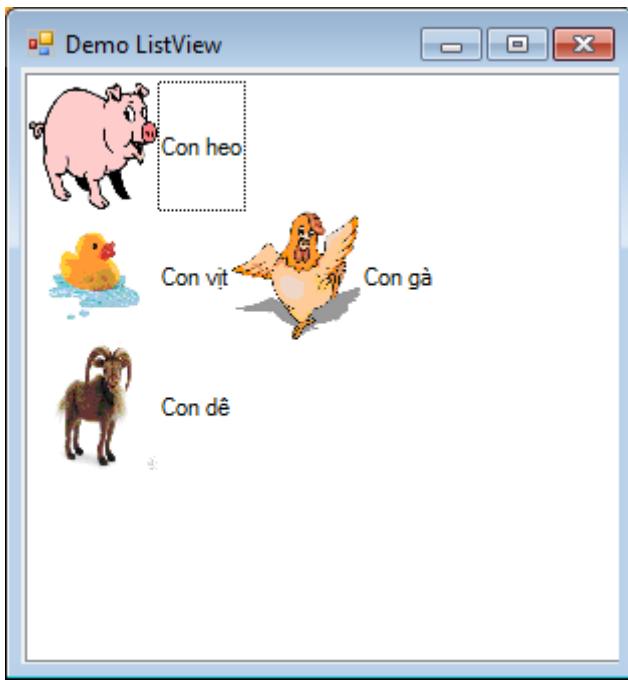
List View



Large Icons

Mỗi item xuất hiện với 1 icon kích thước lớn và một label bên dưới

List View

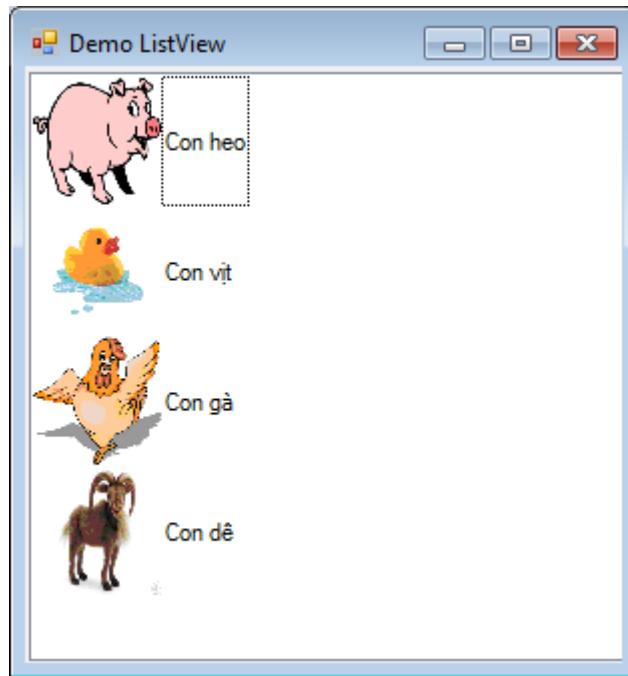


Small Icons

Mỗi item xuất hiện với icon nhỏ và một label bên phải

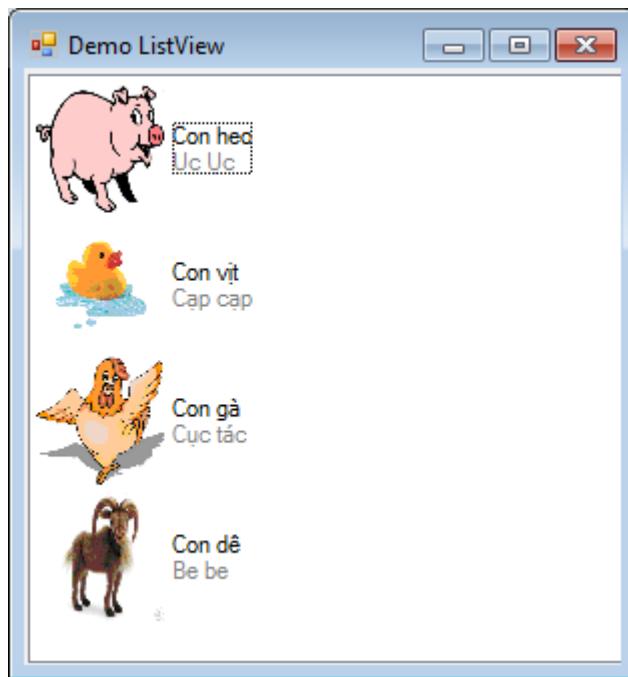
List View

List



Mỗi item xuất hiện với icon nhỏ với label bên phải, item được sắp theo cột nhưng không có tiêu đề cột

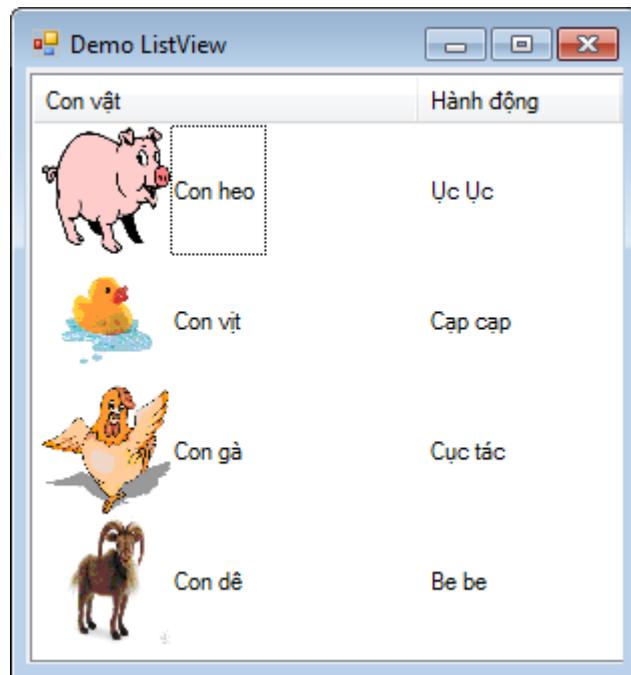
List View



Tile

Mỗi item xuất hiện với icon kích thước lớn, bên phải có label chứa item và subitem

List View



Detail

Mỗi item xuất hiện trên một dòng, mỗi dòng có các cột chứa thông tin chi tiết

List View

- **Tạo các cột cho ListView – Details qua**
 - Cửa sổ properties → Columns để tạo.
 - Sử dụng code trong chương trình.

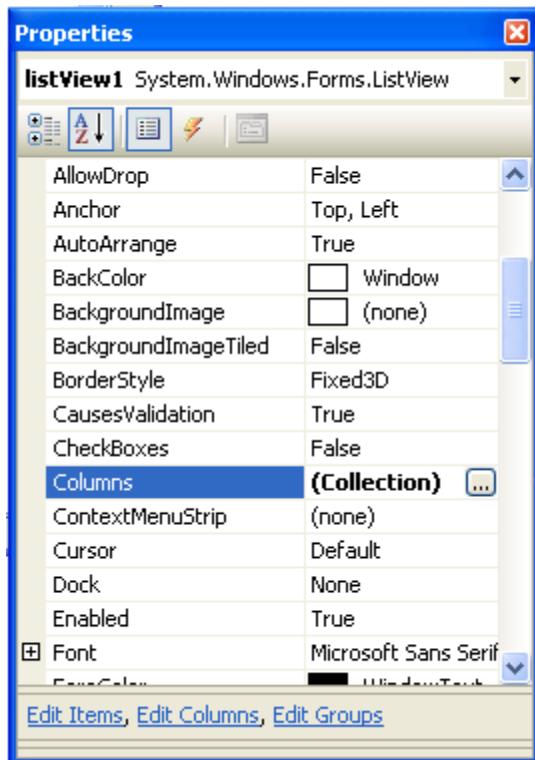
```
ColumnHeader columnHeader1 = new ColumnHeader();
ColumnHeader columnHeader2 = new ColumnHeader();
ColumnHeader columnHeader3 = new ColumnHeader();

columnHeader1.Text = "Name";
columnHeader2.Text = "Address";
columnHeader3.Text = "Telephone Number";

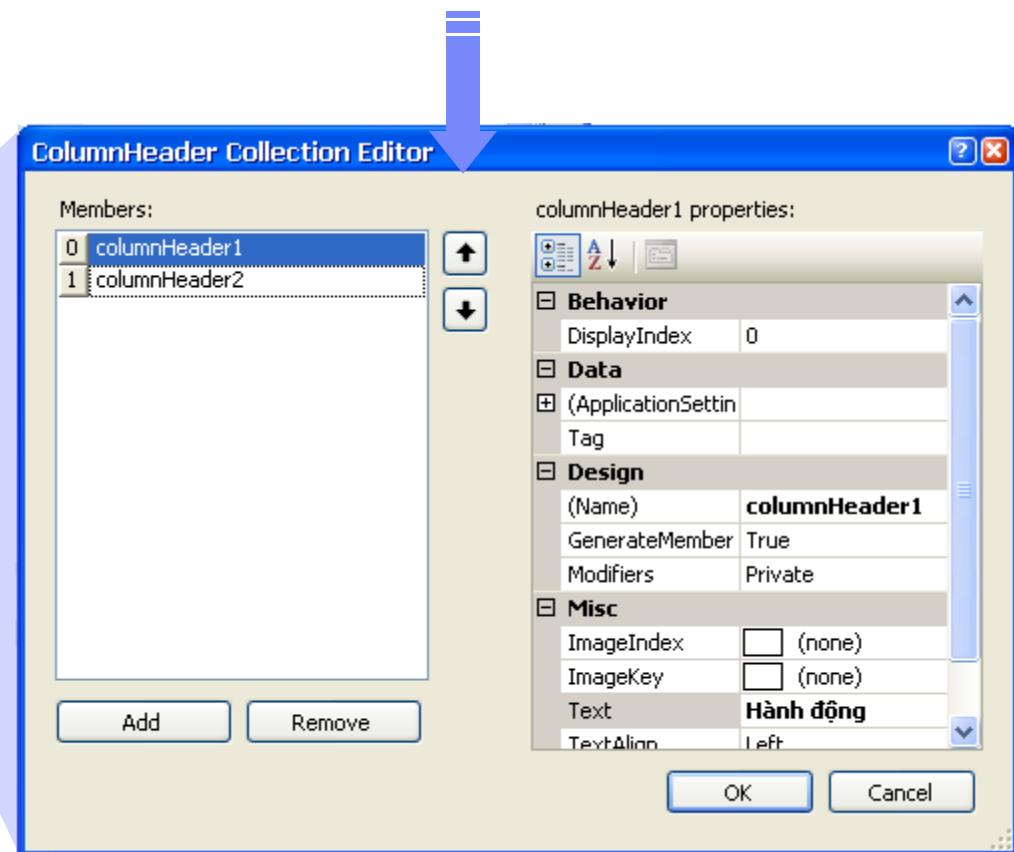
listView1.Columns.Add(columnHeader1);
listView1.Columns.Add(columnHeader2);
listView1.Columns.Add(columnHeader3);
```

Code

List View



Dialog soạn thảo cột



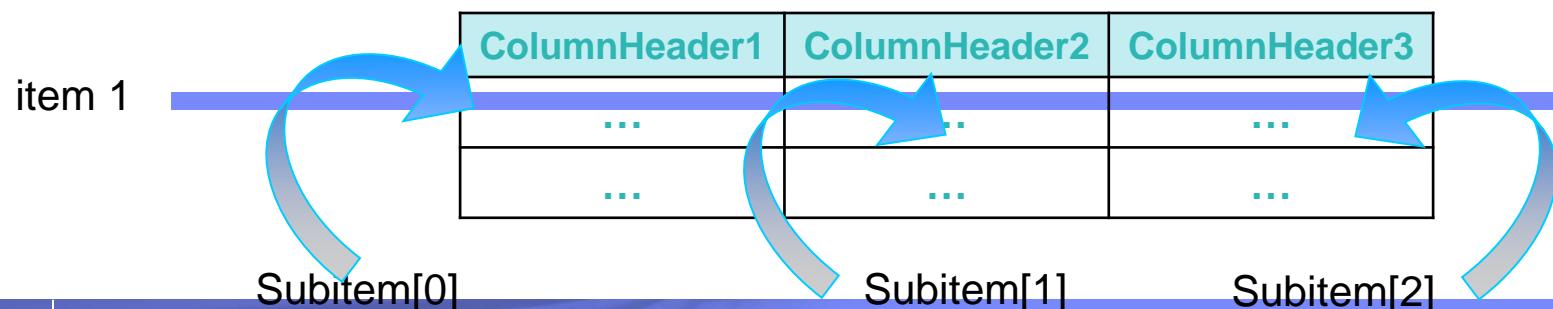
List View

■ Thêm các item vào ListView

- Thêm item trong màn hình thiết kế form
- Thêm item thông qua code

■ Các lớp định nghĩa Item

- System.Windows.Forms.ListViewItem
- Mỗi item trong ListView có các item phụ gọi là subitem
 - Lớp ListViewItem.ListViewSubItem định nghĩa các subitem của ListView
 - Lớp ListViewSubItem là inner class của ListViewItem



ListView

■ Minh họa thêm item qua code

```
ListViewItem item1 = new ListViewItem();
ListViewItem.ListViewSubItem subitem1;
subitem1 = new ListViewItem.ListViewSubItem();
```

```
item1.Text = "HCMUP";
subitem1.Text = "280 An Duong Vuong, HCMC";
```

```
item1.SubItems.Add(subitem1);
```

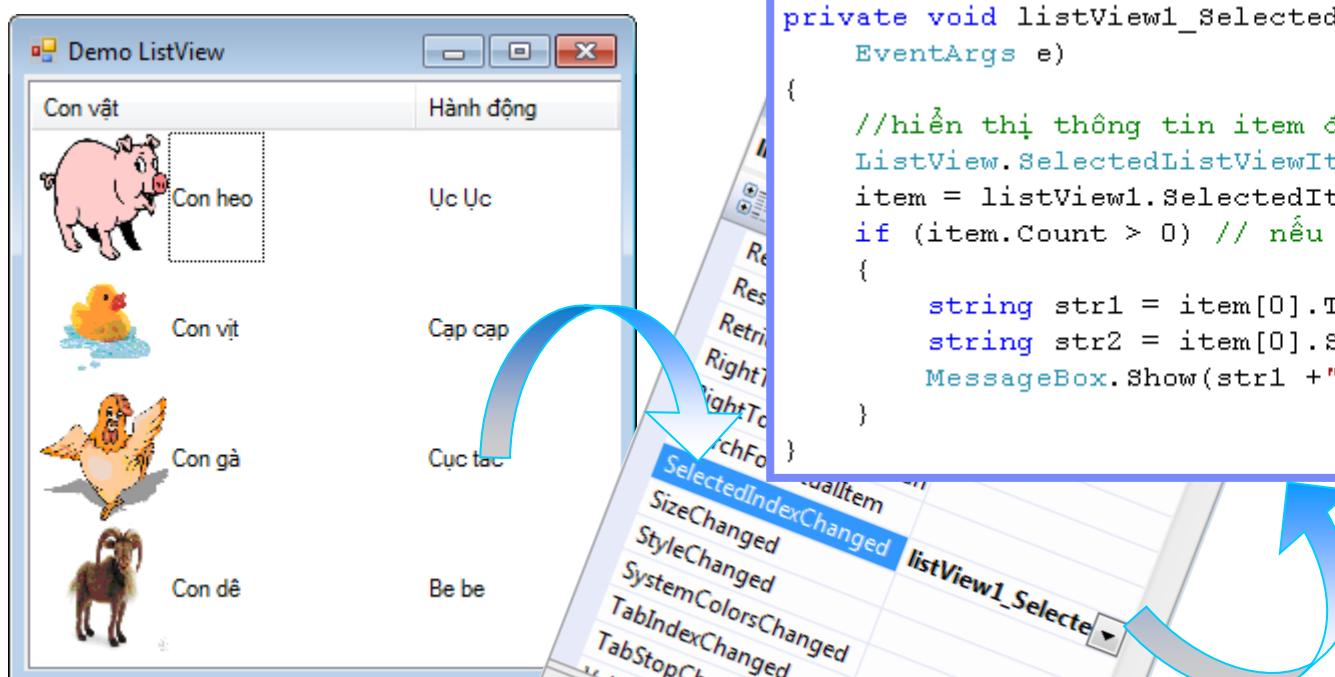
```
listView1.Items.Add(item1);
```

Thêm subitem vào item

Thêm item vào danh sách
items của ListView

ListView

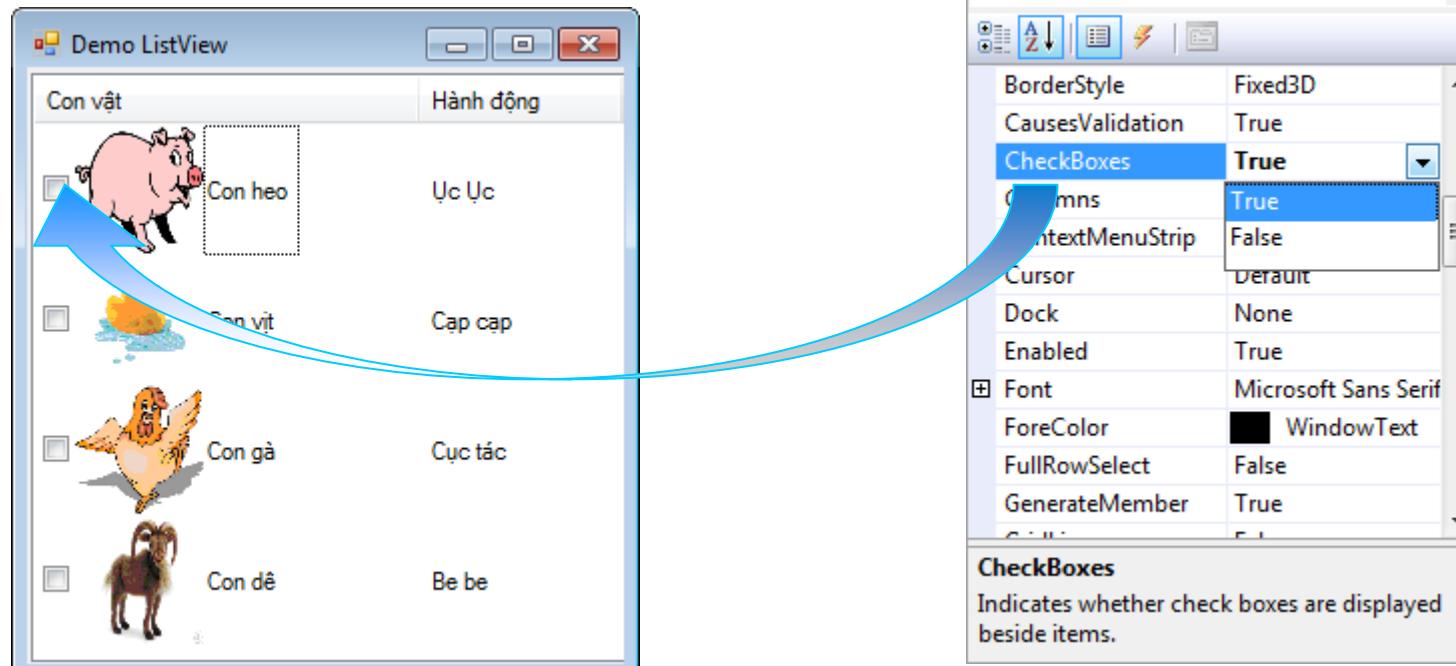
▪ Sự kiện **SelectedIndexChanged**



```
private void listView1_SelectedIndexChanged(object sender, EventArgs e)
{
    //hiển thị thông tin item được chọn
    ListView.SelectedListViewItemCollection item;
    item = listView1.SelectedItems; // lấy item được chọn
    if (item.Count > 0) // nếu có item được chọn
    {
        string str1 = item[0].Text;
        string str2 = item[0].SubItems[1].Text;
        MessageBox.Show(str1 + "-" + str2);
    }
}
```

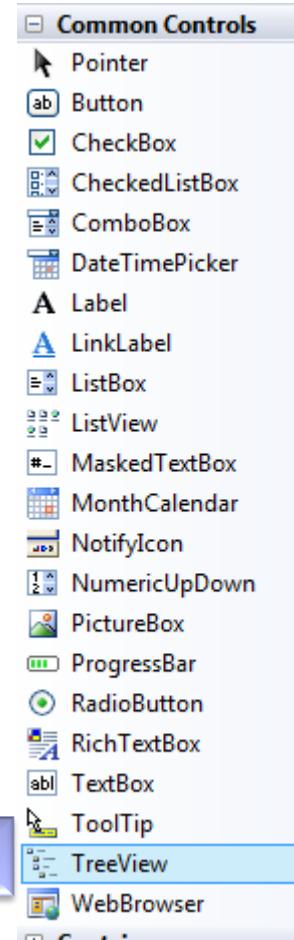
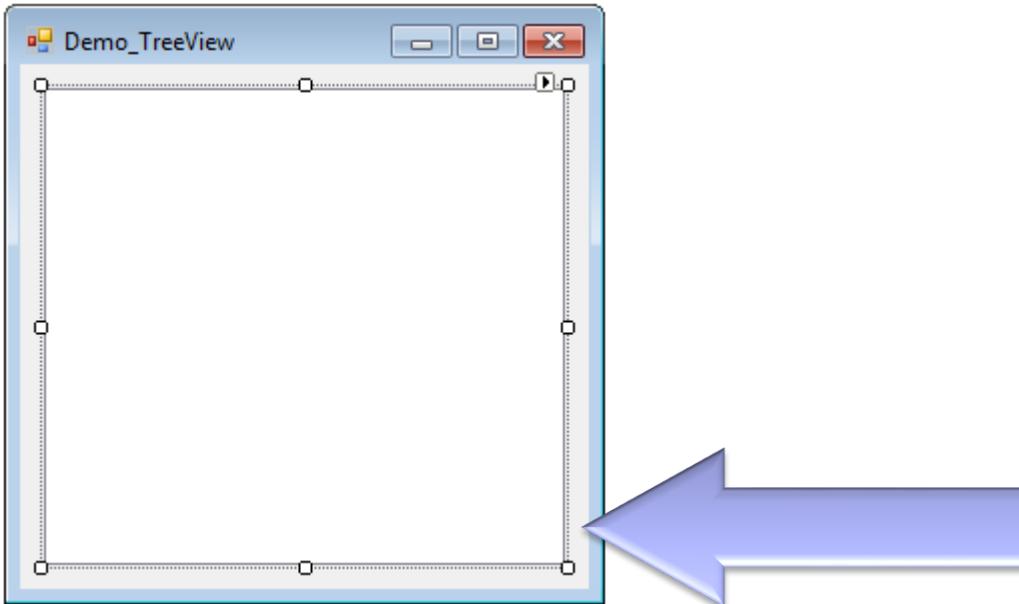
ListView

- Thuộc tính **Checkboxes**: Xuất hiện CheckBox bên cạnh từng phần tử trên điều khiển



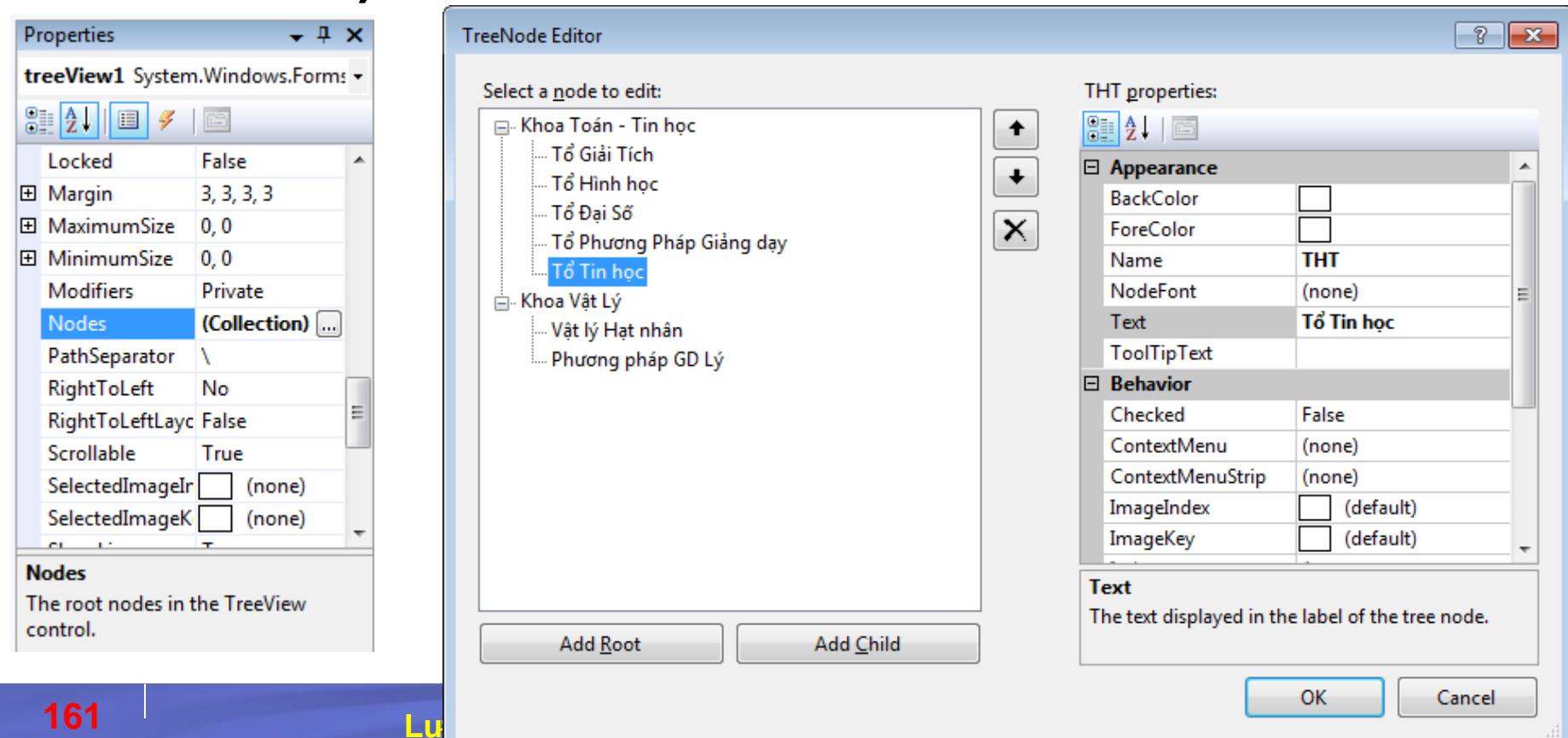
TreeView

- Điều khiển TreeView dùng để trình bày danh sách phần tử phân cấp theo từng nút (Node) hình cây.
- Thực hiện: ta kéo thả điều khiển vào Form từ hộp Toolbox



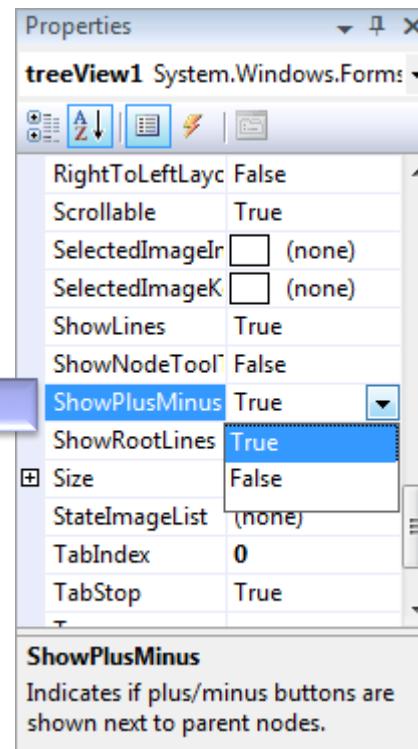
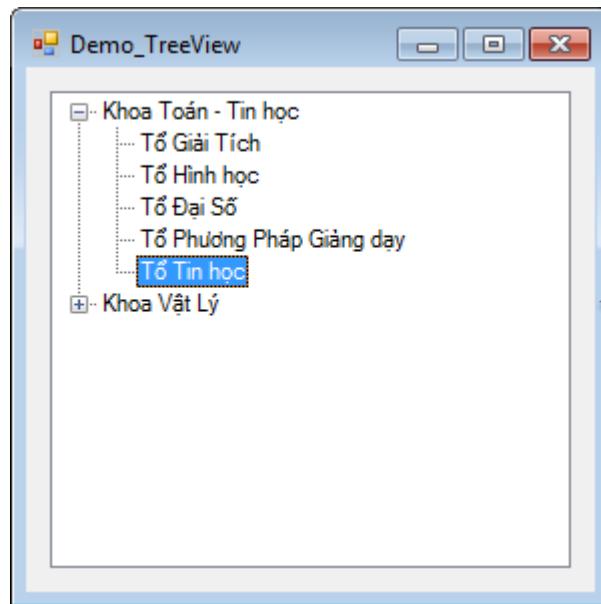
TreeView

- Thuộc tính **Nodes**: Khai báo số Node (có header) của điều khiển TreeView



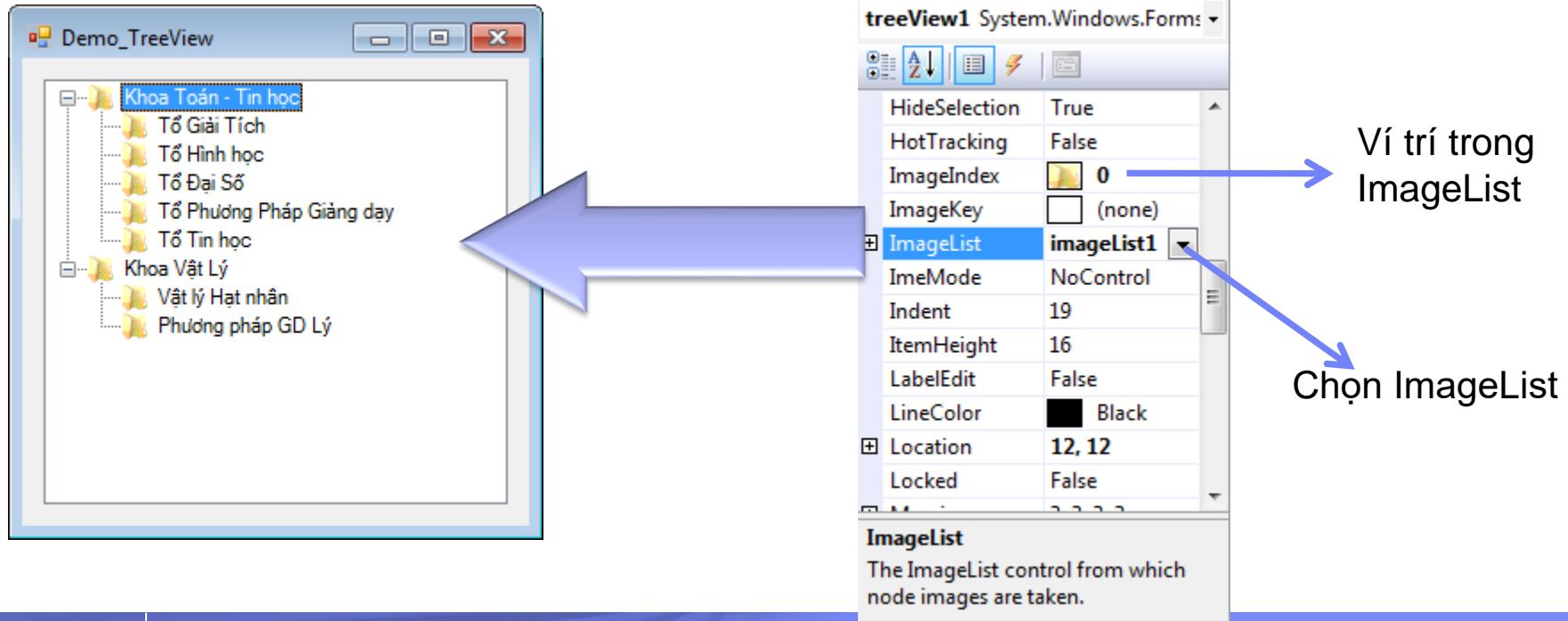
TreeView

- Thuộc tính **ShowPlusMinus**: Chọn giá trị True thì biểu tượng dấu + và – xuất hiện trên mỗi Node, mặc định là True.



TreeView

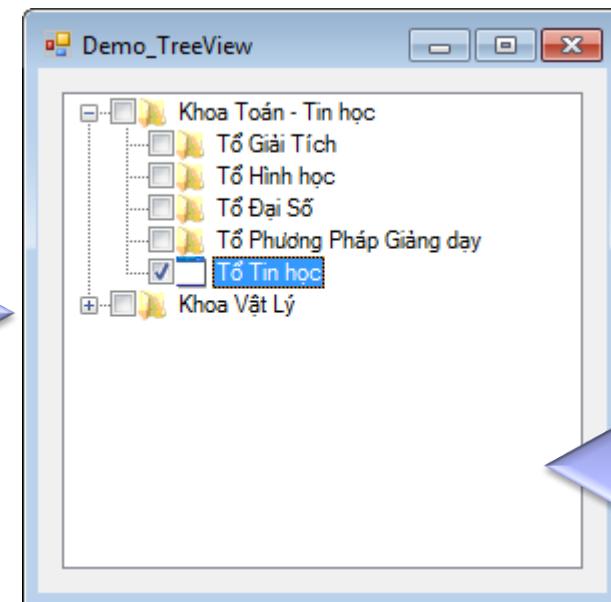
- Thuộc tính **ImageList**: Chứa đối tượng **ImageList**.
- Thuộc tính **ImageIndex**: Tất cả các Node được thể hiện bởi hình ảnh mà thuộc tính này đã chọn.



TreeView

- Thuộc tính **SelectedImageIndex**: Node được chọn sẽ có dạng hình ảnh này.
- Thuộc tính **Checkboxes**: Xuất hiện CheckBox bên cạnh Node

Properties	
treeView1	System.Windows.Forms
	(Collection)
Nodes	(Collection)
PathSeparator	\
RightToLeft	No
RightToLeftLayout	False
Scrollable	True
SelectedImageIndex	<input type="text"/> 1 <input type="button" value="▼"/>
SelectedImageKey	<input type="text"/> (none)
ShowLines	True
ShowNodeToolTips	False
ShowPlusMinus	True
ShowRootLines	True
Size	260, 238
SelectedImageIndex	
The default image index for selected nodes.	



Properties	
treeView1	System.Windows.Forms
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
BackColor	<input type="color"/> Window
BorderStyle	Fixed3D
CausesValidation	True
Checkboxes	<input checked="" type="checkbox"/>
ContextMenuStrip	(none)
Cursor	Default
Dock	None
DrawMode	Normal
Checkboxes	
Indicates whether check boxes are displayed beside nodes.	

Chọn hình cho
node được chọn

TreeView

■ Phương thức:

- **CollapseAll()**: Trình bày tắt cả các Node trên điều khiển TreeView
`treeView1.CollapseAll();`
- **ExpandAll()**: Thu gọn tất cả các Node trên điều khiển TreeView
`treeView1.ExpandAll();`
- **Clear()**: Xóa tất cả các Node đang tồn tại.

■ Các biến cố thông dụng như:

- **AfterCheck**: Xảy ra khi người dùng Check vào CheckBox
- **AfterCollapse**: Xảy ra khi thu gọn tất cả các Node
- **AfterExpand**: Xảy ra khi mở rộng tất cả các Node
- **AfterSelect**: Xảy ra khi Click vào Node
- Tương tự đối với BeforeCheck, BeforeCollapse, BeforeExpand, BeforeSelect

Bài tập

- Sử dụng TreeView và ListView mô phỏng chương trình Windows Explorer

Tham khảo

- **Slide bài giảng C#, ĐH KHTN, TpHCM**
- **Nguyễn Hà Giang, Slide bài giảng C#, ĐH KTCN, TpHCM**
- **MSDN**

