

CHƯƠNG 2: WINDOWS APPLICATION (2)

1

**KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

Common file dialogs

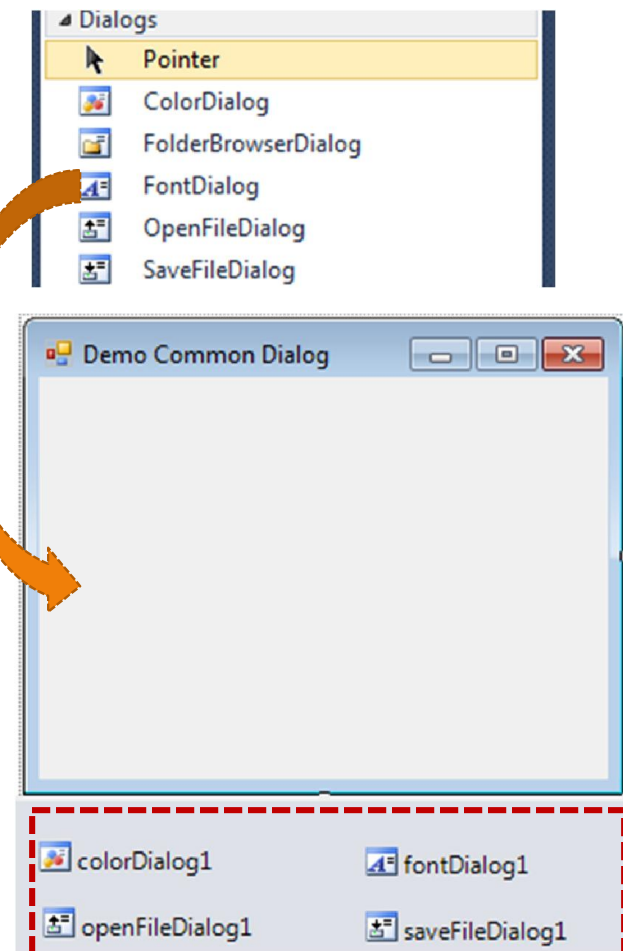
2

**OPENFILEDIALOG
SAVEFILEDIALOG
FONTDIALOG
COLORDIALOG**

Common Dialog

3

- Cách sử dụng:
 - Kéo thả dialog vào Form.



OpenFileDialog, SaveFileDialog

4

Thuộc tính	Mô tả
FileName, FileNames	-Trả về tên file chọn trong dialog khi nhấn OK -Trả về tên tập file chọn trong dialog khi nhấn OK
Filter	- Thuộc tính này dùng để quy định các loại tập tin sẽ được hiển thị trong hộp thoại. Ví dụ, hiển thị các tập tin ảnh:

```
OpenFileDialog dlg = new OpenFileDialog();  
dlg.Title = "Open Dialog";// "Add Photos";  
dlg.Multiselect = true;  
dlg.Filter = "Image Files (JPEG, GIF, BMP, etc.) |"  
            + "*.jpg;*.jpeg;*.gif;*.bmp;"  
            + "*.tif;*.tiff;*.png |"  
            + "JPEG files (*.jpg;*.jpeg)|*.jpg;*.jpeg |"  
            + "GIF files (*.gif)|*.gif |"  
            + "BMP files (*.bmp)|*.bmp |"  
            + "TIFF files (*.tif;*.tiff)|*.tif;*.tiff |"  
            + "PNG files (*.png)|*.png |"  
            + "All files (*.*)|*.*";
```

OpenFileDialog, SaveFileDialog

5

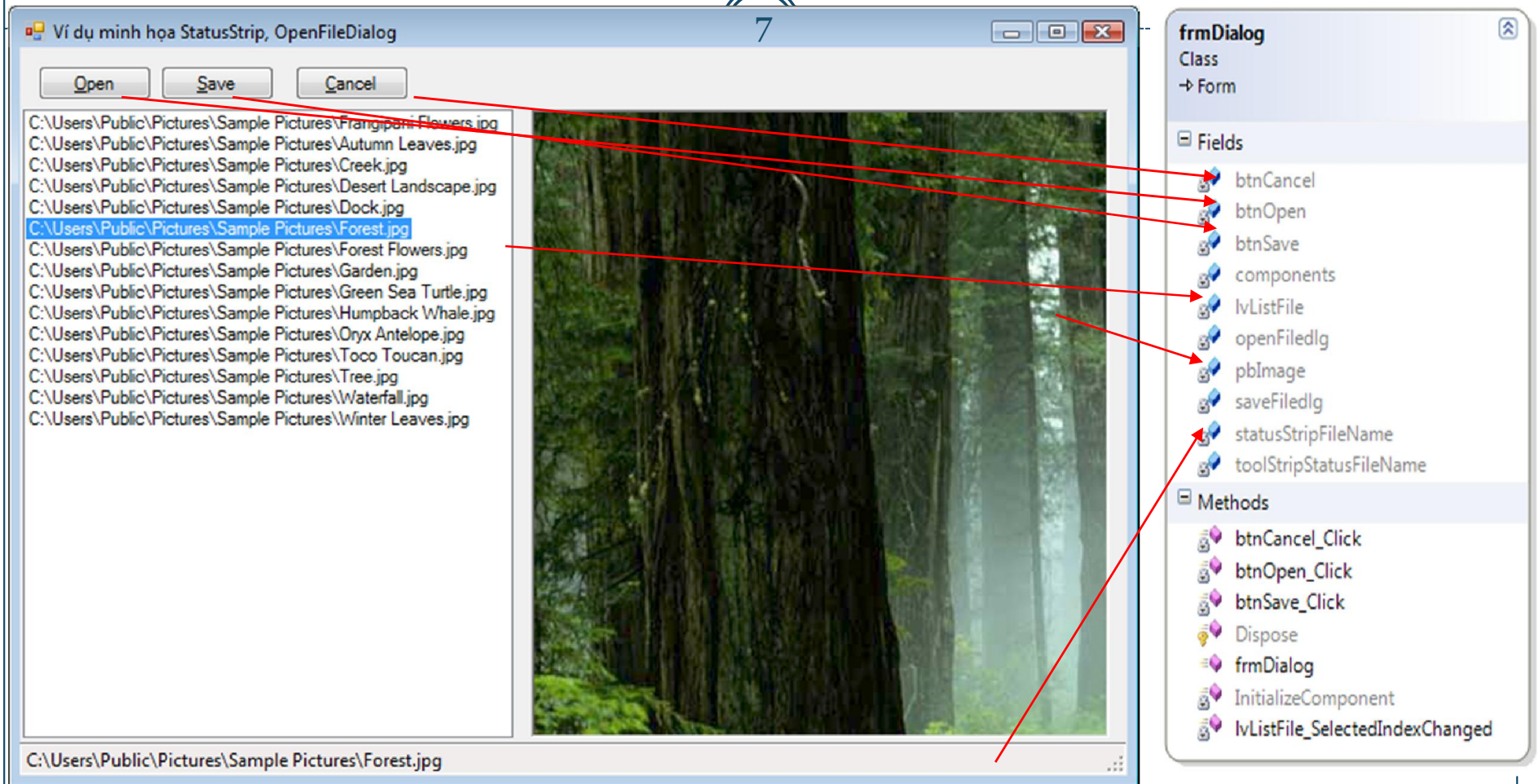
Thuộc tính	Mô tả
InitialDirectory	Quy định tên của thư mục khi mở hộp thoại <code>dlg.InitialDirectory = Environment.CurrentDirectory;</code> <code>dlg.InitialDirectory = "C:\\Users\\Public\\Pictures\\Sample Pictures";</code>
RestoreDirectory	Mỗi khi mở hộp thoại Open và Save , thư mục hiện hành là thư mục được người dùng chọn lần cuối trước đó sẽ được hiển thị. <code>dlg.RestoreDirectory = true; // false</code>
MultiSelect	true, false: cho phép chọn nhiều tập tin
ShowReadOnly	true, false: Cho phép hiển thị Check box “Open As Read Only” trong hộp thoại Open hay không. <code>dlg.ShowReadOnly = true;</code> <code>dlg.ReadOnlyChecked = true;</code>
ReadOnlyChecked	true, false: Thuộc tính này quy định giá trị của check box ReadOnlyChecked có được chọn hay không.

Thuộc tính chỉ dành cho OpenFileDialog

6

Thuộc tính	Mô tả
MultiSelect	true, false: cho phép chọn nhiều tập tin
ShowReadOnly	true, false: Cho phép hiển thị Check box “Open As Read Only” trong hộp thoại Open hay không. <code>dlg.ShowReadOnly = true;</code> <code>dlg.ReadOnlyChecked = true;</code>
ReadOnlyChecked	true, false: Thuộc tính này quy định giá trị của check box ReadOnlyChecked có được chọn hay không.

OpenDialog, SaveDialog, StatusStrip



```
private void btnOpen_Click(object sender, EventArgs e)
{
    openFileDialog.Title = "Open Dialog";
    openFileDialog.Multiselect = true;
    openFileDialog.Filter = "Image Files (JPEG, GIF, BMP, etc.)|"
        + "*.jpg;*.jpeg;*.gif;*.bmp;"
        + "*.tif;*.tiff;*.png|"
        + "JPEG files (*.jpg;*.jpeg)|*.jpg;*.jpeg|"
        + "GIF files (*.gif)|*.gif|"
        + "BMP files (*.bmp)|*.bmp|"
        + "TIFF files (*.tif;*.tiff)|*.tif;*.tiff|"
        + "PNG files (*.png)|*.png|"
        + "All files (*.*)|*.*";

    openFileDialog.InitialDirectory = Environment.CurrentDirectory;
    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        string[] filenames = openFileDialog.FileNames;
        for(int i=0; i<filenames.Length; i++)
            this.lvListFile.Items.Add(new ListViewItem(filenames[i]));
    }
}

private void btnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}
```



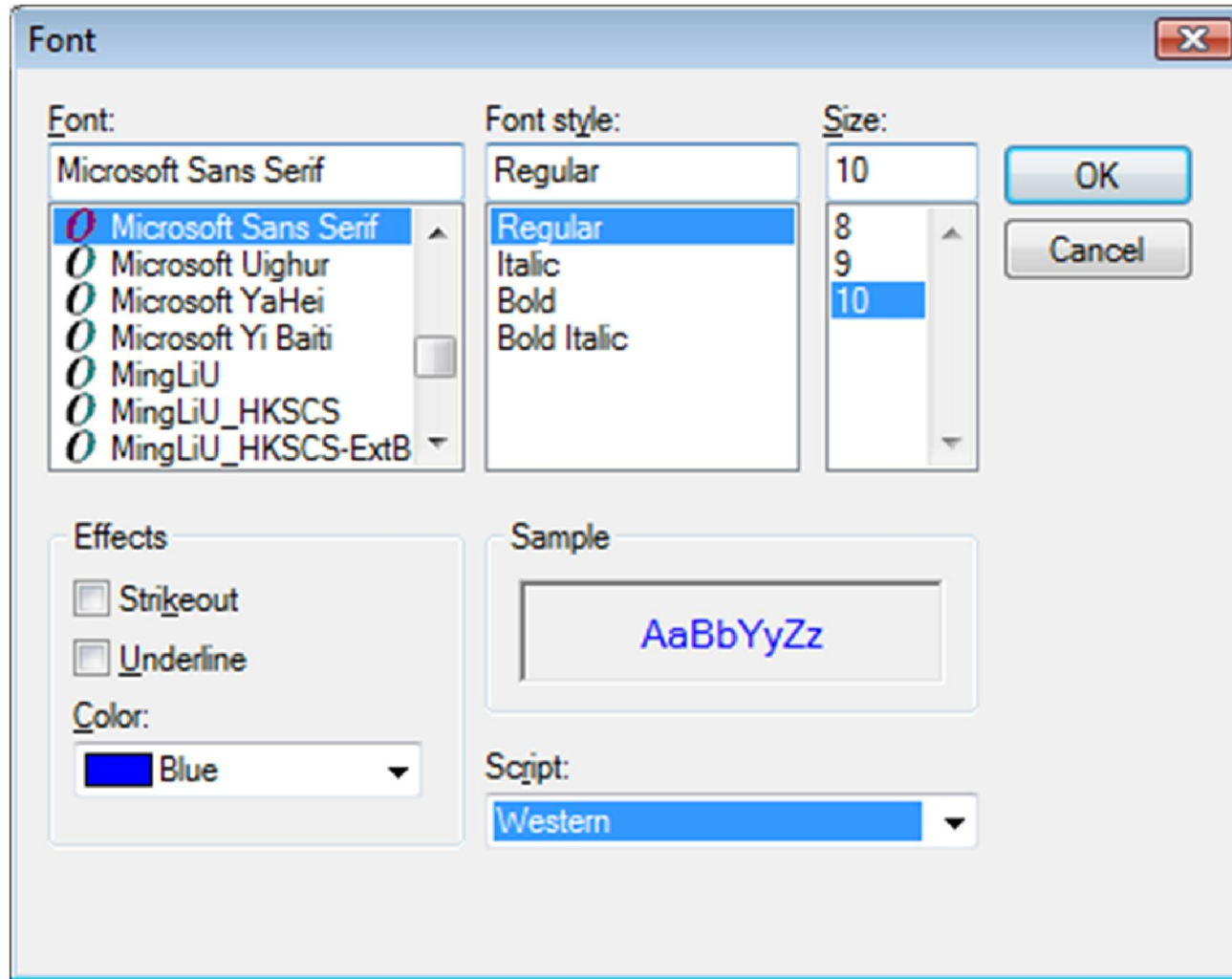
```

private void lvListFile_SelectedIndexChanged(object sender, EventArgs e)
{
    int i = lvListFile.SelectedItems.Count - 1;
    if (i >= 0)
    {
        ListViewItem lvitem = lvListFile.SelectedItems[0];
        this.pbImage.ImageLocation = lvitem.Text;
        toolStripStatusLabelFileName.Text = lvitem.Text;
    }
}
private void btnSave_Click(object sender, EventArgs e)
{
    saveFileDialog.Title = "Save file ..";
    saveFileDialog.Filter = "Image Files (JPEG, BMP, GIF, ..) "
+ "|(*.jpeg; *.jpg;)|"+
    "Bitmap files (*.bmp)|*.bmp|"+
    "All files (*.*)|*.*";
    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        try
        {
            Image im = pbImage.Image;
            im.Save(saveFileDialog.FileName, ImageFormat.Bmp);
        }
        catch
        {
            MessageBox.Show("Loi luu anh");
        }
    }
}

```

FontDialog

10

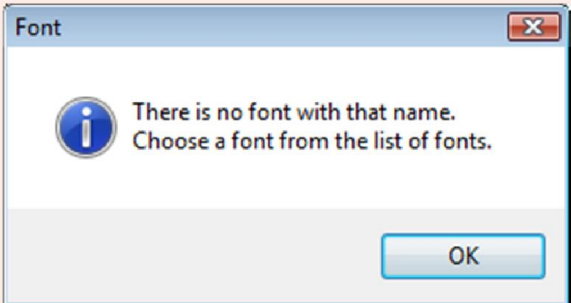


FontDialog

11

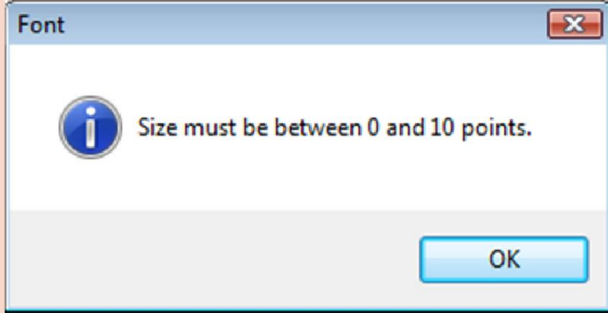
- FontDialog dùng chọn Font, kiểu chữ, kích cỡ và chọn màu chữ

Thuộc tính	Giá trị
AllowScriptChange	Cho phép combo Script hiển thị (true, false)
Color	Get hoặc set giá trị màu trên hộp thoại
Font	Get hoặc set giá trị font của hộp thoại
FontMustExist	Thông báo nhập font hợp lệ (true, false)



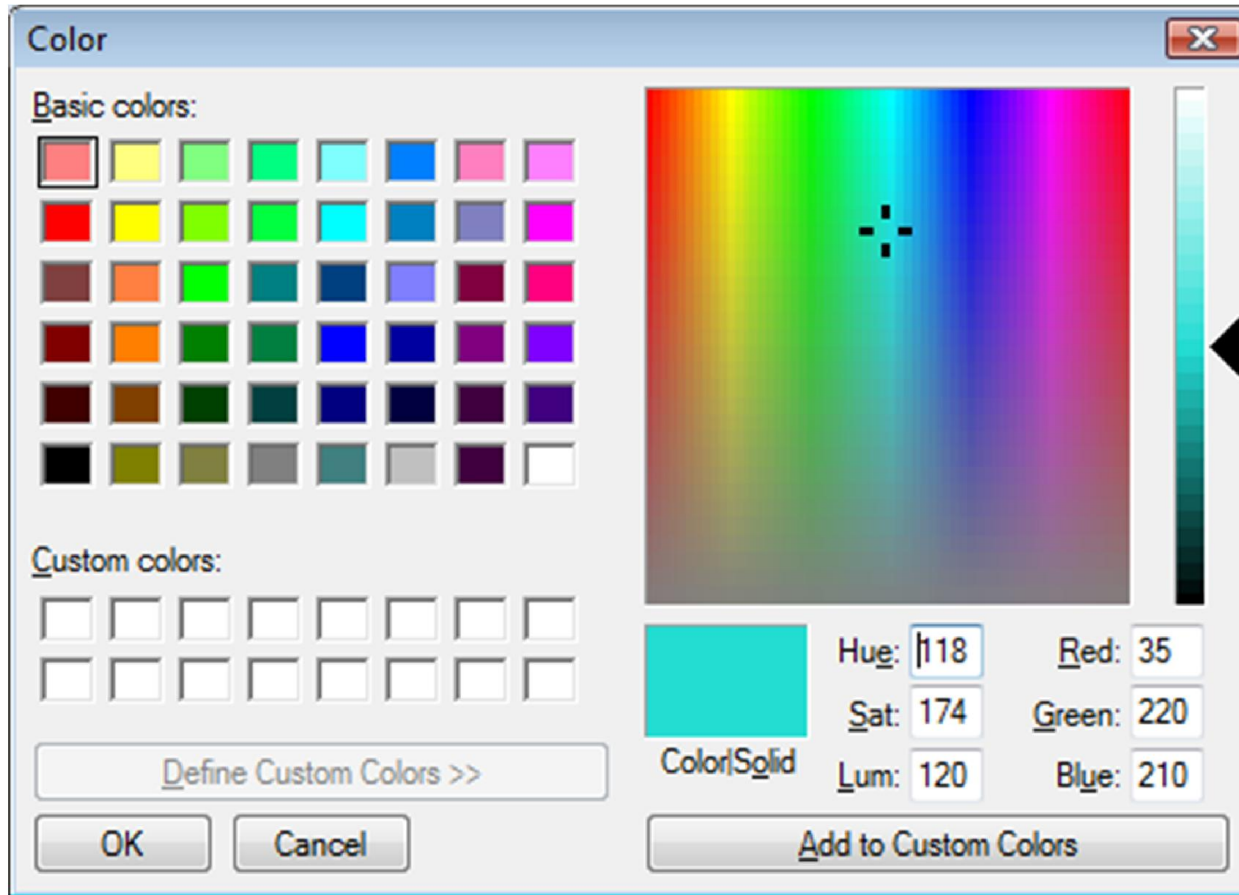
FontDialog

12

Thuộc tính	Giá trị
MaxSize, MinSize	Thông báo người dùng nhập nằm trong giới hạn 
ShowColor	Hiển thị combo cho phép lựa chọn màu (true, false)
ShowEffects	Hiển thị lựa chọn Effects: StrileOut, Underline (true, false)
ShowApply	Hiển thị nút Apply trong hộp thoại (true, false)

ColorDialog

13

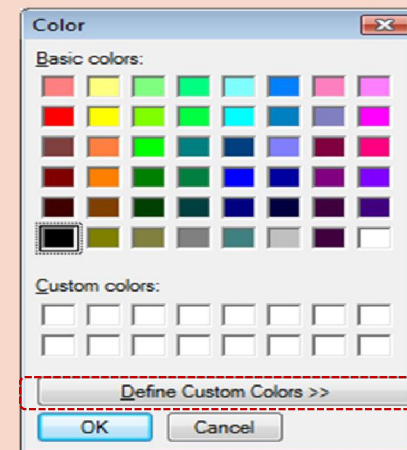


ColorDialog

14

- Hộp thoại cho phép lựa chọn màu

Thuộc tính	Giá trị
AllowFullOpen	Cho phép hiển thị đầy đủ hộp thoại (true, false)
FullOpen	Hiển thị hộp thoại đầy đủ cả Custom color (true, false)
ShowHelp	Hiển thị nút Help
Color	Get hoặc set giá trị màu cho hộp thoại



Advanced Control

15

TIMER

PROGRESSBAR

TOOLTIP

TRACKBAR

LISTVIEW

TREEVIEW

ERRORPROVIDER

NOTIFYICON

Timer

16

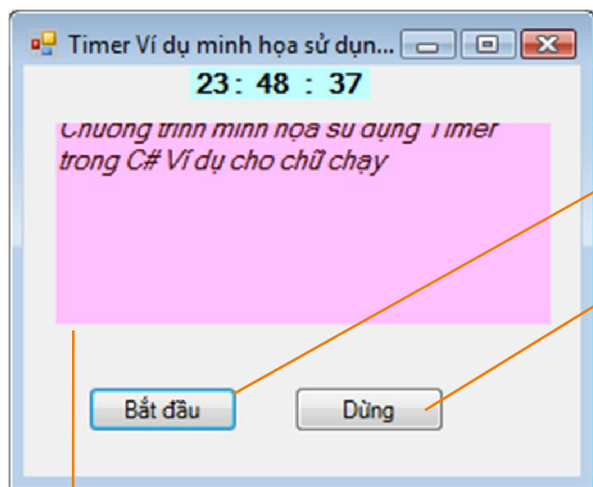
- Bộ định thời gian, thiết lập 1 khoảng thời gian xác nhận (interval) và khi hết khoảng thời gian đó Timer sẽ phát sinh sự kiện Tick
- Multitasking
- Autosave
- Demo version
- Game
- Quản lý và thông báo các trạng thái



Timer

17

Thuộc tính	Mô tả	Giá trị
Name	Tên của thuộc tính	Timer<ten>
Enable	Khởi động Timer?	True, False
Interval	Thời gian kích hoạt sự kiện Tick	Số nguyên, 1000=1 giây
Sự kiện		
Tick	Được kích hoạt khi thời gian đủ trong Interval	



lblGio, lblPhut, lblGiay

btnBatdau

btnDung

Sử dụng 3 timer :

- timer1 đặt interval:20
- timer2 đặt interval:20, Enable: True
- timer3 đặt interval:1000, Enable:True

textBox1, Panel

```
int h=0, m=0, s=0;
string str = "Ví dụ minh họa sử dụng Control Timer ",
s1="", s2="";
public frmTimer()
{
    InitializeComponent();
}
private void frmTimer_Load(object sender, EventArgs e)
{
    this.lblGio.Text = DateTime.Now.Hour.ToString();
    this.lblPhut.Text = DateTime.Now.Minute.ToString();
    this.lblGiay.Text = DateTime.Now.Second.ToString();
}
```

```
private void btnBatdau_Click(object sender, EventArgs e)
{
    this.timer1.Enabled = true;
}
private void btnDung_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
}
private void timer1_Tick(object sender, EventArgs e)
{
    textBox1.Top--;
    if (textBox1.Top + textBox1.Height == panel1.Top)
        textBox1.Top += textBox1.Height + panel1.Height;
}

private void timer2_Tick(object sender, EventArgs e)
{
    s1 = str.Substring(0, 1);
    s2 = str.Substring(1, str.Length - 1);
    str = s2 + s1;
    this.Text = str;
}
```

Cách 1: định nghĩa timer3_Tick

```
private void timer3_Tick(object sender, EventArgs e)
{
    h = int.Parse(this.lblGio.Text);
    m = int.Parse(this.lblPhut.Text);
    s = int.Parse(this.lbGiay.Text);
    if (s < 60) s++;
    else
    {
        s = 0;
        if (m < 60) m++;
        else
        {
            m = 0;
            if (h < 24) h++;
            else h = 0;
        }
    }
    this.lblGio.Text = h.ToString();
    this.lblPhut.Text = m.ToString();
    this.lbGiay.Text = s.ToString();
}
```

Cách 2: Có thể định nghĩa timer3_Tick như sau

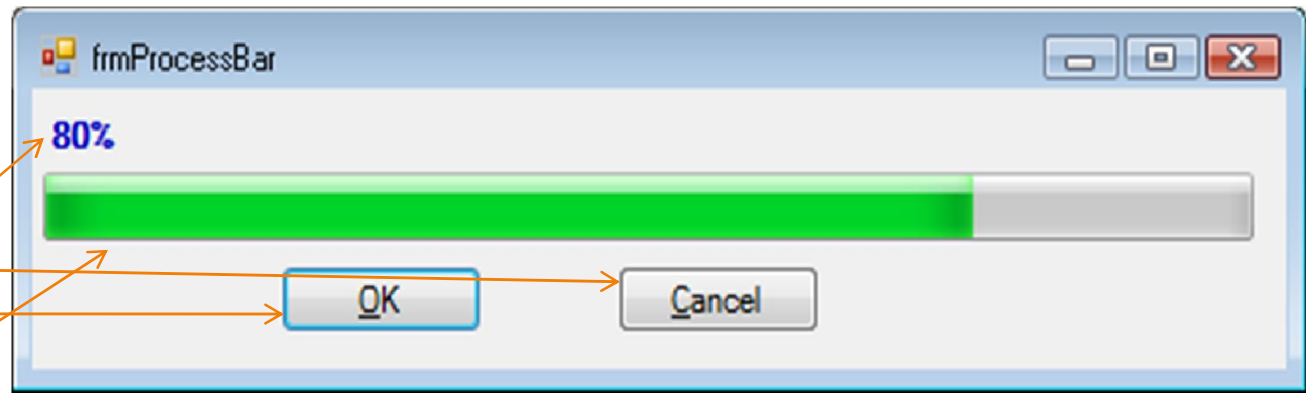
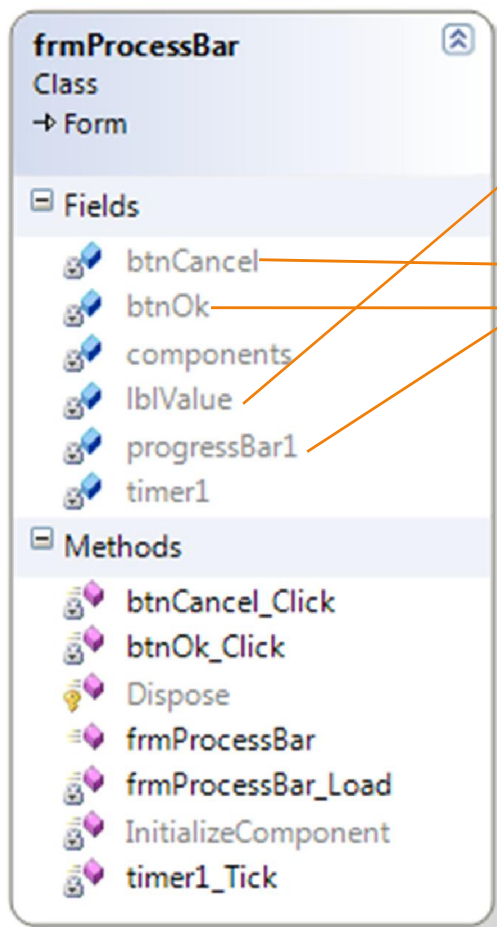
```
private void timer3_Tick(object sender, EventArgs e)
{
    this.lblGio.Text = DateTime.Now.Hour.ToString();
    this.lblPhut.Text = DateTime.Now.Minute.ToString();
    this.lbGiay.Text = DateTime.Now.Second.ToString();
}
```

ProgressBar

21

- Hiện thị tiến độ thực hiện của một công việc nào đó

Thuộc tính	Mô tả
Minimize	Get hoặc set giá trị min
Maximize	Get hoặc set giá trị max
Value	Get hoặc set giá trị của thanh ProgressBar
Step	Get hoặc set giá trị bước tăng khi gọi hàm PerformStep
Phương thức	Mô tả
PerformStep();	Thực hiện tăng giá trị của ProgressBar theo giá trị Step
Increment(int value);	Tăng vị trí hiện tại của tiến độ với giá trị xác định



```
private void btnOk_Click(object sender, EventArgs e)
{
    timer1.Start();
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (progressBar1.Value == progressBar1.Maximum)
    {
        timer1.Stop();
        MessageBox.Show("Đã chạy xong");
        progressBar1.Value = progressBar1.Minimum;
        this.lblValue.Text = "0%";
    }
    else
    {
        progressBar1.PerformStep();
        this.lblValue.Text = progressBar1.Value.ToString() + "%";
    }
}

private void btnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}
```

ToolTip

23

- 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 của control
- Thuộc tính:
 - IsBalloon: Bo góc của cửa sổ ToolTip
 - ToolTipIcon: Biểu tượng xuất hiện trên cửa sổ ToolTip
 - ToolTipTitle: Tiêu đề cửa sổ ToolTip
- Phương thức:
 - SetToolTip: Khai báo chuỗi xuất hiện trên cửa sổ ToolTip
 - GetToolTip: Nhận chuỗi xuất hiện trên cửa sổ ToolTip
 - RemoveAll: Loại bỏ tất cả các ToolTip cho các điều khiển

ToolTip

24

- Cách sử dụng:
 - Từ ToolBox kéo thả 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

The image illustrates the process of adding a tooltip to a Windows Forms application. It consists of several parts:

- Toolbox:** The Visual Studio Toolbox on the left shows various controls. The **ToolTip** control is highlighted with a red dashed box and an orange arrow pointing to the demo application.
- Demo ToolTip:** A central window titled "Demo ToolTip" contains a text box labeled "Họ tên" with a "Thêm" button, and a list box labeled "listBoxSV". An orange arrow points from the text box to the first Properties window.
- Properties (txtHoTen):** The first Properties window shows the selected control is **txtHoTen** (System.Windows.Forms.TextBox). The **ToolTip on toolTip1** property is set to "Nhập họ tên".
- Properties (listBoxSV):** The second Properties window shows the selected control is **listBoxSV** (System.Windows.Forms.ListBox). The **ToolTip on toolTip1** property is set to "Danh sách đã nhập".
- Final Application:** Two screenshots at the bottom show the final application. The left one shows a tooltip with a warning icon and the text "Thông báo Nhập họ tên" appearing over the text box. The right one shows the text box with the tooltip "Danh sách đã nhập" appearing over it.

ToolTip xuất hiện khi user di chuyển chuột vào vùng control

TrackBar

26

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

Thuộc tính	Mô tả
Minimize	Get hoặc set giá trị min
Maximize	Get hoặc set giá trị max
Value	Get hoặc set giá trị của thanh TrackBar

Properties

Minimum

Maximum

TickFrequency

TickStyle

Value

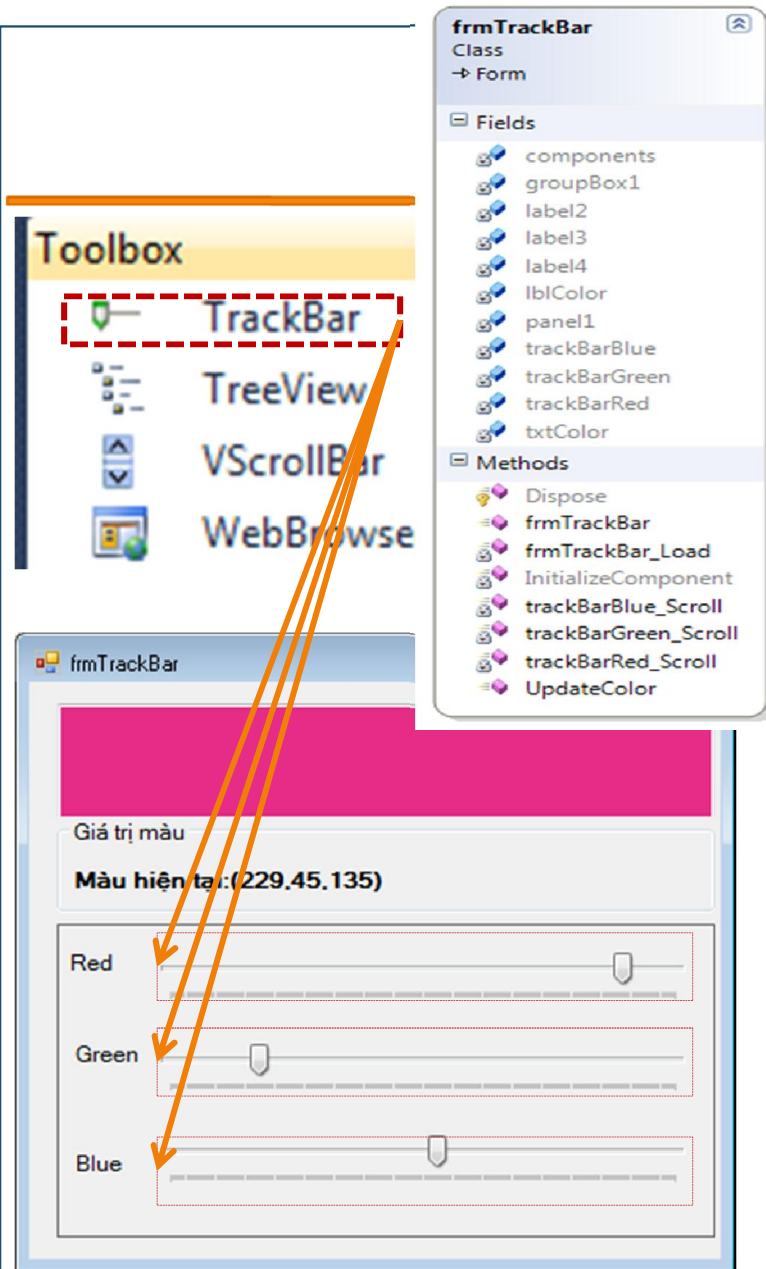


ValueChanged
Scroll

Methods

SetRange

TrackBar



```
private void frmTrackBar_Load(object sender, EventArgs e)
{
    this.trackBarGreen.Value = 0;
    this.trackBarRed.Value = 0;
    this.trackBarBlue.Value = 0;
    this.trackBarGreen.Maximum = 255;
    this.trackBarRed.Maximum = 255;
    this.trackBarBlue.Maximum = 255;
    UpdateColor();
}

public void UpdateColor()
{
    Color c = Color.FromArgb(trackBarRed.Value,
                             trackBarGreen.Value, trackBarBlue.Value);
    txtColor.BackColor = c;
    this.lblColor.Text = "Màu hiện tại: (" + trackBarRed.Value
                        + "," + trackBarGreen.Value
                        + "," + trackBarBlue.Value + ")";
}

private void trackBarRed_Scroll(object sender, EventArgs e)
{
    UpdateColor();
}

private void trackBarGreen_Scroll(object sender, EventArgs e)
{
    UpdateColor();
}

private void trackBarBlue_Scroll(object sender, EventArgs e)
{
    UpdateColor();
}
```

ScrollBar

Properties

HScrollBar and VScrollBar

SmallChange



Value

Events

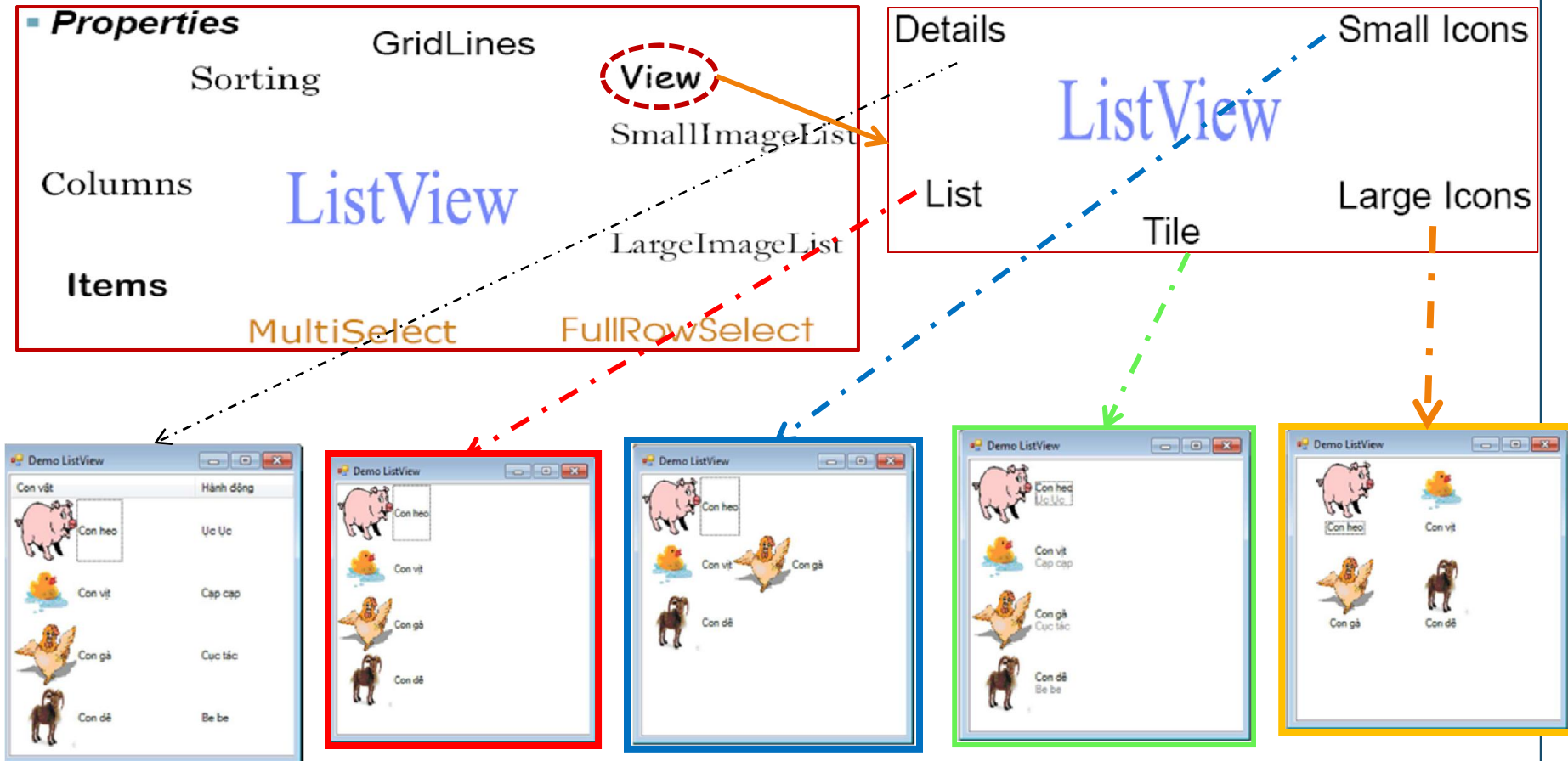
LargeChange

Scroll

ValueChanged

ListView

- Dạng control phổ biến hiện thị 1 danh sách item
 - Các item có thể có các item con gọi là subitem

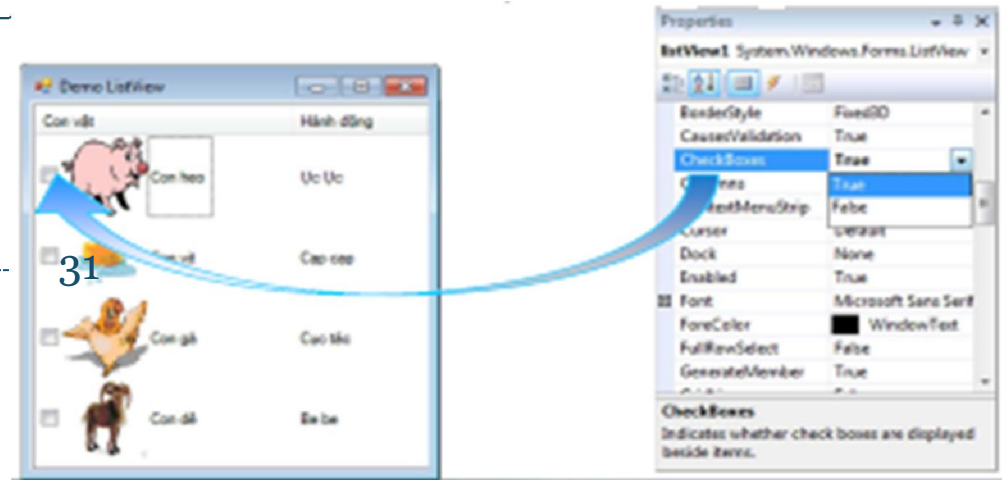


ListView

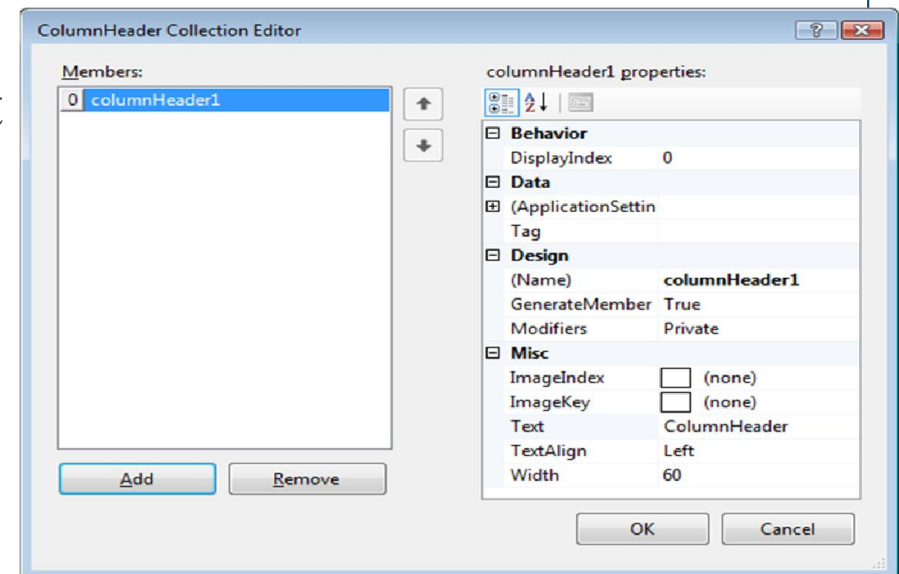
Thuộc tính	Mô tả	Giá trị
Name	Tên của thuộc tính	lv<ten>
View	Quy định cách hiển thị của điều khiển	<ul style="list-style-type: none"> - <i>LargeIcon</i> : với biểu tượng lớn - <i>SmallIcon</i> : với biểu tượng nhỏ. - <i>Details</i> : với các thông tin chi tiết. - <i>List</i> : dạng danh sách.
Items	Tập hợp các mục thông tin của ListView.	
CheckBoxes	Có hiển thị checkBoxes?	True, False
GridLines	Có hiển thị đường kẻ dòng cột khi View là Details hay không. Mặc định là không: True, False	
CheckedItems	Tập hợp các mục chọn khi có CheckBox (được chọn).	
SelectedItems	Tập hợp các mục chọn.	
LargeImageList SmallImageList	Nguồn hình ảnh sử dụng cho các mục trên ListView .	LargeImageList, SmallImageList

ListView

- Ví dụ: CheckBoxes = True



- Thuộc tính **Columns**: Tập hợp các cột hiển thị khi View là **Details**. Có các thuộc tính sau:
 - **Text** - tiêu đề cột
 - **TextAlign** - canh lề nội dung cột
 - **Width** - độ rộng cột.



ListView

32

Thuộc tính	Mô tả
ListViewItem: Các mục thông tin trên ListView được gọi là ListViewItem với các thông tin sau :	
BackColor, ForeColor	Màu nền, màu chữ.
Bounds	Các biên của ListViewItem .
Checked	Có được chọn hay không khi có CheckBox
Focused	Có đang nhận focus hay không?
Font	Font chữ hiển thị.
Selected	Có được chọn hay không khi không có CheckBox .
Text	Nội dung hiển thị (giá trị cột đầu tiên).
ImageIndex, ImageKey	Quy định thứ tự hay khóa của hình ảnh nào trên ImageList hiển thị trên ListItem .
SubItems	Các mục tin khác của ListViewItem .

ListView

33

- Một số thao tác trên **ListViewItem**:

- Thêm item vào listview (lvDanhSach):

```
ListViewItem lvitem = new ListViewItem(new string[] { "001", "Nguyễn Văn Hùng",  
"Nam", "08/08/1980", "0633777.888", "8.9" });  
this.lvDanhSach.Items.Add(lvitem);
```

```
ListViewItem lvitem1 = new ListViewItem("002");  
lvitem1.SubItems.Add("Lê Thị Lan Hương");  
lvitem1.SubItems.Add("Nữ");  
lvitem1.SubItems.Add("09/11/1987");  
lvitem1.SubItems.Add("0633333.555");  
lvitem1.SubItems.Add("5.7");  
lvDanhSach.Items.Add(lvitem1);
```

- Xóa 1 item ra khỏi listview

```
lvDanhSach.Items.Remove(lvitem);  
lvDanhSach.Items.RemoveAt(0);
```

- Xóa tất cả item

```
lvDanhSach.Items.Clear();
```

ListView

34

- Sự kiện:
 - **ItemCheck, ItemChecked**
 - ✦ Nếu **ListView** có **CheckBoxes = True**, sự kiện xảy ra khi trạng thái **Checked** của **ListItem** thay đổi.
 - **ItemSelectionChanged**
 - ✦ Sự kiện xảy ra khi một **ListItem** được chọn.

Ví dụ sử dụng MaskedTextBox, ComboBox, ListView, DateTimePicker

35

frmHocSinh

cboMaso txtHoten rdNam, rdNu

Ví dụ sử dụng MaskedTextBox, ComboBox, ListView

Mã số 005 Ngày Sinh Tuesday , September 01, 2009

Họ và tên Trần Văn Phước Số điện thoại (063) 333-3333

Giới tính ☐ Nữ ☒ Nam Điểm tổng kết 05.6

Thêm Xóa Trống

Mã Số	Họ và Tên	Giới Tính	Ngày Sinh	Số điện thoại	Điểm Tổng kết
001	Nguyễn Văn Hùng	Nam	08/08/1980	0633777.888	8.9
002	Trương Thị Lệ	Nữ	08/08/1980	0633999.888	8.9
003	Nguyễn Hoàng Nam	Nam	08/08/1980	0633666.888	8.9
004	Lê Thị Lan Hương	Nữ	09/11/1987	0633333.555	5.7

lvDanhSach

btnThem btnXoaTrang

btnXoa btnXoaAll btnThoat

Xóa item chọn Xóa Tất Cả Thoát

Code thêm dữ liệu cho lvDanhSach khi form load

36

```
private void frmHocSinh_Load(object sender, EventArgs e)
{
    ListViewItem lvitem = new ListViewItem(new string[] { "001",
    "Nguyễn Văn Hùng", "Nam", "08/08/1980", "0633777.888", "8.9" });
    this.lvDanhSach.Items.Add(lvitem);
    lvitem = new ListViewItem(new string[] { "002", "Trương Thị Lệ",
    "Nữ", "08/08/1980", "0633999.888", "8.9" });
    this.lvDanhSach.Items.Add(lvitem);
    lvitem = new ListViewItem(new string[] { "003", "Nguyễn Hoàng
    Nam", "Nam", "08/08/1980", "06336666.888", "8.9" });
    this.lvDanhSach.Items.Add(lvitem);
    lvitem = new ListViewItem("004");

    lvitem.SubItems.Add("Lê Thị Lan Hương");
    lvitem.SubItems.Add("Nữ");
    lvitem.SubItems.Add("09/11/1987");
    lvitem.SubItems.Add("0633333.555");
    lvitem.SubItems.Add("5.7");
    lvDanhSach.Items.Add(lvitem);
}
```

Xử lý khi Click vào btnThem

37

```
private void btnThem_Click(object sender, EventArgs e)
{
    string maso, hoten;
    maso = this.cboMaSo.Text;
    hoten = this.txtHoTen.Text;
    if (maso.Length>0 && hoten.Length>0)
    {
        ListViewItem lvitem = new ListViewItem(this.cboMaSo.Text);
        lvitem.SubItems.Add(this.txtHoTen.Text);
        if (this.rbNam.Checked == true)
            lvitem.SubItems.Add("Nam");
        else
            lvitem.SubItems.Add("Nữ");
        lvitem.SubItems.Add(this.dtpNgaysinh.Value.ToString());
        lvitem.SubItems.Add(this.mtbSodt.Text);
        lvitem.SubItems.Add(this.mtbDiem.Text);
        this.lvDanhSach.Items.Add(lvitem);
    }
    else
        MessageBox.Show("Hãy nhập mã số và Họ tên");
}
```

Thêm dữ liệu từ thông tin học
sinh xuống lvDanhSach

Xử lý khi Click vào btnXoaTrang

38

```
private void btnXoaTrang_Click(object sender, EventArgs e)
{
    this.cboMaSo.Text = "";
    this.txtHoTen.Text = "";
    this.dtpNgaysinh.Text = "";
    this.mtbSodt.Text = "(---) ---.---";
    this.mtbDiem.Text = "__._";
}
```

Xóa trắng thông tin học sinh

```
private void btnXoa_Click(object sender, EventArgs e)
{
    int i = this.lvDanhSach.SelectedItems.Count - 1;
    while (i >= 0)
    {
        lvDanhSach.Items.RemoveAt(lvDanhSach.SelectedIndex[i]);
        //lvDanhSach.Items.Remove(lvDanhSach.SelectedItems[i]);
        i--;
    }
}
```

Xóa những Item chọn trên lvDanhSach

```
private void btnXoaAll_Click(object sender, EventArgs e)
{
    this.lvDanhSach.Items.Clear();
    this.lvDanhSach.Items.Clear();
}
```

Xóa toàn bộ item trên lvDanhSach

Treeview

39

- Dùng để trình bày danh sách phần tử phân cấp theo từng node hình cây. Kéo control TreeView thả vào Form.

The screenshot illustrates the steps to add and configure a TreeView control in a Windows Forms application:

- Toolbox:** The **TreeView** control is highlighted with a red dashed box.
- Properties Window:** The **treeView1** properties are shown. The **Nodes** property is highlighted with a red dashed box, indicating it is a **(Collection)**.
- TreeNode Editor:** A window showing the tree structure. The root node is **Loai Hoa**, with child nodes **Hoa Lan**, **Hoa Hong**, **Trai Cay**, **Trai Oi**, and **Trai Nho**.
- Members:** A list of image files (e.g., **amaryllis.ico**, **IDK8_1LA.GIF**) is shown, with the **amaryllis.ico** file selected.
- Demo TreeView Form:** The final form showing the TreeView control with the configured tree structure. The **imageList1** control is also visible at the bottom left.

Treeview

40

Thuộc tính	Mô tả
CheckBoxes	True, false: Cho phép hiển thị CheckBox trên Treeview.
ImageIndex	Get hoặc set giá trị index image của tree, index mặc định hiển thị trên tree node
ImageList	Get hoặc set 1 imagelist gắn với Treeview
Nodes	Lấy tập node của đối tượng tree node trên treeview
PathSeparator	Ký tự phân cách đường dẫn.
SelectedNode	Get hoặc set Node được chọn
ShowPlusMinus	Gets hoặc set chỉ ra trạng thái mở rộng node cha bằng dấu '+' hoặc '-'. Mặc định true
TopNode	Trả về TreeNode đầu tiên trên treeview

Treeview

41

Ví dụ minh họa ToolStrip, TreeView

- Loai hoa
 - Hoa Lan
 - Hoa Hong**
- Trái Cay
 - Trái oi
 - Trái nhỏ

Buttons: CollapseAll, Edit Label, Remove, Cancel

Status Bar: \Loai hoa\Hoa Hong

frmTreeView
Class
→ Form

Fields

- btnCancel
- btnCollapseAll
- btnEditLable
- btnRemove
- components
- imageList1
- mySelectedNode
- statusStrip1
- toolStripStatusLabel1
- treeView1

Methods

- btnCancel_Click
- btnCollapseAll_Click
- btnEditLable_Click
- btnRemove_Click
- Dispose
- frmTreeView
- frmTreeView_Load
- InitializeComponent
- treeView1_AfterLabelEdit
- treeView1_MouseDown

Treeview

42

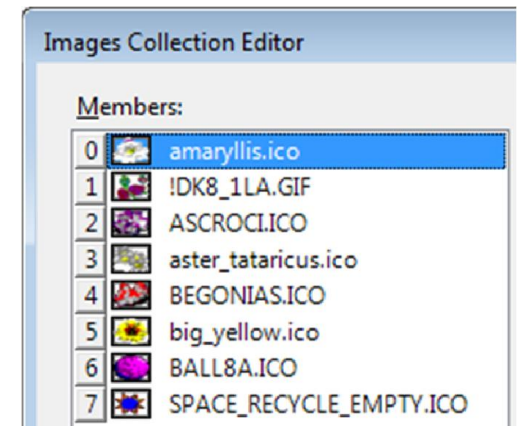
```
private void frmTreeView_Load(object sender, EventArgs e)
{
    TreeNode tn, subtn;
    //this.treeView1.CheckBoxes = true;
    tn = this.treeView1.Nodes.Add("Loai hoa");
    tn.ImageIndex = 0;
    subtn = new TreeNode("Hoa Lan");
    subtn.ImageIndex = 0;
    tn.Nodes.Add(subtn);

    subtn = new TreeNode("Hoa Hong", 1, 2);
    tn.Nodes.Add(subtn);

    tn = this.treeView1.Nodes.Add("Trai Cay");
    tn.ImageIndex = 6;

    subtn = new TreeNode("Trai oi");
    subtn.ImageIndex = 7;
    tn.Nodes.Add(subtn);

    subtn = new TreeNode("Trai nho");
    subtn.ImageIndex = 6;
    tn.Nodes.Add(subtn);
}
```



TreeNode Class

43

- Mỗi node trong Treeview là một đối tượng TreeNode
- Hàm khởi tạo của TreeNode:
 - `TreeNode(string label);`
 - `TreeNode(string label, int imageIndex, int selectedIndex);`

TreeNode Class

Thuộc tính	Mô tả
FirstNode	Trả về Node con đầu tiên của Node hiện tại
ImageIndex	Get hoặc set giá trị index image của Node.
Index	Trả về vị trí của Node hiện tại trong tập Node hoặc của TreeView
IsEditing	Trả về kết quả liệu Node đang được chỉnh sửa?
IsExpanded	Trả về kết quả liệu các node con của Node hiện tại có được hiển thị?
IsSelected	Trả về kết quả liệu node hiện tại được chọn?
IsVisible	Gets hoặc set giá trị Font cho Node
Nodes	Lấy tập đối tượng tree node của node hiện tại
Phương thức	Mô tả
BeginEdit	Bắt đầu cho việc Edit nhãn của Node
Collapse	Đóng tất cả các node con đang hiển thị của Node hiện tại
ExpandAll	Mở rộng (hiển thị) tất cả các Node con

```
private void treeView1_MouseDown(object sender, MouseEventArgs e)
{
    mySelectedNode = treeView1.GetNodeAt(e.X, e.Y);
    this.toolStripStatusLabel1.Text="\\\"+ mySelectedNode.FullPath;
}
private void treeView1_AfterLabelEdit(object sender, NodeLabelEventArgs e)
{
    if (e.Label != null)
    {
        if (e.Label.Length > 0)
        {
            if (e.Label.IndexOfAny(new char[] { '@', '.', ',', '!', '!' }) == -1)
                e.Node.EndEdit(false);
            else
            {
                e.CancelEdit = true;
                MessageBox.Show("Nhãn không có giá trị:\n" +
                    "Ký tự sai: '@', '.', ',', '!', '!',",
                    "Node Label Edit");
                e.Node.BeginEdit();
            }
        }
        else
        {
            e.CancelEdit = true;
            MessageBox.Show("Nhãn không có giá trị.\nNhãn không thể có khoảng trắng",
                "Node Label Edit");
            e.Node.BeginEdit();
        }
        this.treeView1.LabelEdit = false;
    }
}
```

```
private void btnCollapseAll_Click(object sender, EventArgs e)
{
    this.treeView1.CollapseAll();
}

private void btnCancel_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnEditLable_Click(object sender, EventArgs e)
{
    if (mySelectedNode != null && mySelectedNode.Parent != null)
    {
        treeView1.SelectedNode = mySelectedNode;
        treeView1.LabelEdit = true;
        if (!mySelectedNode.IsEditing)
            mySelectedNode.BeginEdit();
        else
            mySelectedNode.EndEdit(true);
    }
}

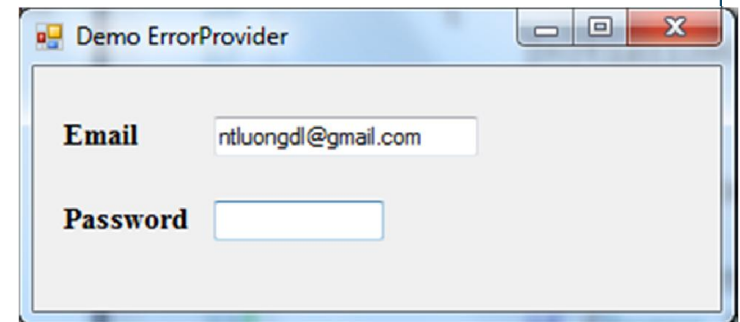
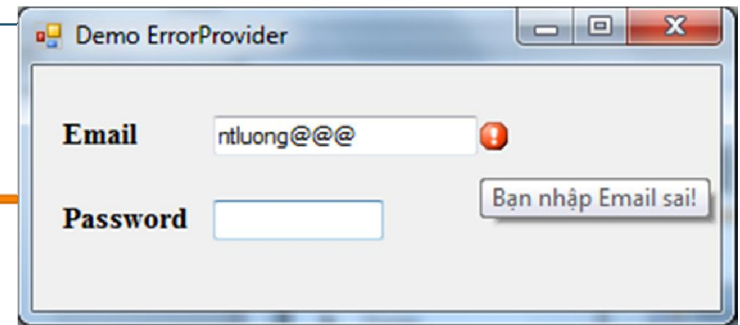
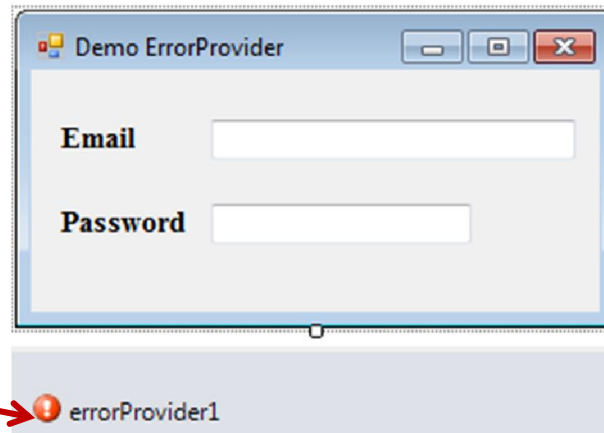
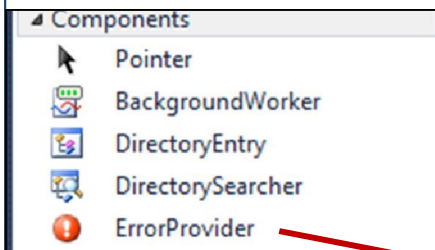
private void btnRemove_Click(object sender, EventArgs e)
{
    if (mySelectedNode != null && mySelectedNode.Parent != null)
    {
        treeView1.SelectedNode = mySelectedNode;
        treeView1.Nodes.Remove(mySelectedNode);
        this.toolStripStatusLabel1.Text = "";
    }
}
```

ErrorProvider

47

- Dùng để thông báo lỗi nhập dữ liệu trên Form
- Thuộc tính:
 - *Icon*: Biểu tượng xuất hiện cạnh điều khiển
 - *BlinkRate*: Tốc độ nhấp nháy (n/1000 giây)
 - *BlinkStyle*: BlinkIfDifferentError, AlwaysBlink
- Phương thức:
 - *SetError*: Khai báo chuỗi báo lỗi cho điều khiển
 - *GetError*: nhận chuỗi báo lỗi cho điều khiển
 - *Clear*: Xóa bỏ các chuỗi báo lỗi cho tất cả các điều khiển trên Form.

ErrorProvider

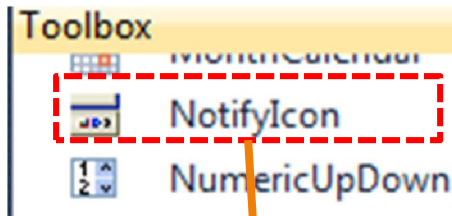
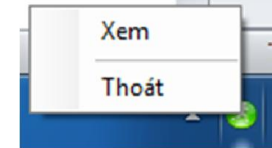
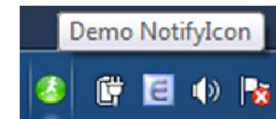


```
private void txtEmail_Validated(object sender, EventArgs e)
{
    string pattern = @"^(\w+)@((yahoo)|(gmail))\.com$";
    if (Regex.IsMatch(this.txtEmail.Text, pattern))
    {
        this.errorProvider1.SetError(txtEmail, "");
    }
    else
        this.errorProvider1.SetError(txtEmail, "Bạn nhập Email sai!");
}
```


NotifyIcon

49

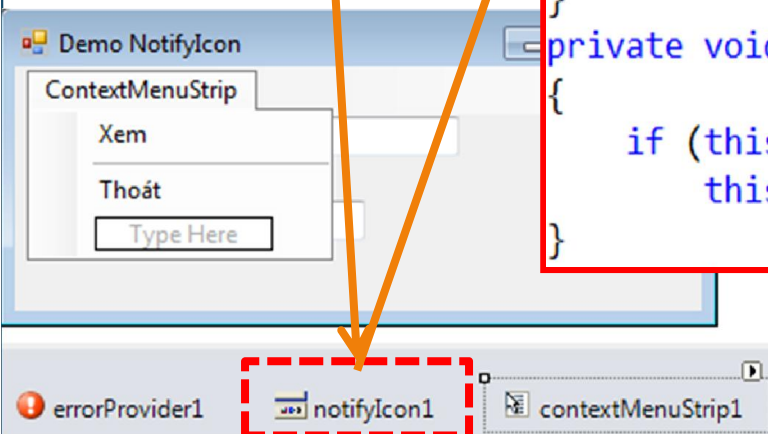
- Trình bày đối tượng trên Taskbar
- Thuộc tính: Icon, Text, ContextMenuStrip



```
private void xemToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Visible = true;
    this.WindowState = FormWindowState.Normal;
}

private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void frmErrorProvider_Resize(object sender, EventArgs e)
{
    if (this.WindowState == FormWindowState.Minimized)
        this.Hide();
}
```



User Control

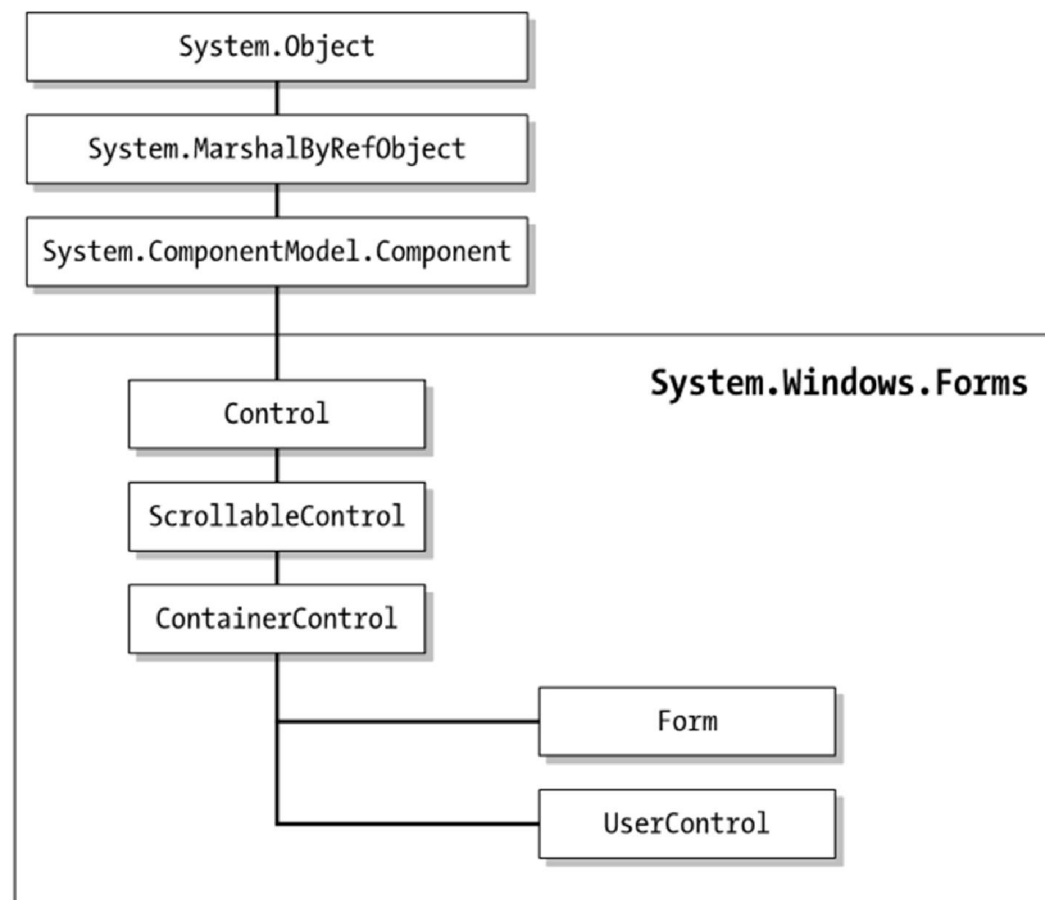
50

**GIỚI THIỆU USER CONTROL
TẠO USER CONTROL
VÍ DỤ**

Giới thiệu User Control

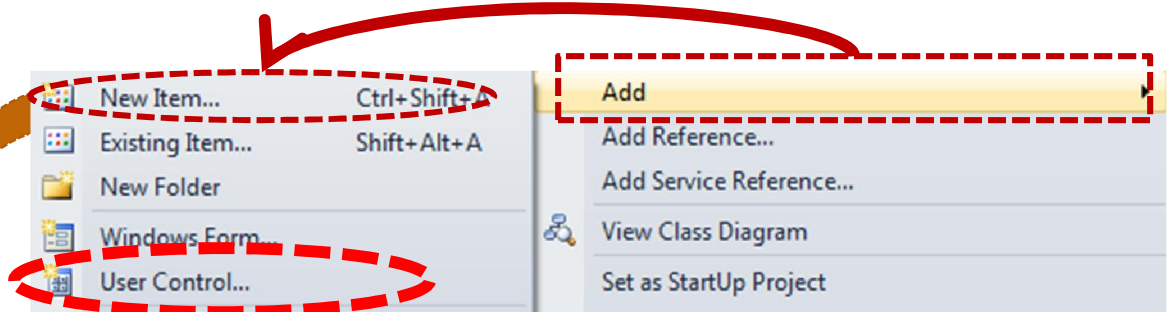
51

- Kế thừa từ lớp *System.Windows.Forms.UserControl*
- Trên thực tế, xây dựng UserControl tương tự như Form

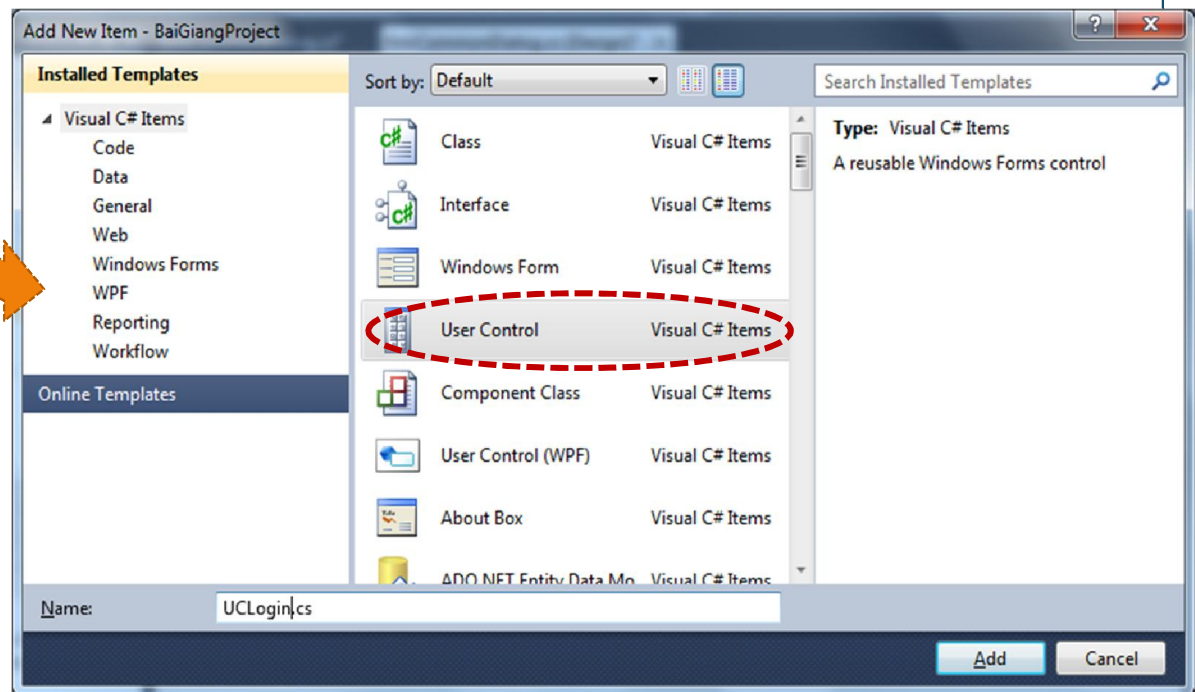


Tạo user control

- Bước 1: Click chuột phải vào project → Add → User Control (hoặc New item)



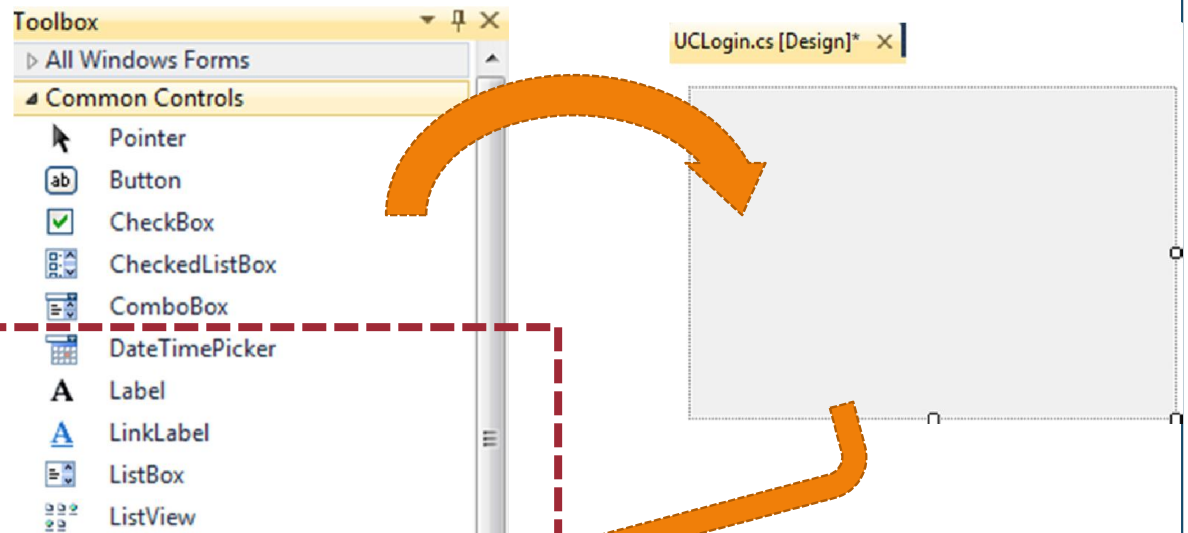
UCLogin.cs [Design]* X



Tạo user control

53

- Bước 2: kéo các control cơ bản từ Toolbox vào để thiết kế cho control mới

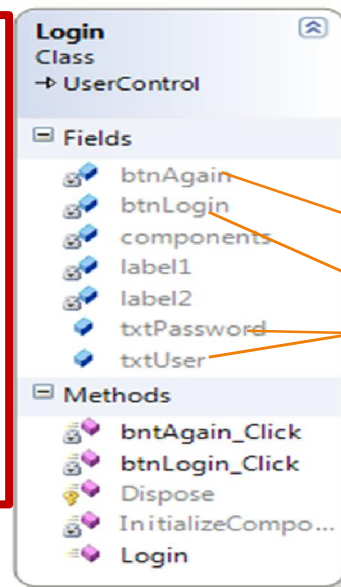


```
using System;
...
namespace BaiGiangProject
{
    public partial class UCLogin : UserControl
    {
        public UCLogin()
        {
            InitializeComponent();
        }
    }
}
```

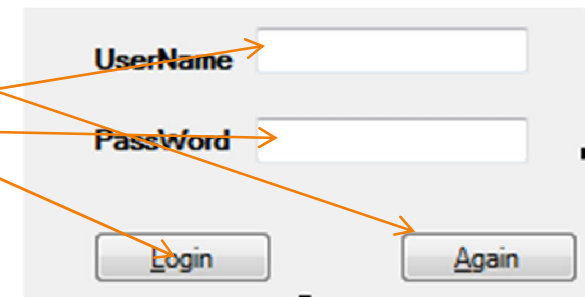
Ví dụ

Tạo user control Login gồm:

- TextBox: UserName, Password
- Button: Login, Again



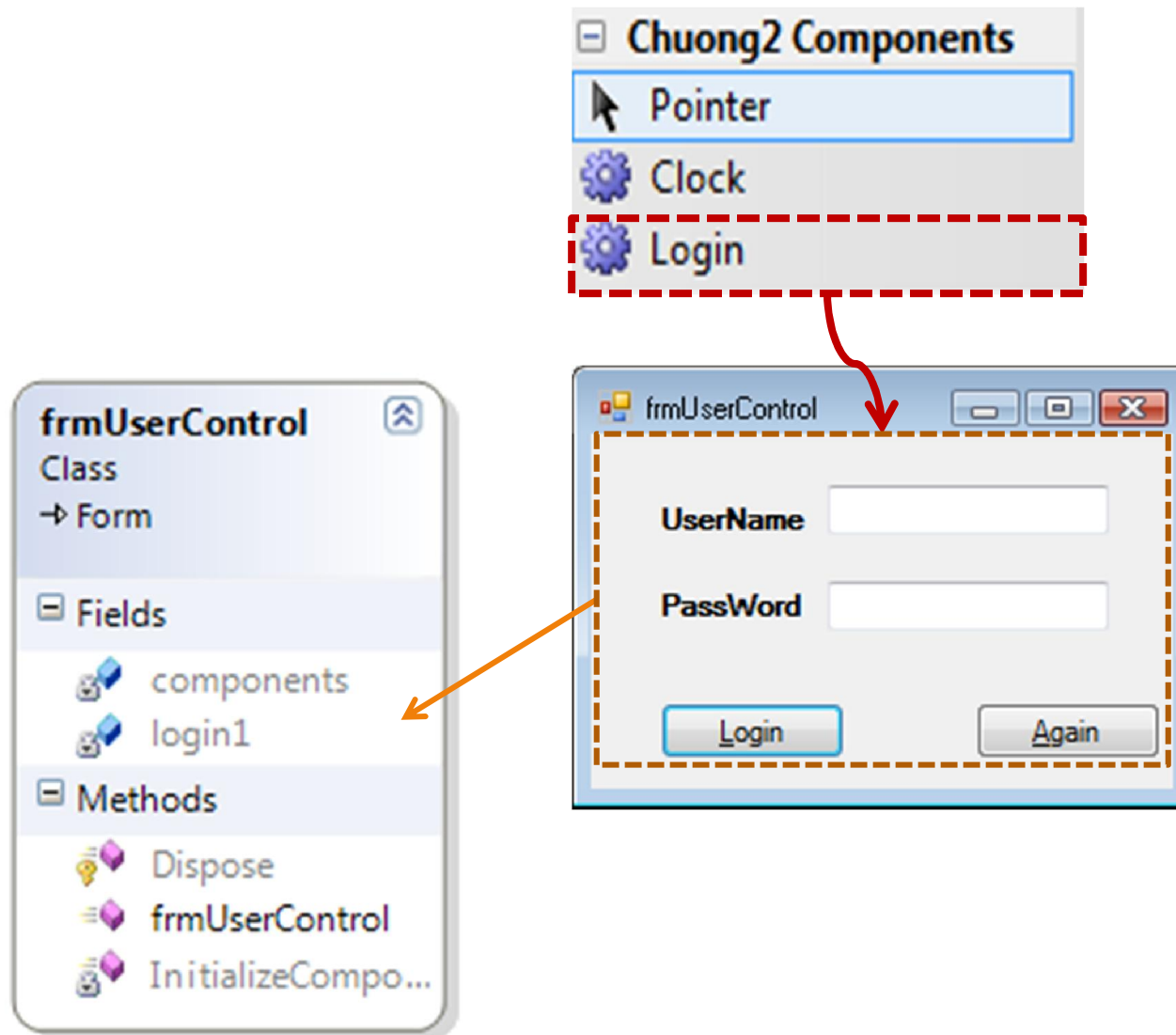
Create UserControl: Login



```
private void btnLogin_Click(object sender, EventArgs e)
{
    if (txtUser.Text == "user1" && txtPassword.Text == "123456")
        MessageBox.Show("Login success!");
    else
        MessageBox.Show("User or Password incorrect!");
}

private void bntAgain_Click(object sender, EventArgs e)
{
    txtPassword.Text = "";
    txtUser.Text = "";
}
```

Ví dụ



Q&A

- Containes: Group, panel, tabcontrol, Splitcontain
- NumberUpDown, DomainUpDown. ImageList, PictureBox,
- ListView
- Menu
- Dialogs: