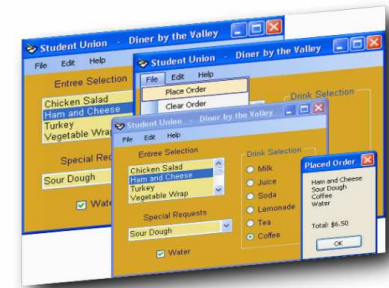


# SDI & MDI



ThS. Nguyễn Hà Giang  
Khoa CNTT - Hutech

# Nội Dung

- Tổng quan ứng dụng Document Interface
- Single Document Interface (SDI)
- Multiple Document Interface (MDI)
- GUI hỗ trợ
  - MenuStrip
  - ContextMenuStrip
  - ToolStrip
  - StatusStrip
- Ứng dụng MDI tổng hợp

# Ứng dụng document Interface

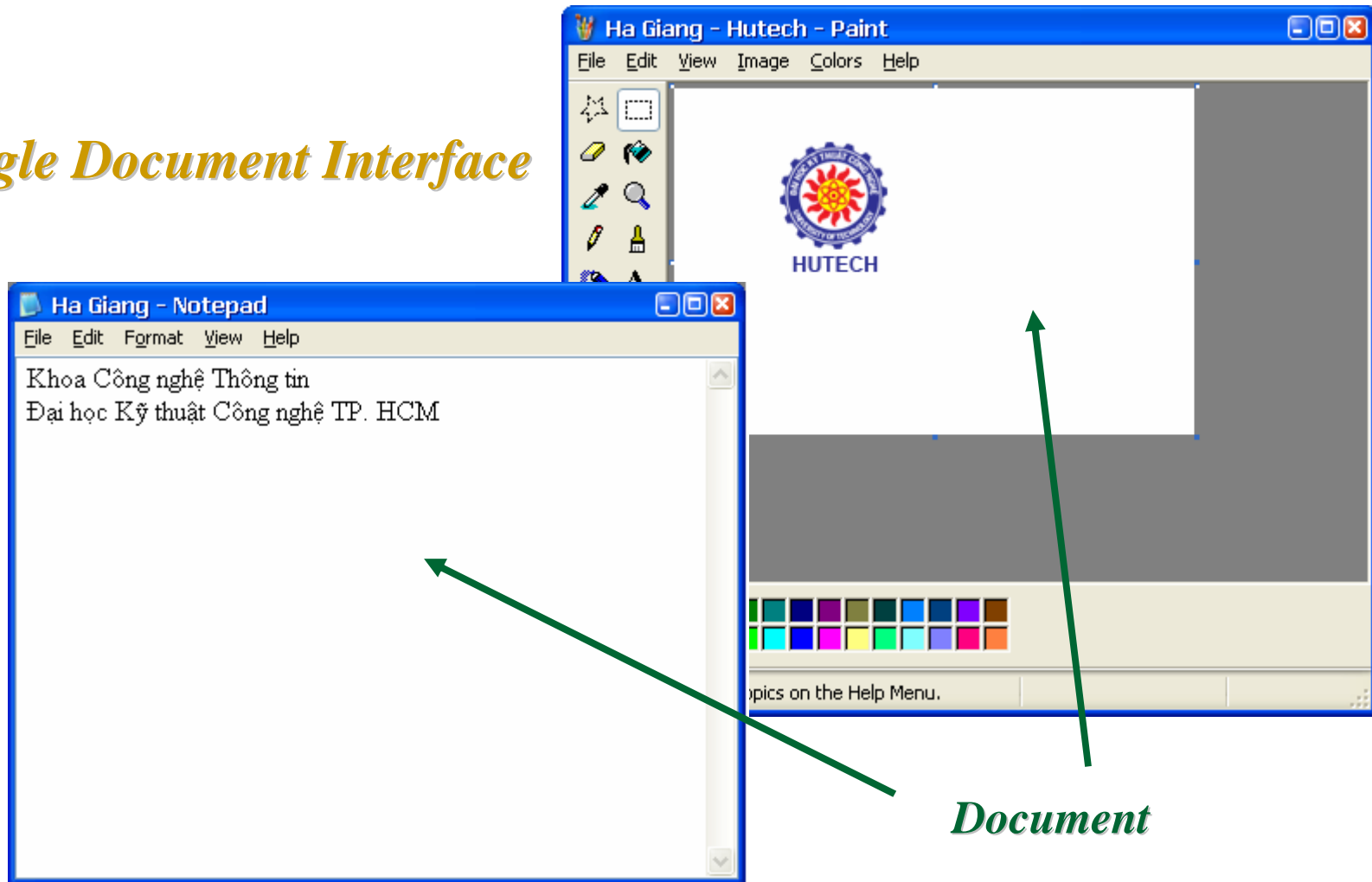
- **Dạng ứng dụng làm việc với các document**
  - Document chứa nội dung dữ liệu, được lưu trữ trên file đặc trưng, hoặc trong CSDL.
- **Các dạng document**
  - Document soạn thảo văn bản
  - Document xử lý ảnh đồ họa
  - Document bảng tính
  - Document làm việc csdl
  - ...
- **Ứng dụng dạng DI sẽ cung cấp các chức năng**
  - Mở file (hoặc CSDL), xử lý nội dung và lưu file (hoặc CSDL)
- **Có 2 dạng ứng dụng DI**
  - Single Document Interface
  - Multiple Document Interface

# Single Document Interface

- Chỉ hỗ trợ một document hoặc một cửa sổ tại một thời điểm.
- Dạng ứng dụng như NotePad, MS Paint với các đặc tính xử lý ảnh và văn bản giới hạn.
- Để làm việc với nhiều tài liệu thì phải mở các instance khác của ứng dụng.
- Việc quản lý ứng dụng SDI khá đơn giản.

# Single Document Interface

## *Single Document Interface*

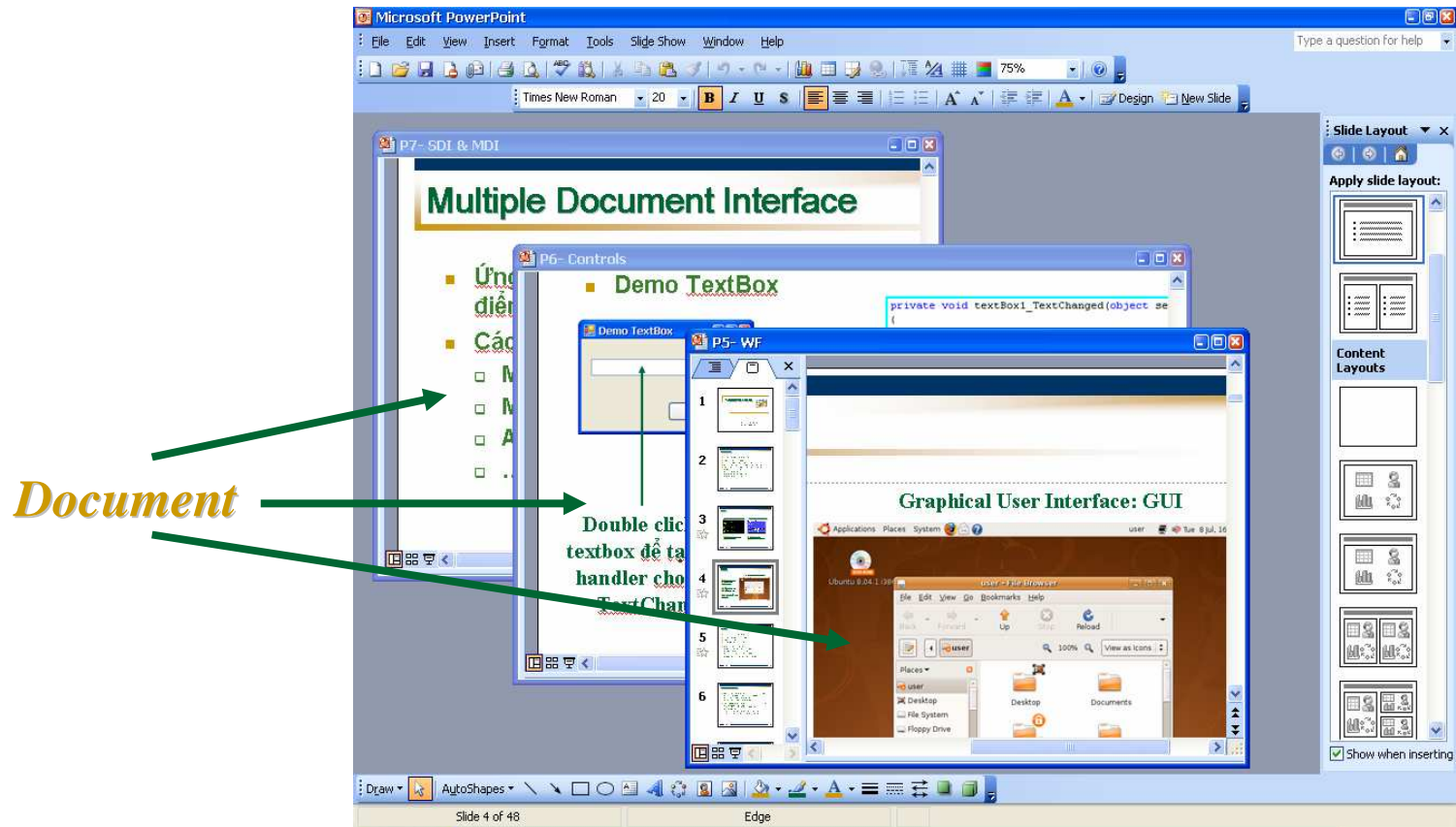


# Multiple Document Interface

- Ứng dụng cho phép xử lý nhiều document tại một thời điểm.
- Các ứng dụng hiện tại thường là dạng MDI
  - MS Office
  - MS Studio.NET
  - Adobe Photoshop
  - ...
- Ứng dụng MDI thường phức tạp và chức năng xử lý đa dạng
  - Xử lý một cùng một dạng document
    - MS Word, Adobe Photoshop
  - Xử lý nhiều dạng document
    - Visual Studio 6: code, design, image...

# Multiple Document Interface

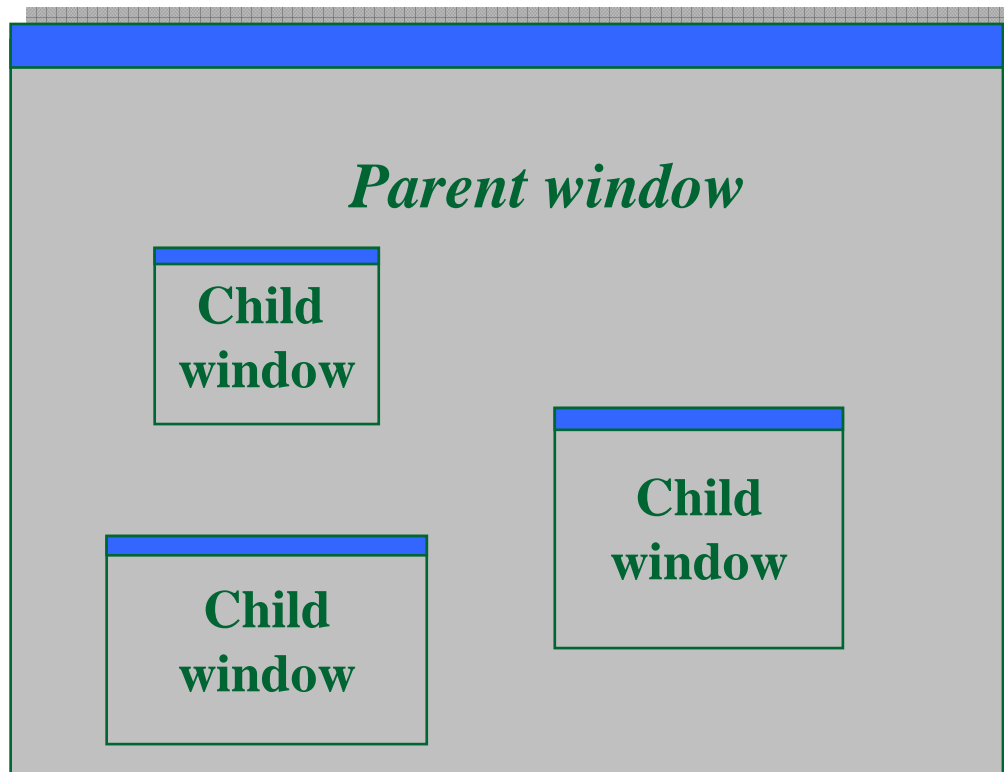
## Ứng dụng MDI – MS PowerPoint



# Multiple Document Interface

- MDI Parent Window – MDI Child Window

## MDI Application



Chỉ có duy nhất  
một parent  
windows

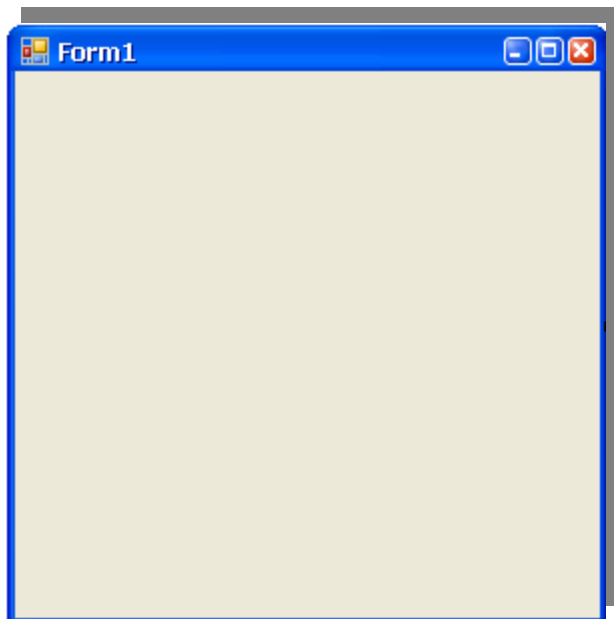
Có nhiều child  
windows

# Multiple Document Interface

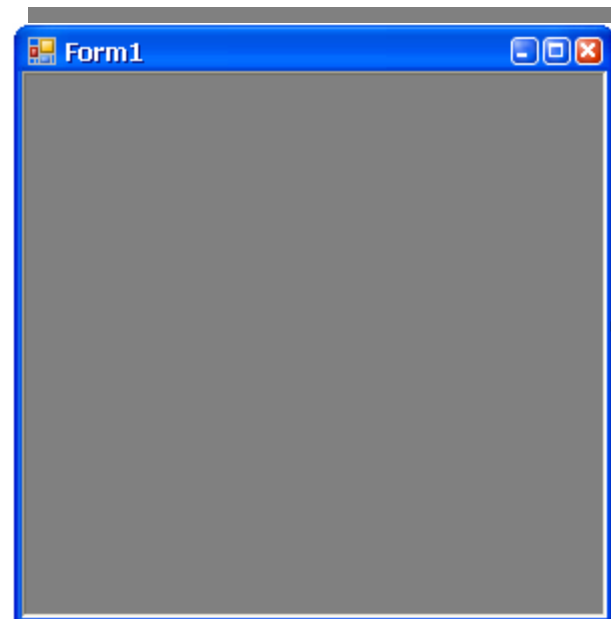
- Cách tạo MDI Form

- Tạo form và thiết lập thuộc tính `IsMdiContainer = true`, khi đó form sẽ thay đổi cách thể hiện

SDI Form



MDI Form

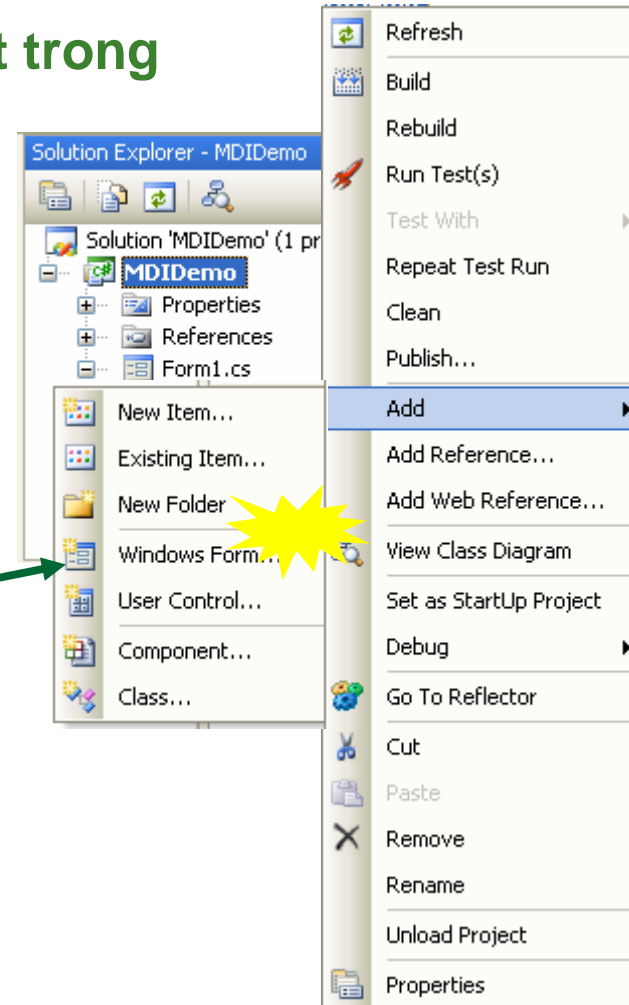


# Multiple Document Interface

- Tạo một form con để add vào form chính

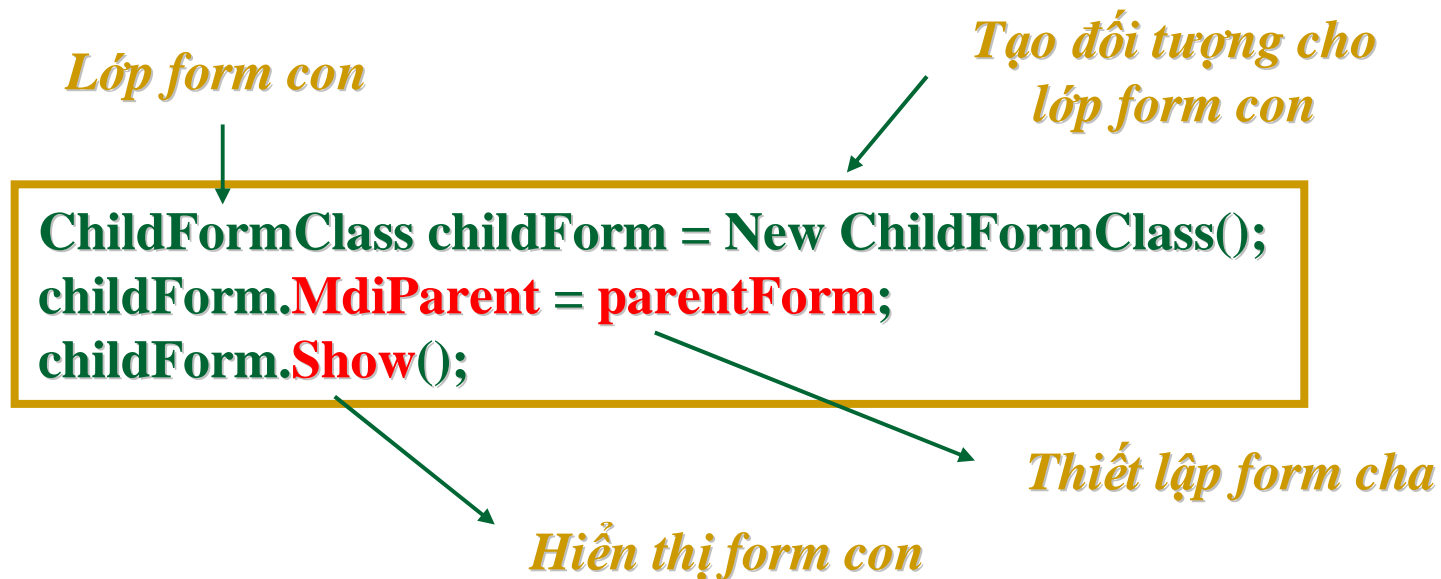
- Click chuột phải lên tên project trong Solution Explorer
- Chọn Add-> Windows Form
- Trong cửa sổ Add New Item
  - Chọn tên của form con
- Thiết kế form con theo yêu cầu của ứng dụng.

*Tạo form con*



# Multiple Document Interface

- Add form con vào form chính
  - Tạo đối tượng của form con
  - Thiết lập thuộc tính MdiParent của form con là form chính
  - Gọi phương thức Show của form con
- Một cách tổng quát để add form con vào form chính



# Multiple Document Interface

## *Thuộc tính, phương thức và sự kiện*

### *Thuộc tính của MDI child*

IsMdiChild	Cho biết form là child hay không
MdiParent	Xác định form cha của form

### *Thuộc tính của MDI parent*

ActiveMdiChild	Trả về form con đang active
IsMdiContainer	Xác định form là MDI parent hoặc không
MdiChildren	Trả về mảng chứa các form con

### *Phương thức*

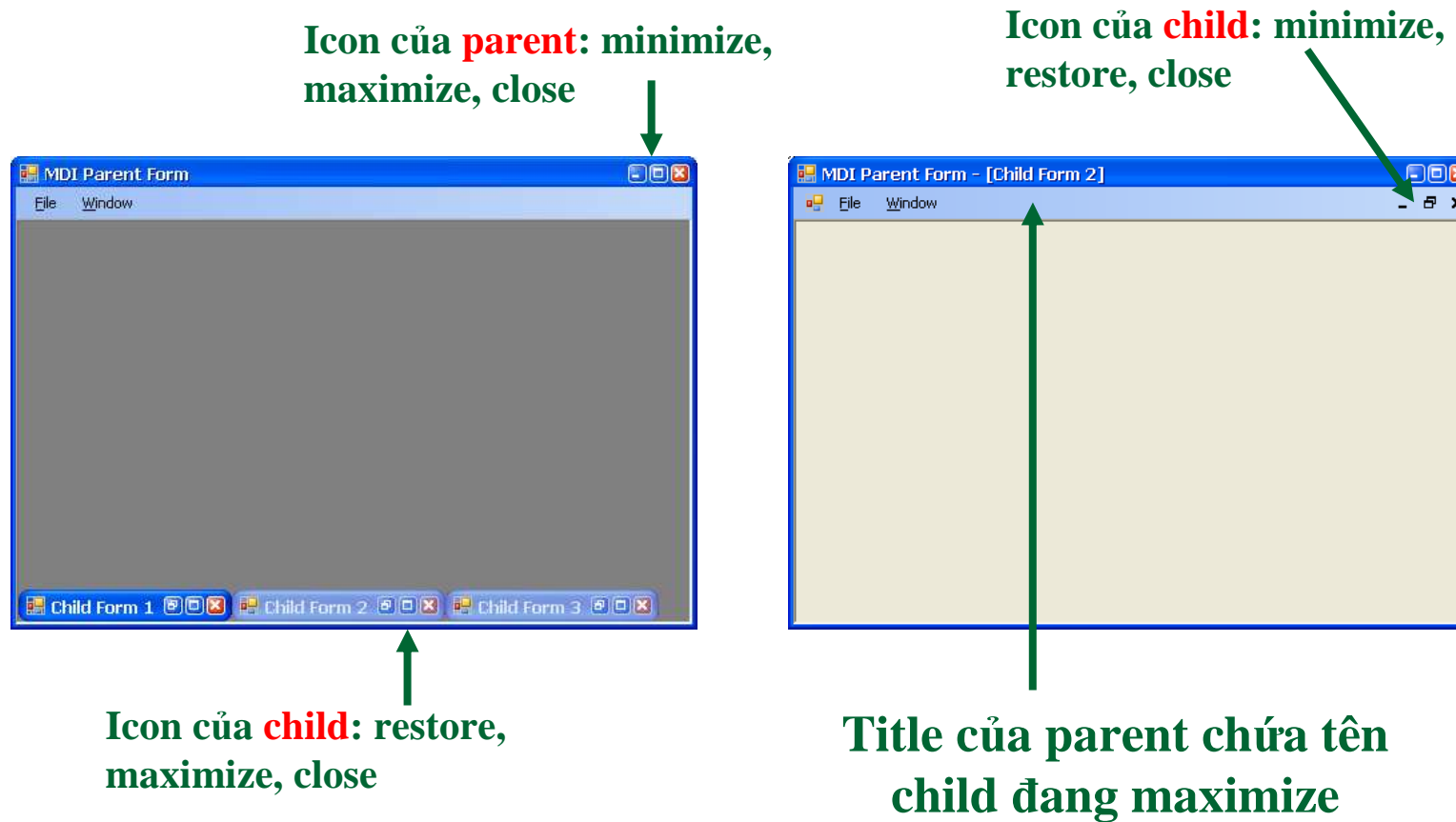
LayoutMdi	Xác định cách hiển thị các form con
-----------	-------------------------------------

### *Sự kiện*

MdiChildActivate	Xuất hiện khi child form đóng hoặc active
------------------	---

# Multiple Document Interface

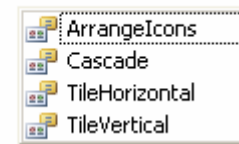
- Trạng thái Minimize, Maximize và Close



# Multiple Document Interface

- Sắp đặt các child form
  - Sử dụng phương thức `LayoutMdi`
  - Tham số là các giá trị `MdiLayout` enumeration

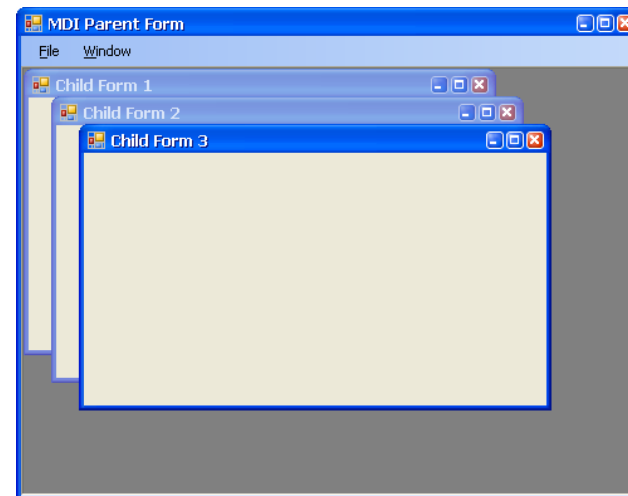
`MdiLayout.`



## ArrangeIcons



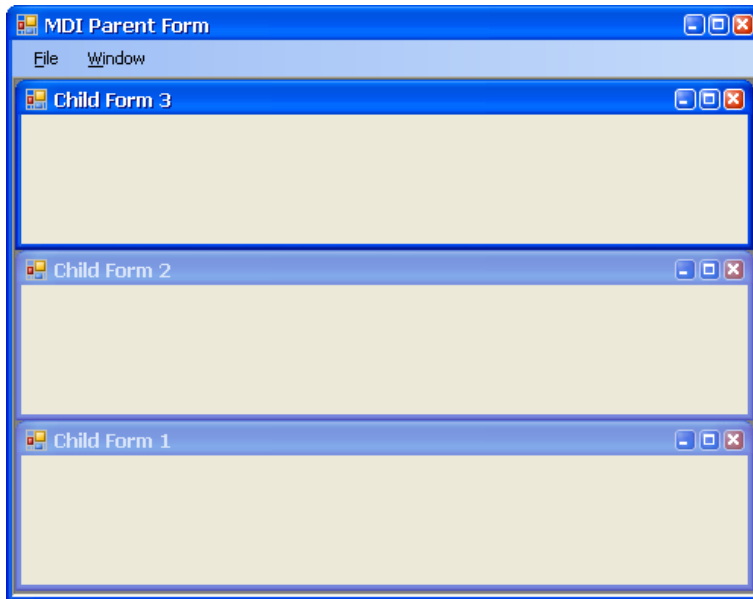
## Cascade



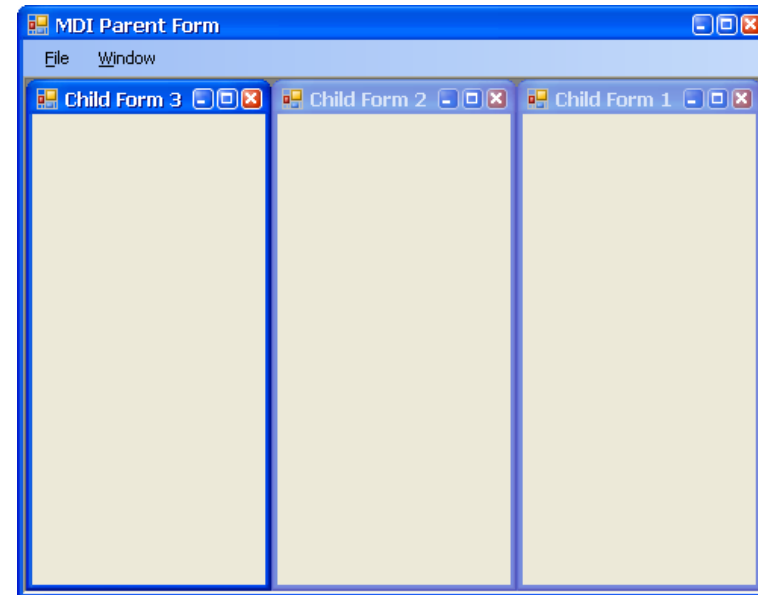
# Multiple Document Interface

- Sắp đặt các child form

**TileHorizontal**



**TileVertical**

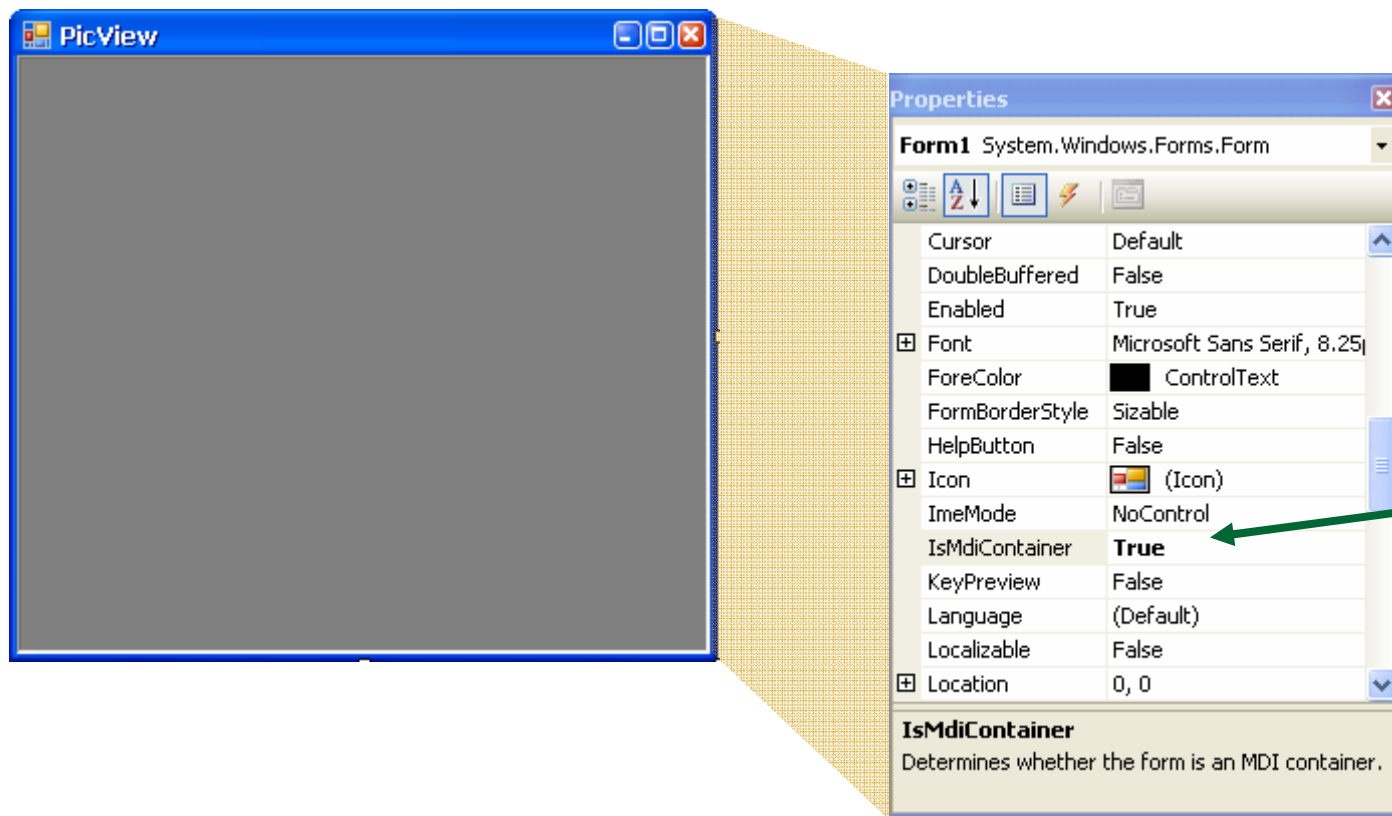


# Multiple Document Interface

- Tạo ứng dụng MDI: PixView xem ảnh
- Tạo ứng dụng MDI trong đó
  - Form cha
    - Chức năng open: mở file ảnh từ đĩa
    - Gọi form con hiển thị ảnh
    - Cho phép sắp xếp các cửa sổ con
  - Form con
    - Lấy thông tin file từ form cha
    - Hiển thị file ảnh lên PictureBox

# Multiple Document Interface

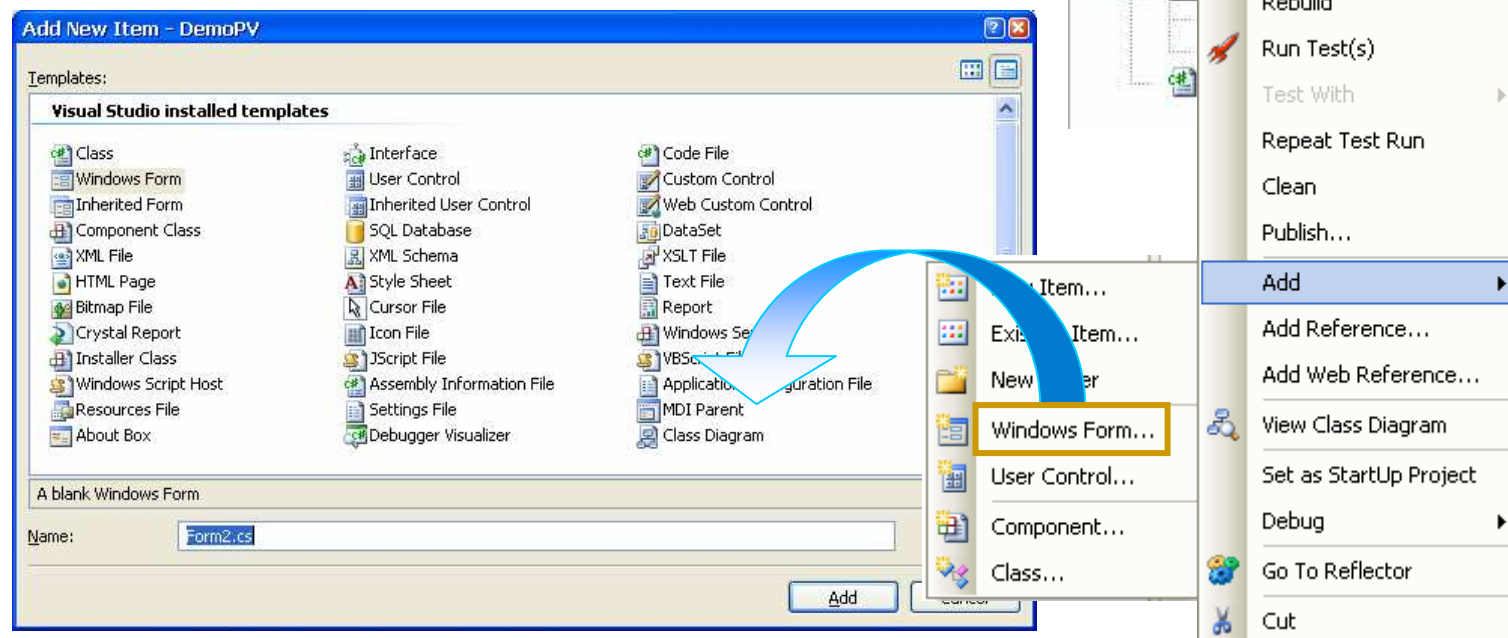
- Bước 1: Tạo ứng dụng Windows App
  - Thiết lập Form1 là form cha



MDI

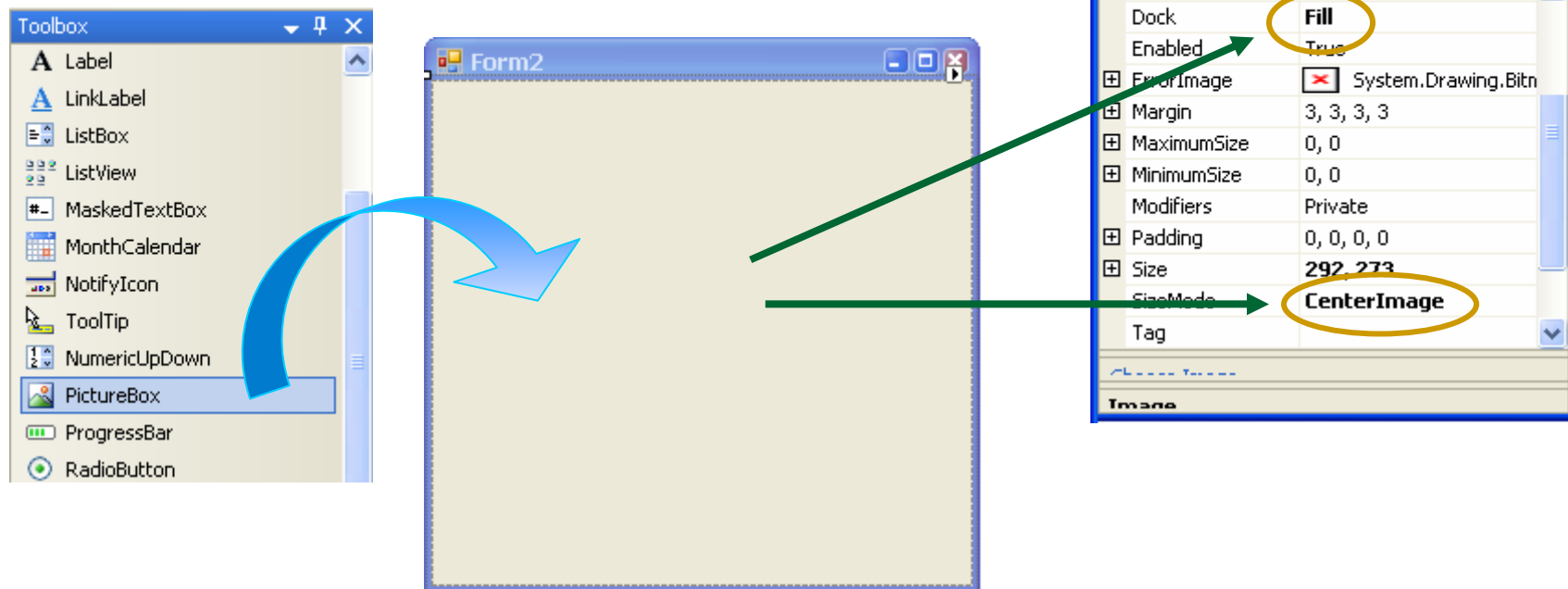
# Multiple Document Interface

- **Bước 2: tạo form con**
  - ❑ Trong cửa sổ Solution Explorer kích chuột phải lên tên Project
  - ❑ Chọn Add -> Windows Form
  - ❑ Đặt tên cho form con



# Multiple Document Interface

- Bước 3: bổ sung control PictureBox vào Form2
- Thiết lập các thuộc tính cho PictureBox
  - Dock = Fill
  - SizeMode = CenterImage



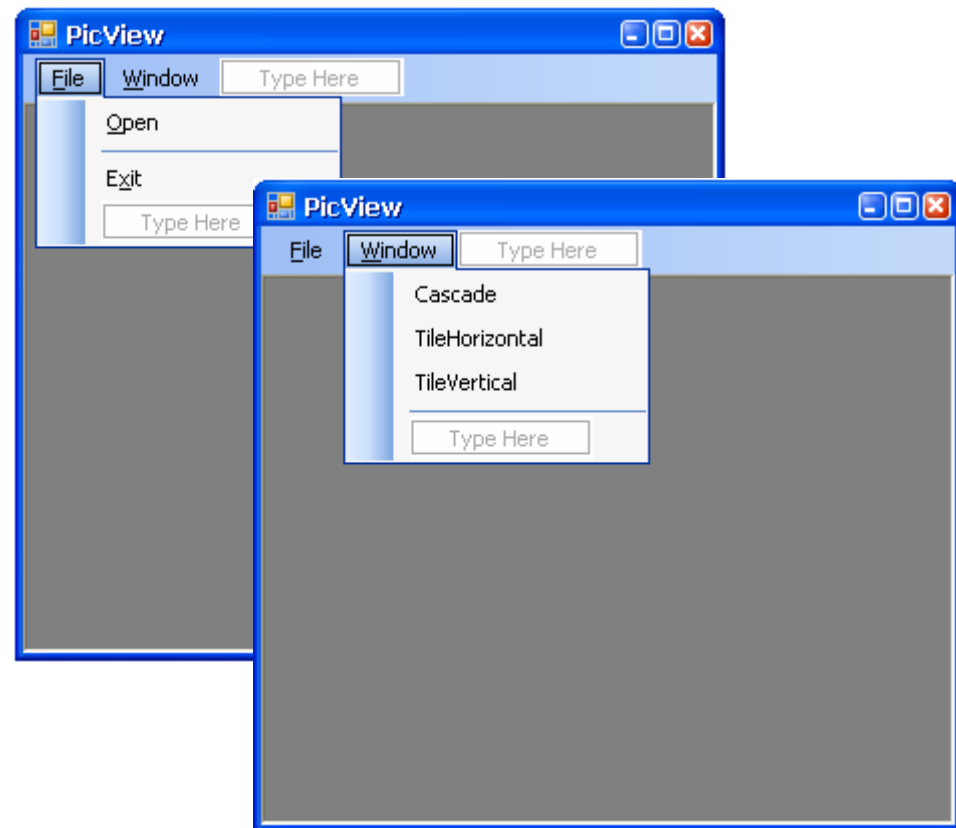
# Multiple Document Interface

- Bước 4: bổ sung constructor cho form2,
  - Constructor này có tham số là tên file cần mở

```
public partial class Form2 : Form
{
    public Form2(string imageFile)
    {
        InitializeComponent();
        //cho pictureBox hiển thị file ảnh
        pictureBox1.Image = Image.FromFile(imageFile);
        //hiển thị tên của file ảnh lên title bar
        Text = imageFile.Substring(imageFile.LastIndexOf('\\')+1);
    }
    public Form2()
    {
        InitializeComponent();
    }
}
```

# Multiple Document Interface

- Bước 5: tạo menu cho form chính (Form1)
- Kéo MenuStrip thả vào Form1
  - Tạo các top menu:
    - File
      - Open
      - Exit
    - Window
      - Cascade
      - TileHorizontal
      - TileVertical



# Multiple Document Interface

- **Bước 5: viết phần xử lý cho menu item “Open”**
  - Trong màn hình thiết kế kích đúp vào menu item Open
  - VS.NET sẽ tạo trình xử lý cho menu item
  - Phần xử lý được mô tả như sau
    - Mở dialog để chọn file
    - User chọn 1 file ảnh
    - Lấy đường dẫn file ảnh
    - Tạo mới đối tượng form2 truyền tham số đường dẫn file
    - Thiết lập liên kết giữa form cha và form con
    - Hiện thị form2 ra màn hình

# Multiple Document Interface

- Đoạn code minh họa xử lý cho menu item “Open”

```
private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    //sử dụng dialog chuẩn để open file
    OpenFileDialog oFile = new OpenFileDialog();
    // chỉ open những file *.bmp, *.jpeg
    oFile.Filter = "Bitmap file|*.bmp|JPEG file|*.jpg";
    if (oFile.ShowDialog() == DialogResult.OK) // hiển thị dialog
    {
        // tạo thể hiện form con và truyền tên file cần mở
        Form2 frm = new Form2(oFile.FileName);
        frm.MdiParent = this; // khai báo form cha
        frm.Show(); // hiển thị ra màn hình
    }
}
```

# Multiple Document Interface

- Bổ sung các chức năng sắp xếp form con
  - Kích đúp vào các menu item tương ứng trên menu Window
  - Viết các xử lý cho các menu item
    - Gọi hàm LayoutMdi với các tham số tương ứng

```
private void cascadeToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    LayoutMdi(MdiLayout.Cascade);
}

private void tileHorizontalToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    LayoutMdi(MdiLayout.TileHorizontal);
}

private void tileVerticalToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    LayoutMdi(MdiLayout.TileVertical);
}
```

# Multiple Document Interface

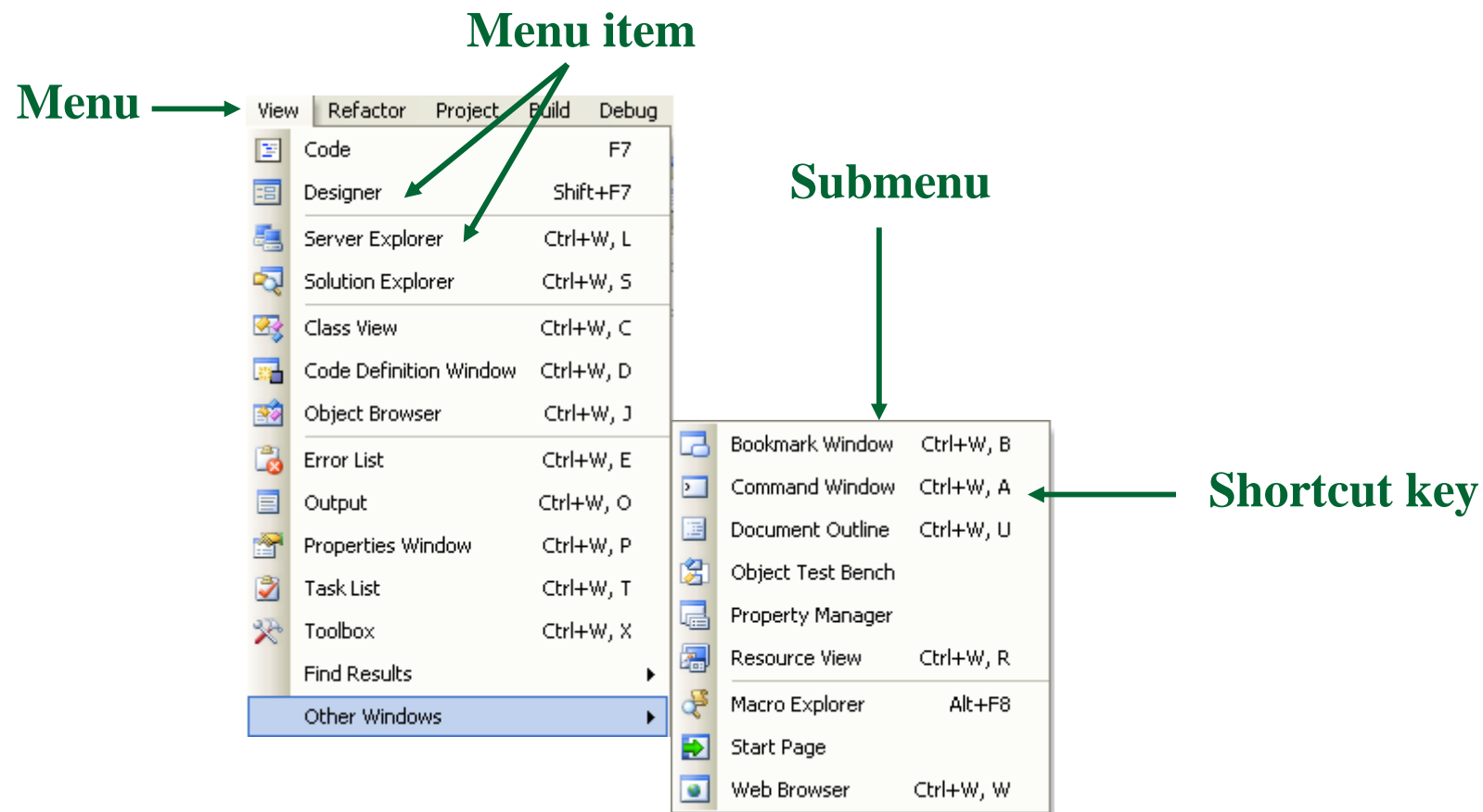




## *Menu - MenuStrip*

# Menu

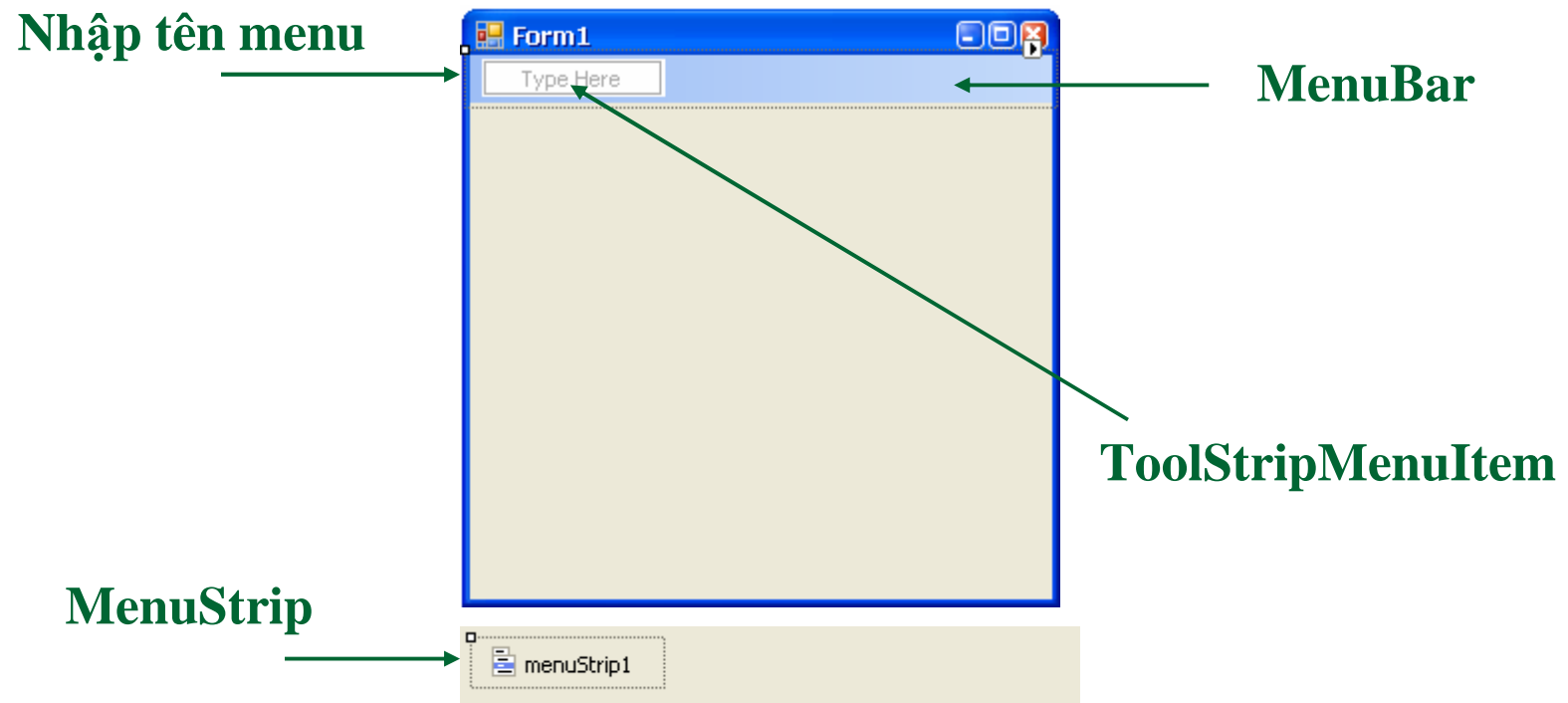
- Menu cung cấp nhóm lệnh có quan hệ với nhau cho các ứng dụng Windows



# Menu

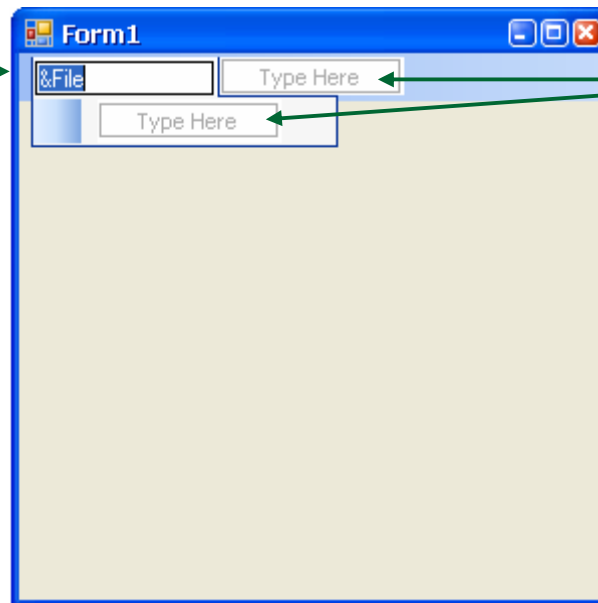
## ■ Cách tạo menu

- ❑ 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

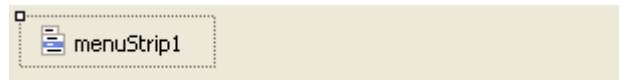


# Menu

Đặt ký tự &  
trước ký tự  
làm phím tắt

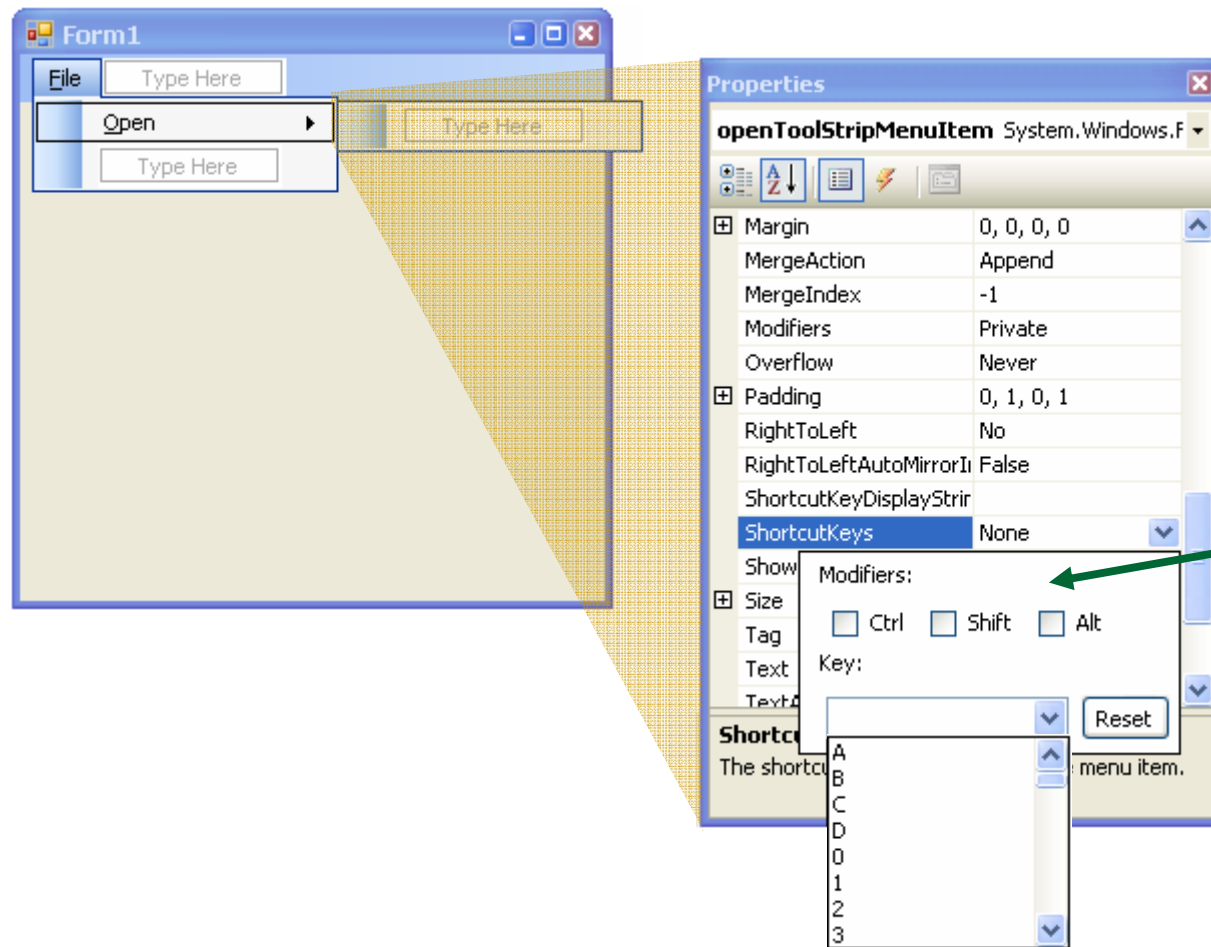


Tạo menu item  
mới bằng cách  
nhập vào textbox



# Menu

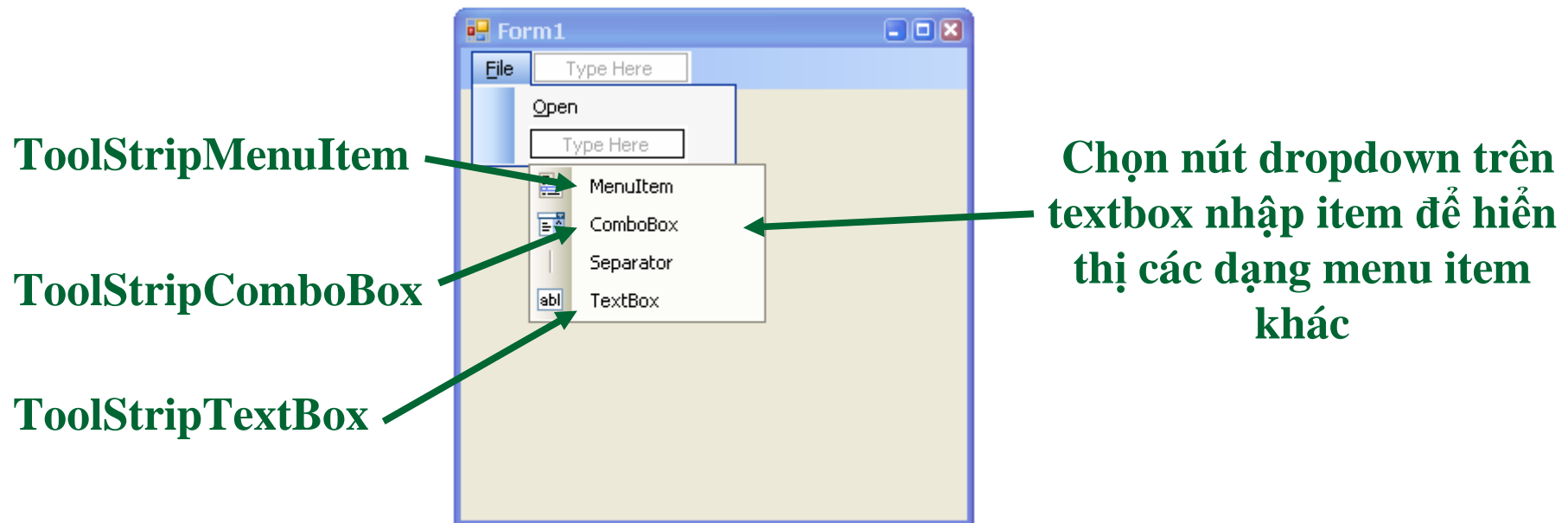
- Thiết lập Shortcut Key cho menu item



Khai báo  
shortcut key

# Menu

- Menu item có thể là TextBox hoặc ComboBox



# Menu

## *Thuộc tính của MenuStrip, ToolStripMenuItem*

### *MenuStrip*

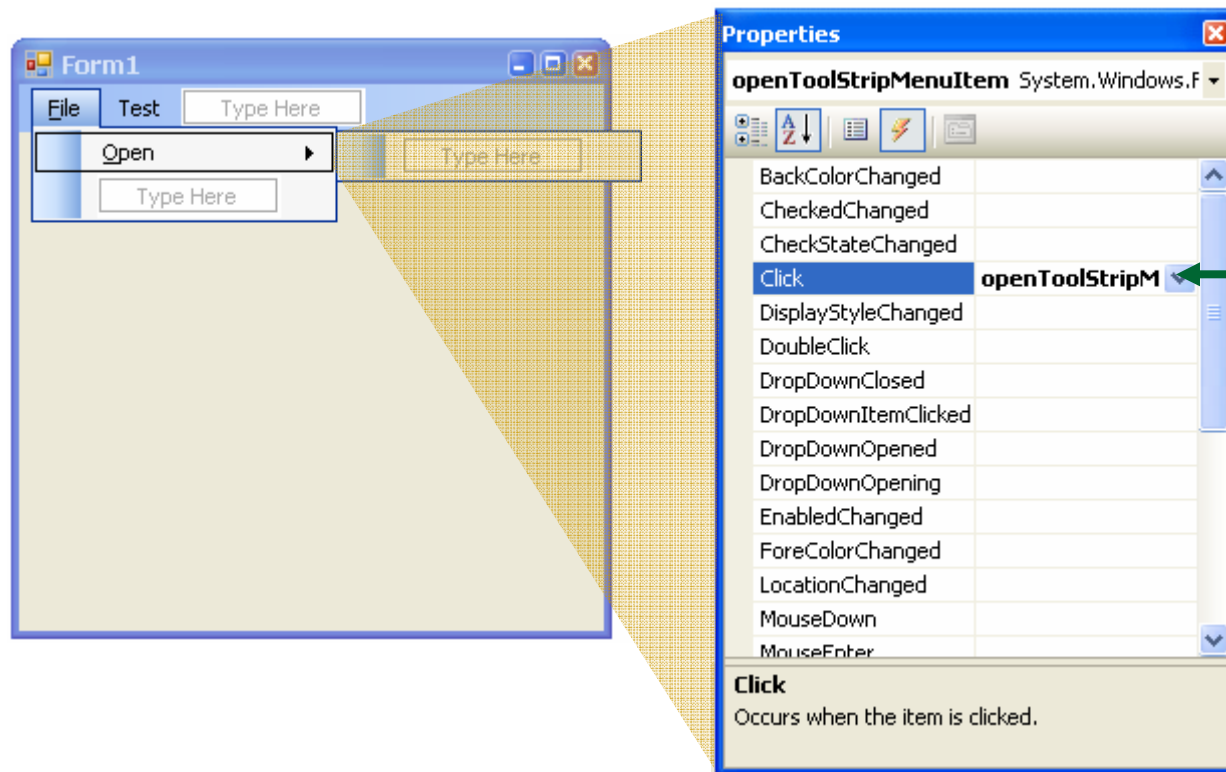
Items	Chứa những top menu item
MdiWindowListItem	Chọn top menu item hiển thị tên các cửa sổ con

### *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 hiển thị phím tắt bên cạnh menu item

# Menu

- 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

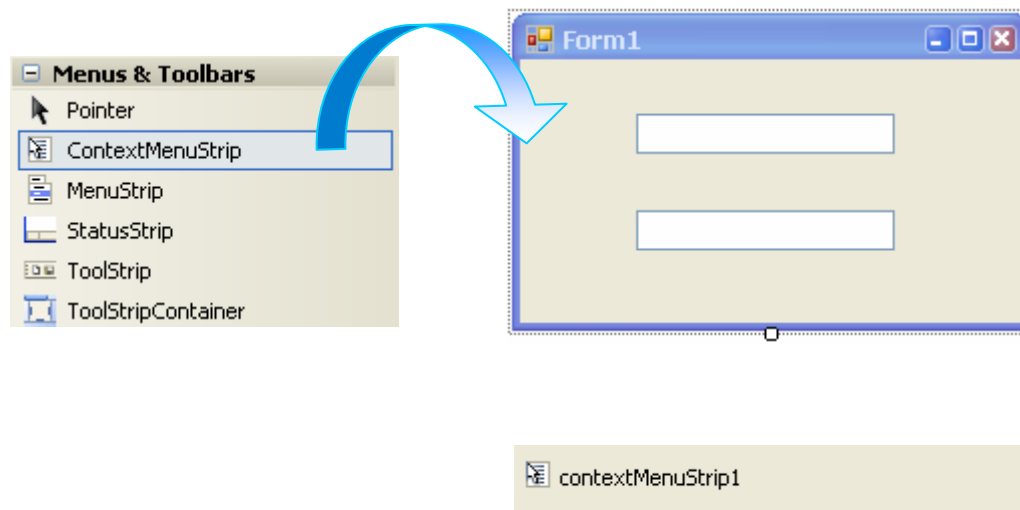


Trình xử lý cho  
sự kiện click của  
menu item  
“Open”

# *ContextMenu*

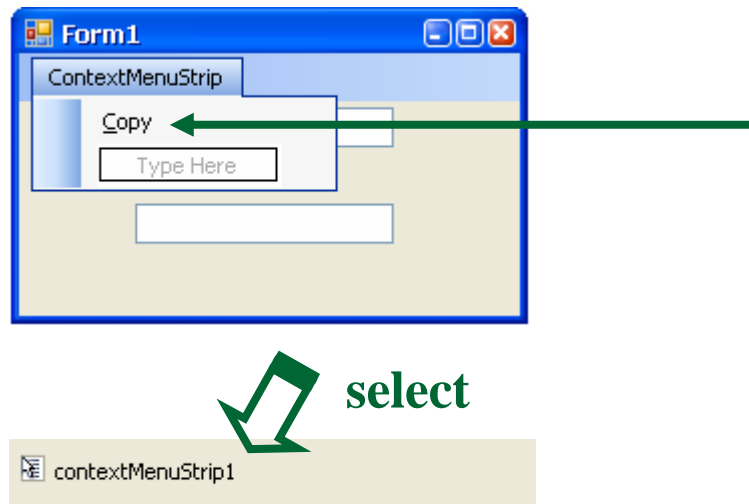
# Context Menu

- 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



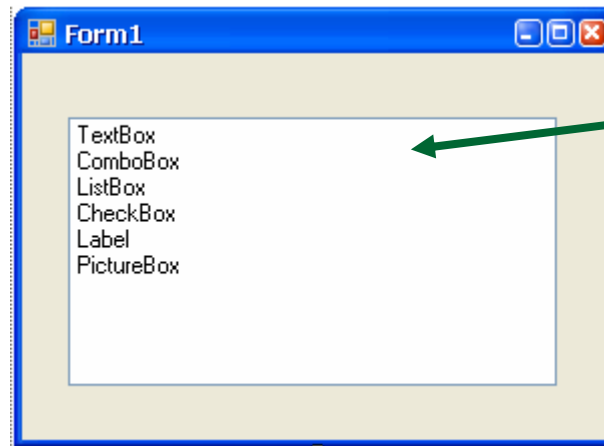
Soạn thảo Context  
Menu tương tự như  
Menu bình thường

# 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

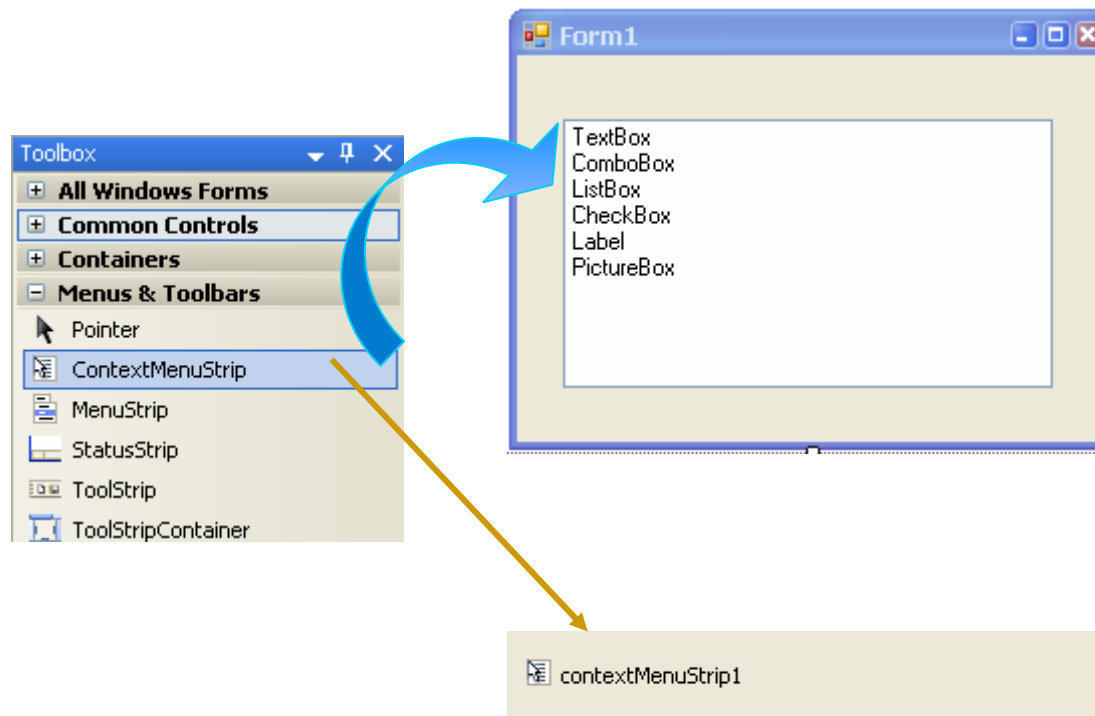
- Demo: tạo context Menu hiển thị trong ListBox có menu item Remove, cho phép xóa item đang được chọn.
- Tạo Form có mô tả như sau



ListBox hiển thị  
các item

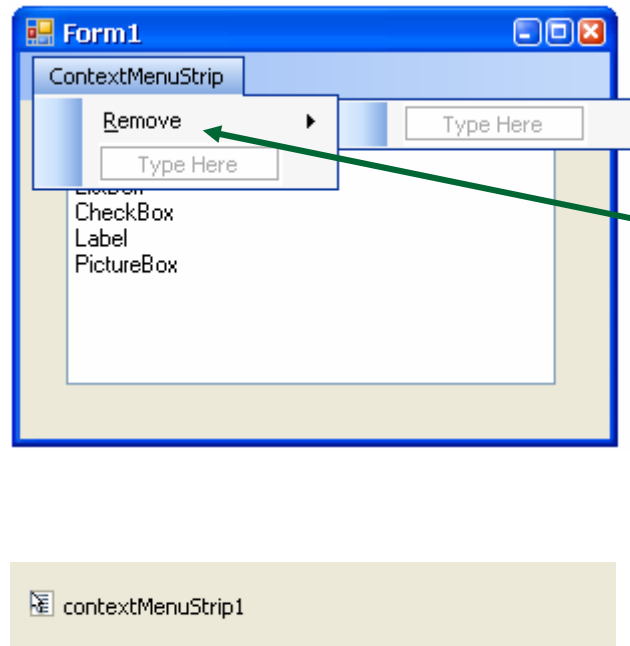
# Context Menu

- Kéo ContextMenuStrip thả vào Form



# Context Menu

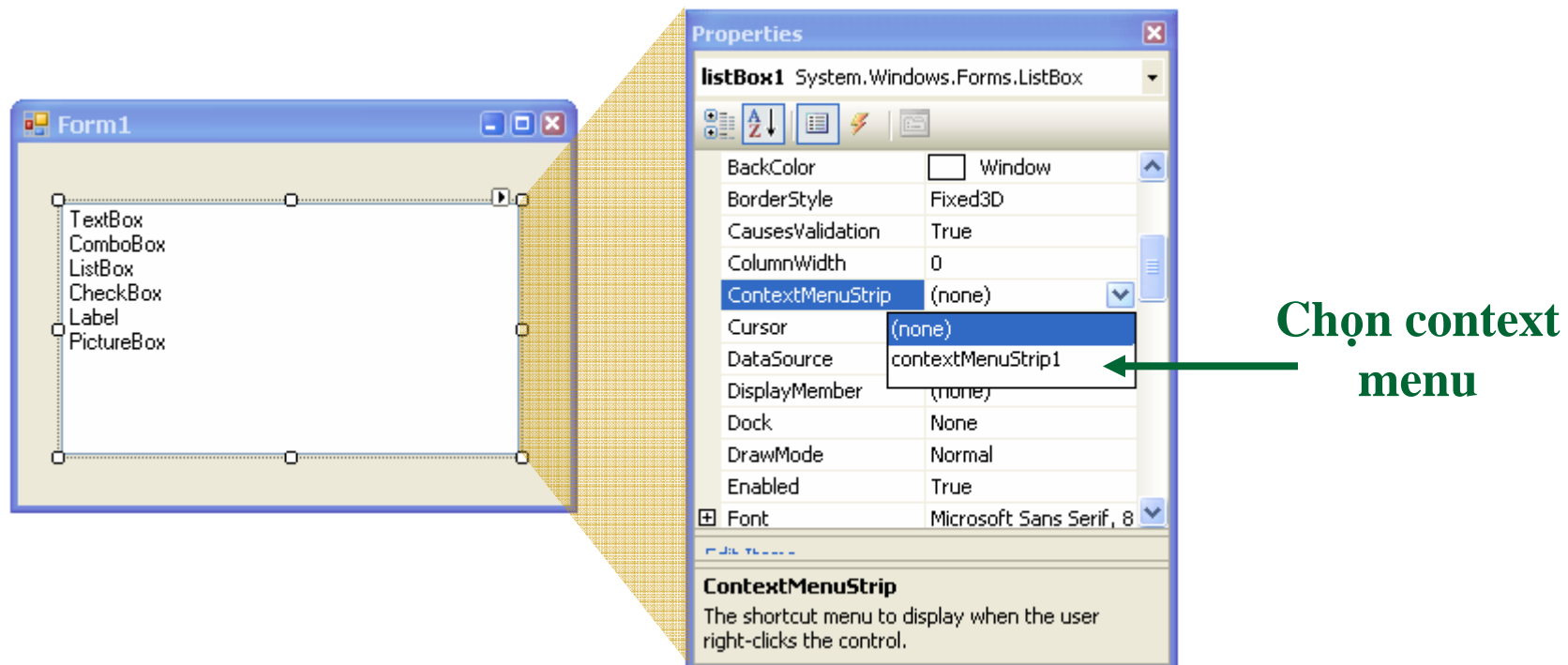
- Kích vào ContextMenuStrip để thiết kế menu
  - Tạo một menu item “Remove” như hình mô tả



**Soạn thảo các  
menu item**

# Context Menu

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



# Context Menu

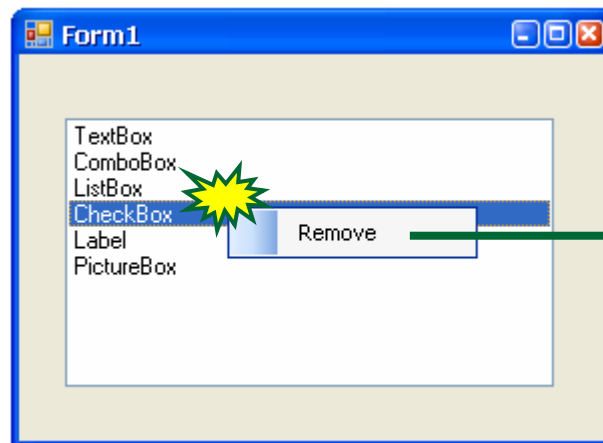
- Khai báo trình xử lý sự kiện Click cho menu item

The image shows a Visual Studio IDE with a Windows Form named 'Form1'. A context menu is displayed over the form, containing a single item '&Remove'. The 'Properties' window is open, showing the 'Click' event for the 'mnuRemove' menu item, which is set to 'mnuRemove\_Click'. A blue arrow points from this event in the Properties window to a code block containing the implementation of the 'mnuRemove\_Click' method.

```
private void mnuRemove_Click(object sender, EventArgs e)
{
    // xác định item được chọn trong listbox
    if (listBox1.SelectedIndex >= 0)
    {
        listBox1.Items.Remove(listBox1.SelectedItem);
    }
}
```

# Context Menu

- Demo



**Context Menu**  
hiển thị khi user  
kích chuột phải  
lên ListBox



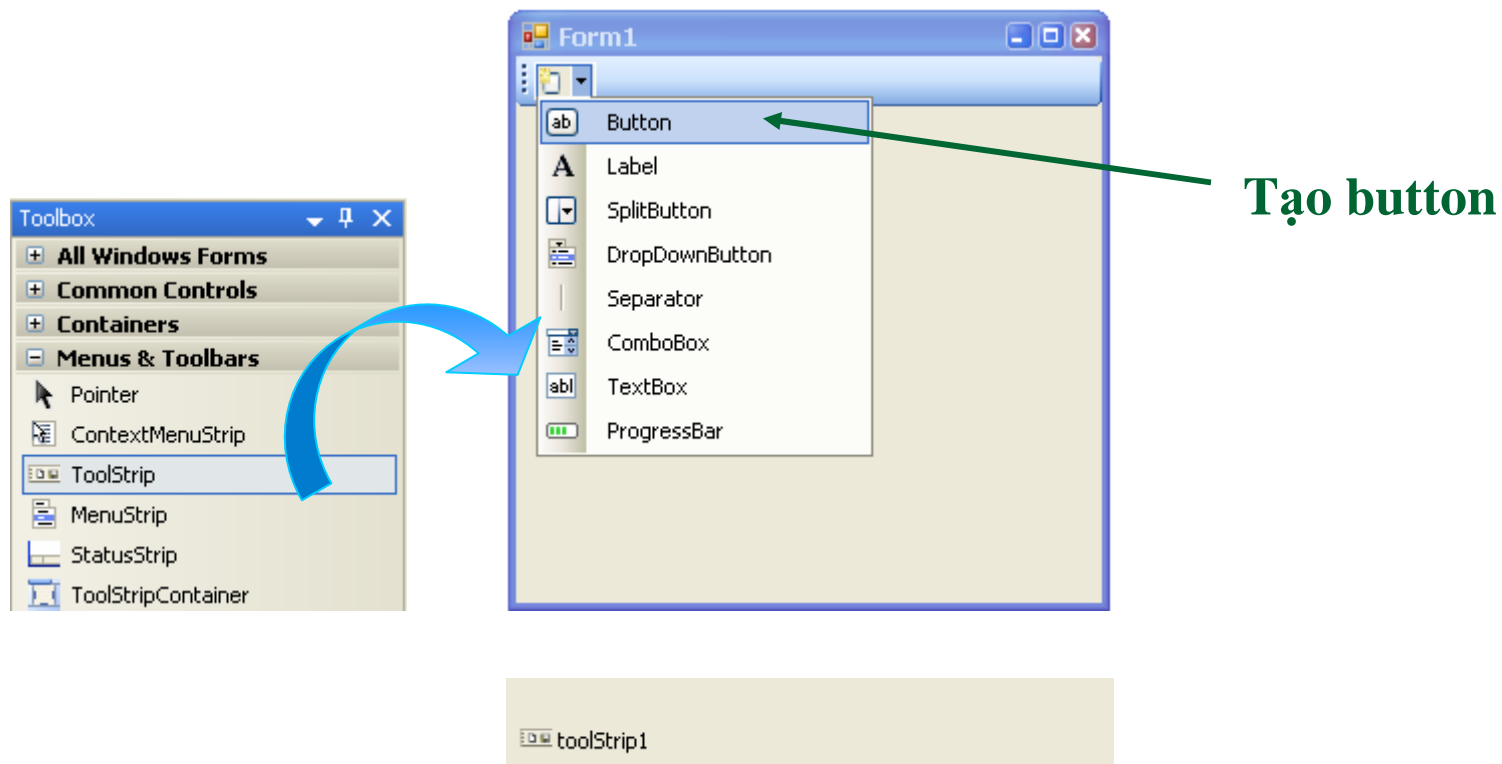
# *ToolStrip*

# ToolStrip

- ToolStrip là sự thay thế cho ToolBar trong các ứng dụng trước đây
- Vị trí thường xuất hiện là ngay bên dưới thanh menu
- Cung cấp các button cho phép thực hiện các chức năng thường dùng trong menu
- ToolStrip là dạng container cho phép chứa các control
  - Các control này dẫn xuất từ ToolStripItem
  - Các control bao gồm
    - ToolStripSplitButton
    - ToolStripDropDownbutton
    - ToolStripLabel
    - ToolStripProgressBar
    - ToolStripSeparator
    - ToolStripComboBox
    - ToolStripTextBox

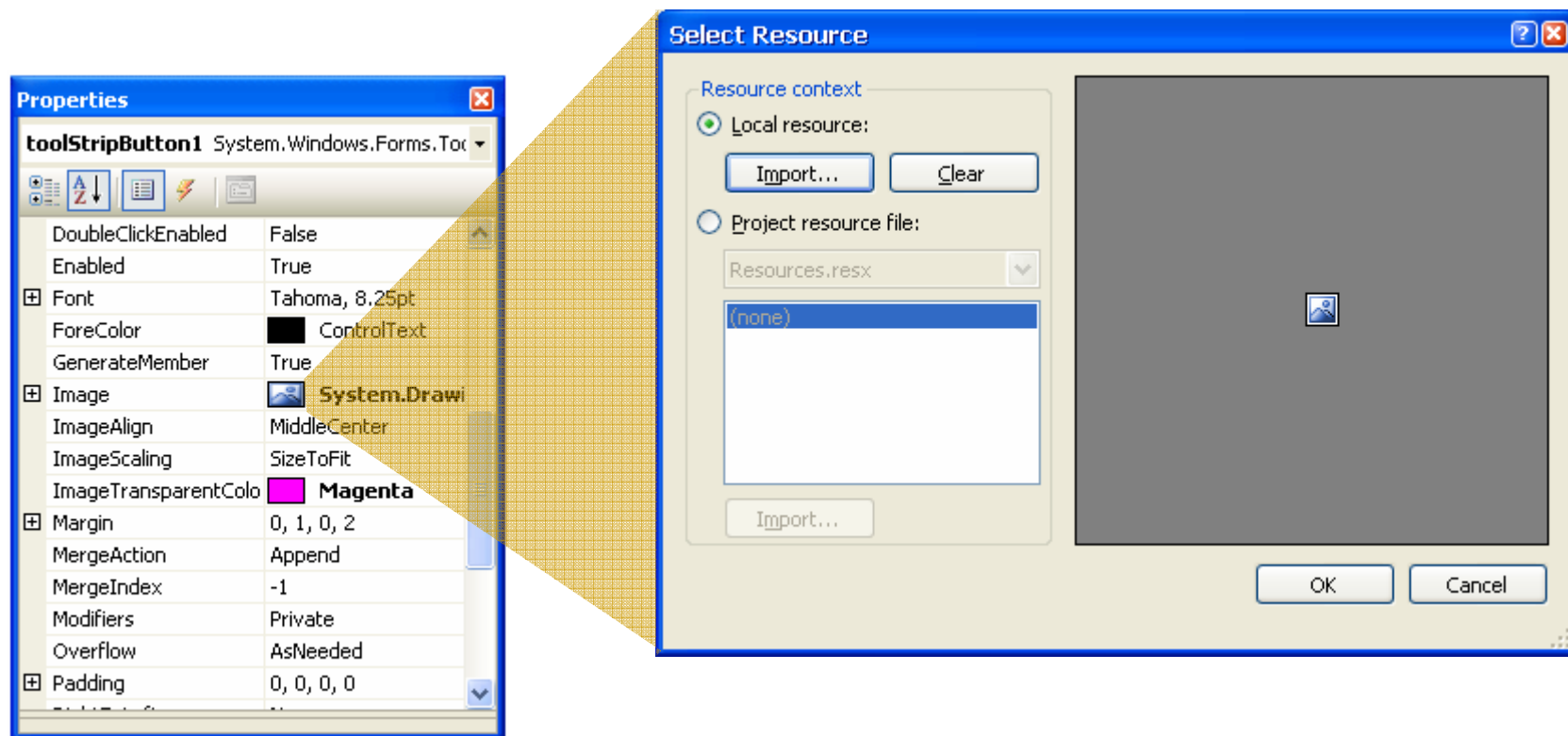
# ToolStrip

- Cách tạo button trên ToolStrip
  - ❑ Kéo ToolStrip thả vào form
  - ❑ Add ToolStripButton

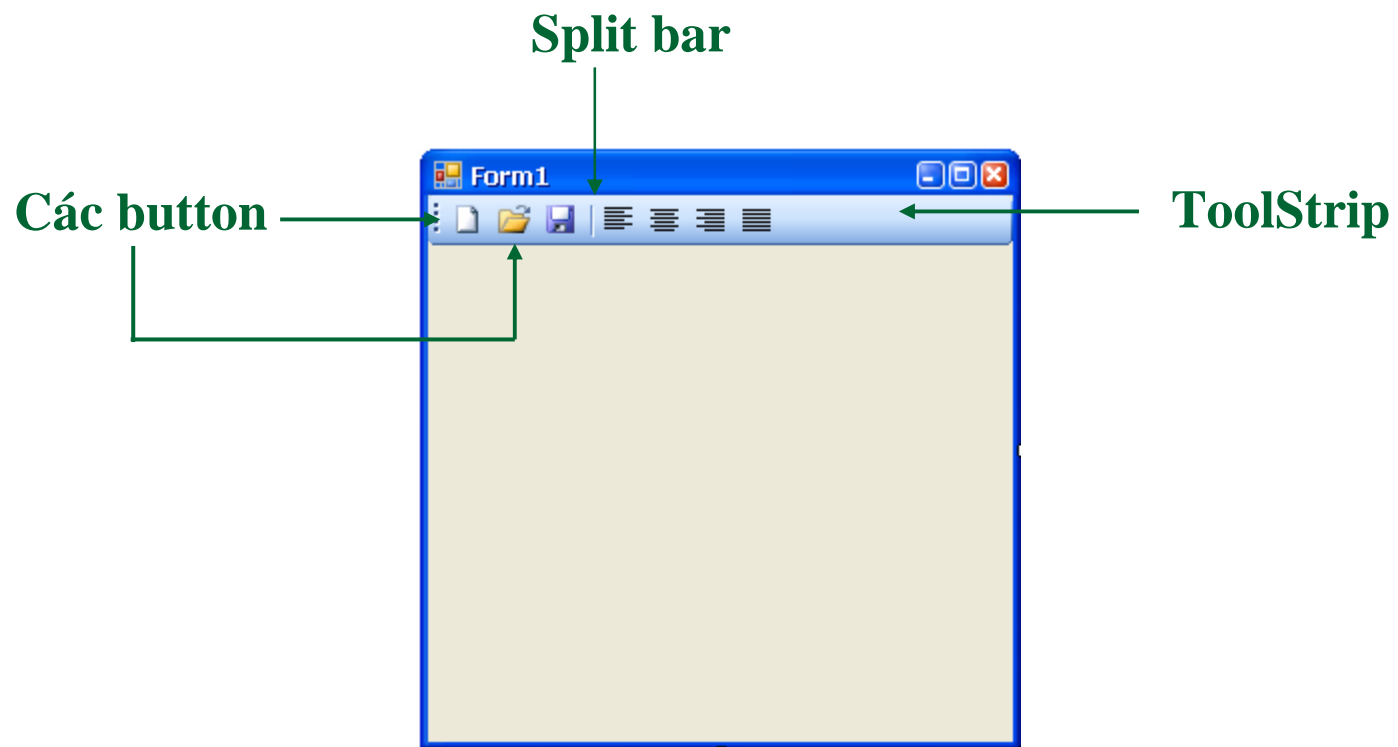


# ToolStrip

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



# ToolStrip

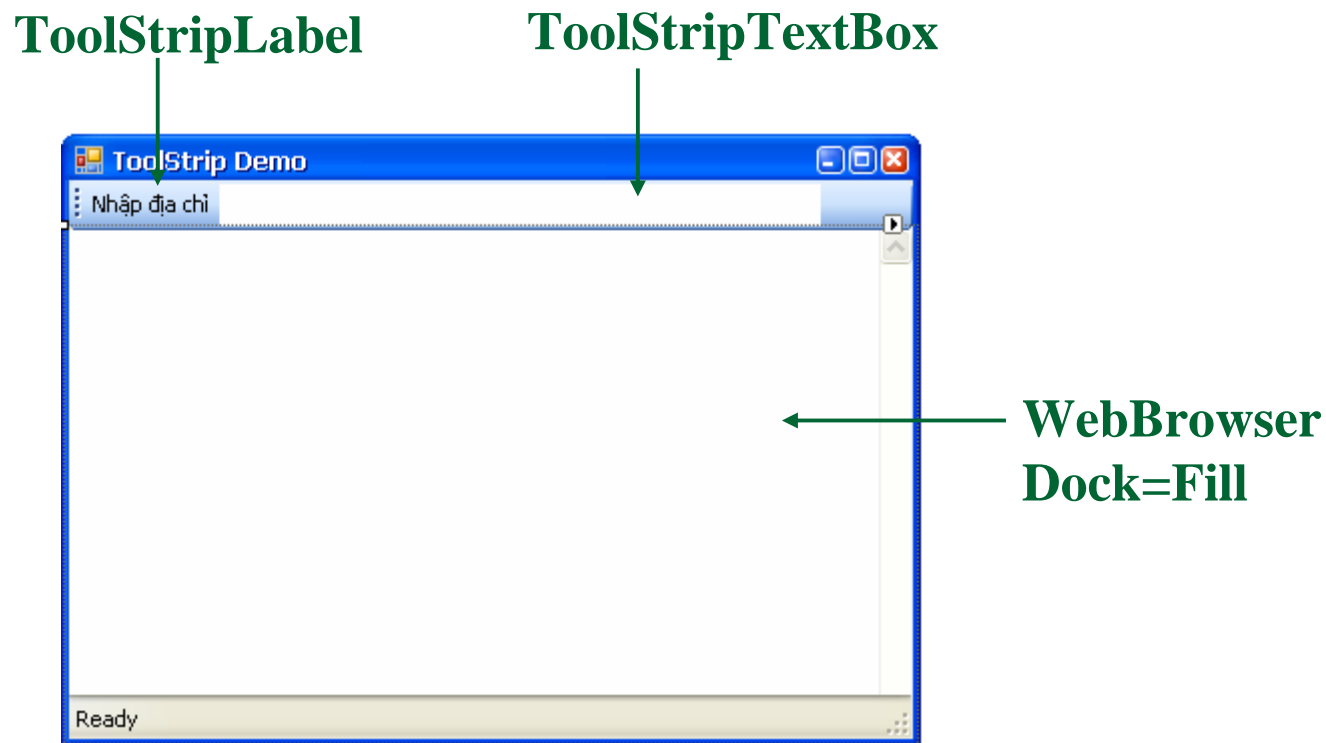


# ToolStrip

- 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”

# ToolStrip

- Demo: tạo thanh menubar cho phép nhập URL và khi nhấn enter chương trình sẽ open địa chỉ đó



# ToolStrip

- Xử lý sự kiện KeyDown cho ToolStripTextBox

The image shows a screenshot of a Visual Studio IDE. On the left, a window titled "ToolStrip Demo" contains a ToolStrip control with a single item, a text box labeled "Nhập địa chỉ". Below the text box is a status bar with the text "Ready". On the right, the "Properties" window is open, showing the selected control as "toolStripTextBox1" of type "System.Windows.Forms.ToolStripTextBox". The "Events" tab is active, and the "KeyDown" event is selected, with the handler "toolStripTextBox1\_KeyDown" assigned. A blue arrow points from the "KeyDown" event in the Properties window to a code block. The code block contains the following C# code:

```
private void toolStripTextBox1_KeyDown(object sender,
    KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter)
        webBrowser1.Navigate(toolStripTextBox1.Text);
}
```

Below the code block, a tooltip for the "KeyDown" event is displayed, stating: "Occurs when a key is first pressed."

# ToolStrip

## ■ Demo

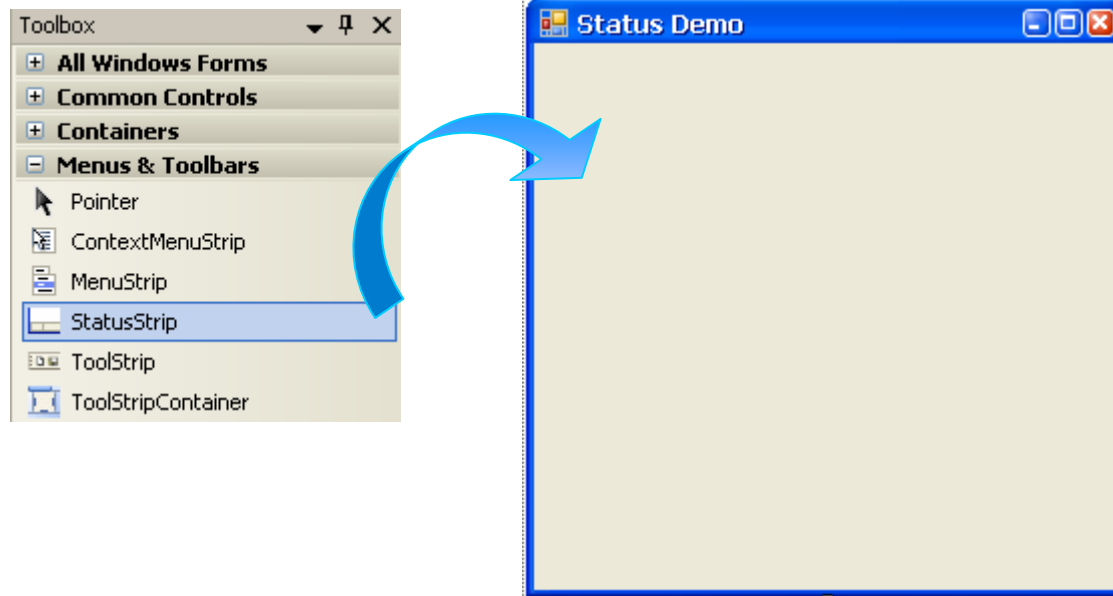




## *StatusStrip*

# StatusStrip

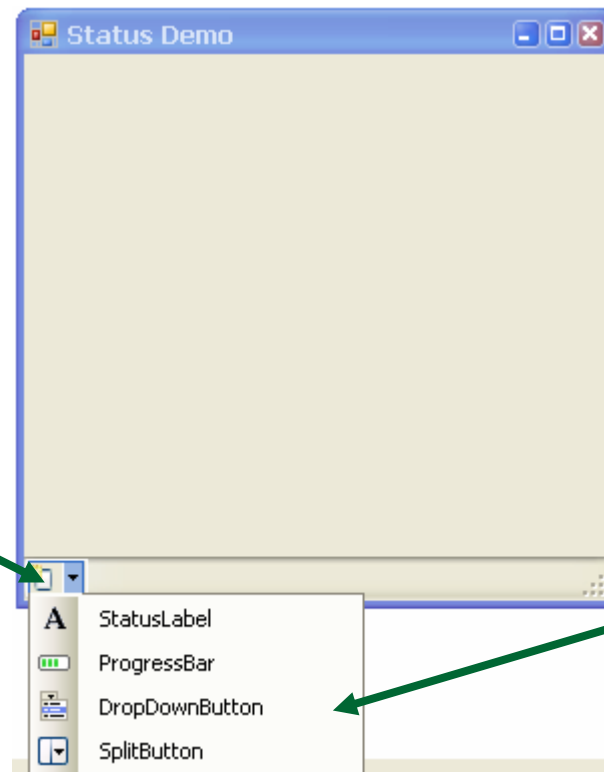
- **Hiển thị thông tin trạng thái của ứng dụng**
- **Nằm bên dưới cùng của Form.**
- **Các lớp liên quan**
  - **StatusStrip**: là container chứa control khác
  - **ToolStripStatusLabel**: control có thể add vào **StatusStrip**



# 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
    - Thiết lập sự kiện Tick cho Timer với Interval là 1000 ms

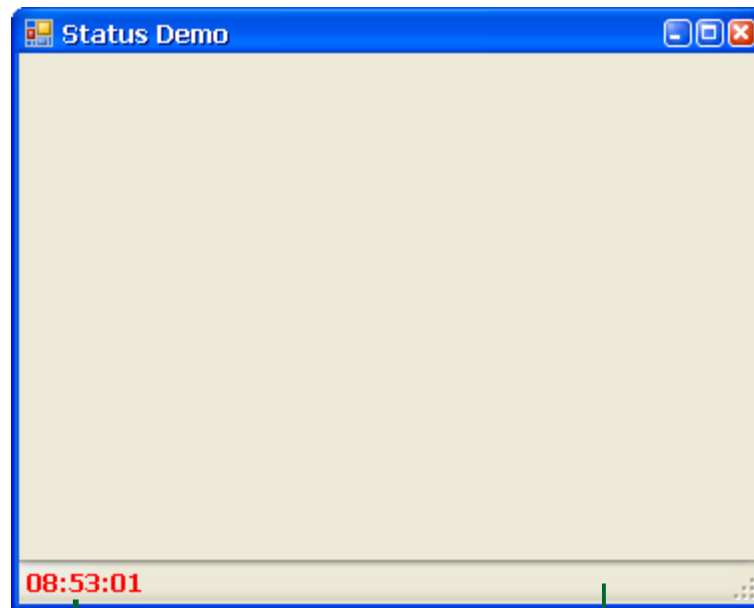
```
private void timer1_Tick(object sender, EventArgs e)
{
    toolStripStatusLabel1.Text =
        DateTime.Now.ToString("hh:mm:ss");
}
```



Item dạng Label của StatusStrip

# StatusStrip

- Demo

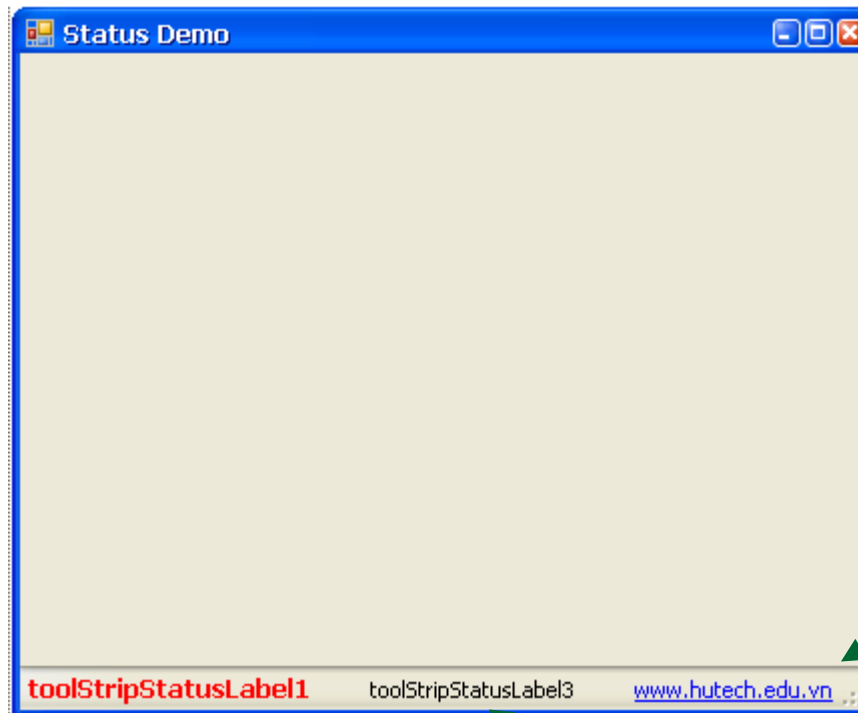


ToolStripStatusLabel

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 = “”

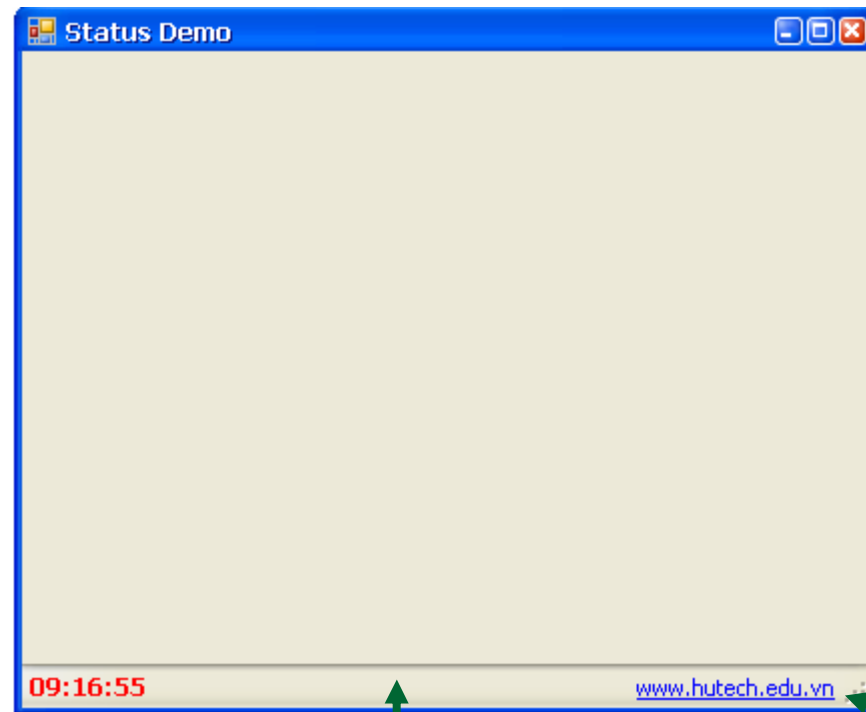
# StatusStrip

- 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 toolStripStatusLabel4_Click(object sender,
    EventArgs e)
{
    // sử dụng lớp Process để mở IE
    ProcessStartInfo startInfo = new ProcessStartInfo();
    startInfo.FileName = "iexplore.exe";
    startInfo.Arguments = "http://www.hutech.edu.vn/";
    Process.Start(startInfo);
}
```

# StatusStrip

## ■ Demo



Label thứ 2 fill đầy khoảng trống

Kích vào để mở web

# Q&A