

# Bài 7: MDI (Multiple Document Interfaces)

Lương Trần Hy Hiến  
FIT, HCMUP

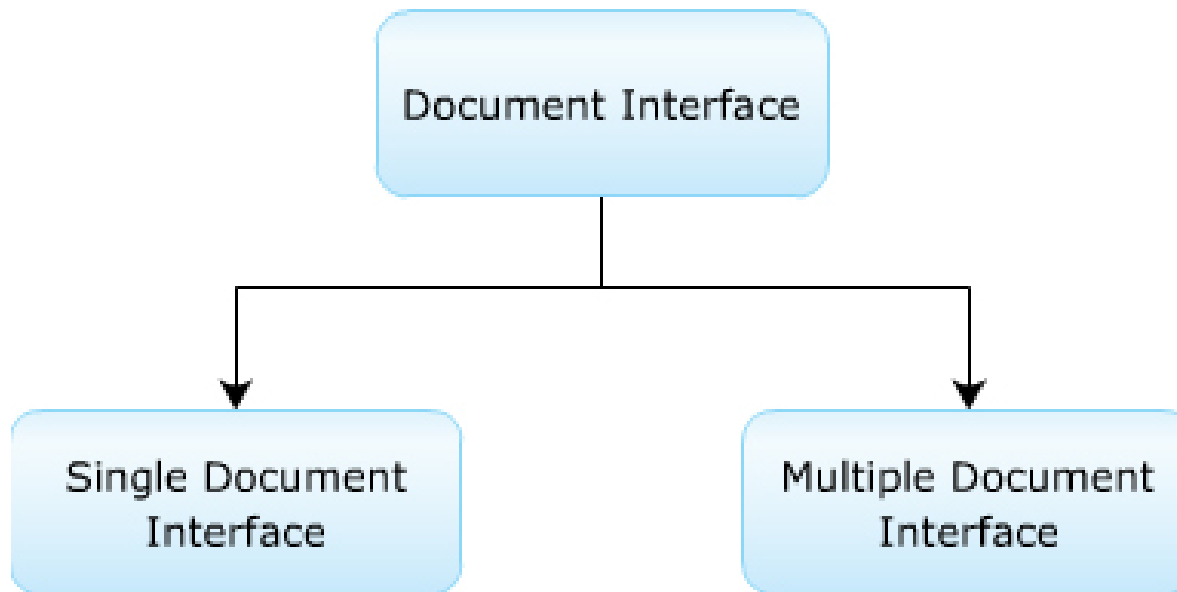


*Lập trình Windows Form với C#*

# Windows Form

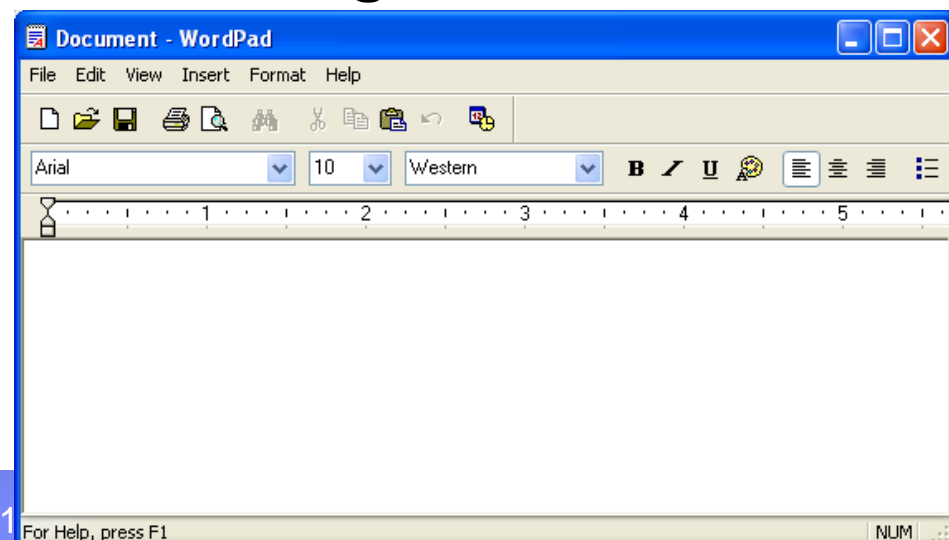
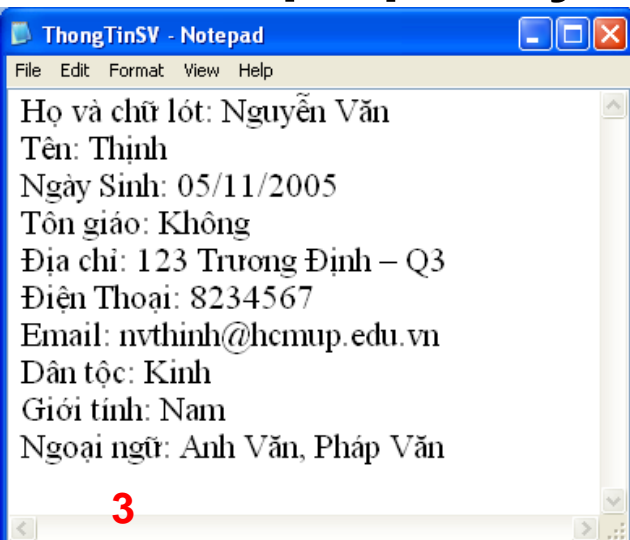
Có 2 loại tài liệu cơ bản:

- Single Document Interface (SDI)
- Multiple Document Interface (MDI)



# Single Document Interfaces

- 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.



## Khái niệm cơ bản về MDI

- **Multiple Document Interface (MDI) là một đặc tả quản lý tài liệu trong Microsoft Windows.**
- **Mô tả một cấu trúc cửa sổ và giao diện người dùng cho phép người sử dụng làm việc với nhiều tài liệu trong một ứng dụng đơn.**

## Khái niệm cơ bản về MDI

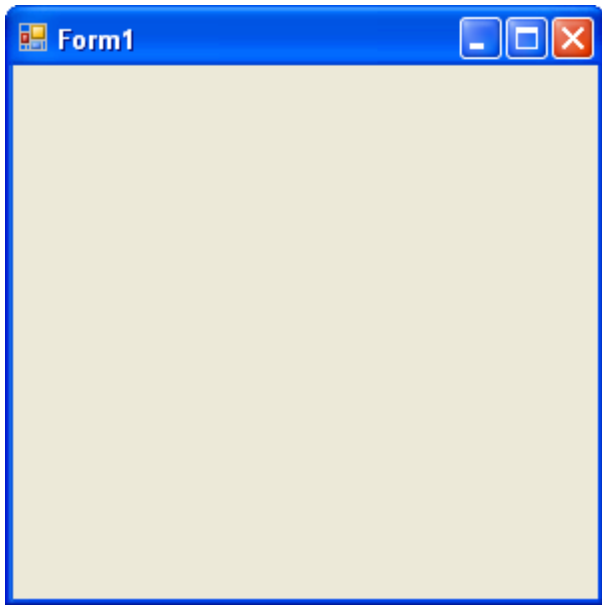
- Đặc tả MDI đã xuất hiện từ Windows 2.0, nhưng các ứng dụng MDI lúc đó rất khó viết và cần nhiều công sức lập trình.
- Từ Windows 3.0 trở đi, nhiều chức năng đã được mở rộng và hỗ trợ được đưa vào.

## Các thành phần MDI

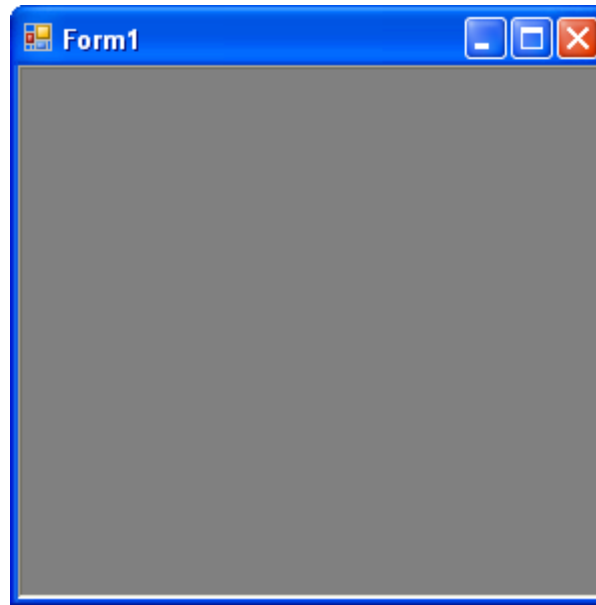
- **Cửa sổ ứng dụng chính của một chương trình MDI theo kiểu cổ điển gồm có:**
  - Thanh tiêu đề.
  - Một trình đơn.
  - Một đường viền thay đổi kích thước.
  - Một icon trình đơn hệ thống.
  - Các nút minimize/maximize/close.

- **Vùng client thường được gọi là “workspace” và không sử dụng trực tiếp để hiển thị output của chương trình.**
- **Workspace có thể không chứa hay chứa nhiều cửa sổ con, mỗi cửa sổ con hiển thị một tài liệu.**
- **Các cửa sổ con rất giống các cửa sổ ứng dụng bình thường và các cửa sổ ứng dụng chính của một chương trình MDI**
- **Ở tại một thời điểm, chỉ một cửa sổ tài liệu được kích hoạt và nó xuất hiện trước tất cả các cửa sổ tài liệu khác.**
- **Tất cả các cửa sổ tài liệu con được giới hạn bởi vùng workspace và không bao giờ xuất hiện bên ngoài cửa sổ ứng dụng.**

# SDI vs MDI



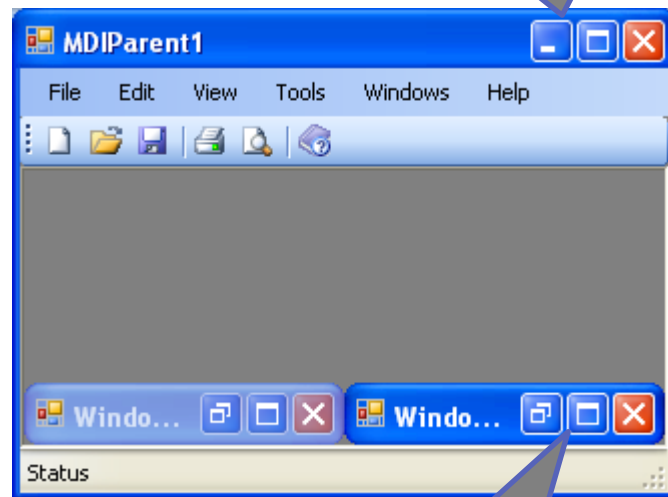
**SDI**



**MDI**

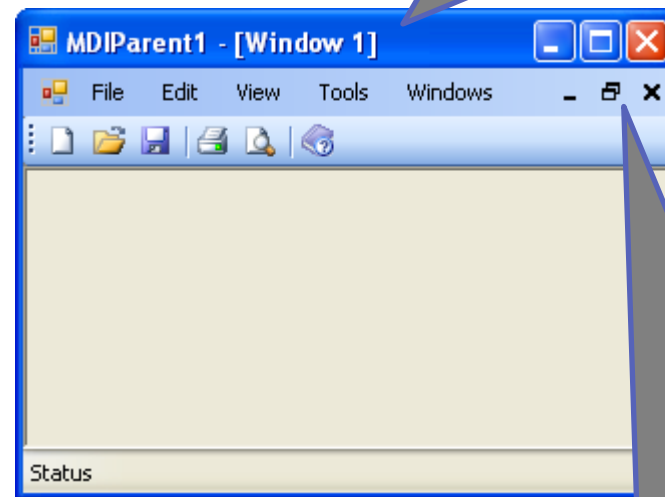


# MDI



Parent's icons: minimize, maximize and close

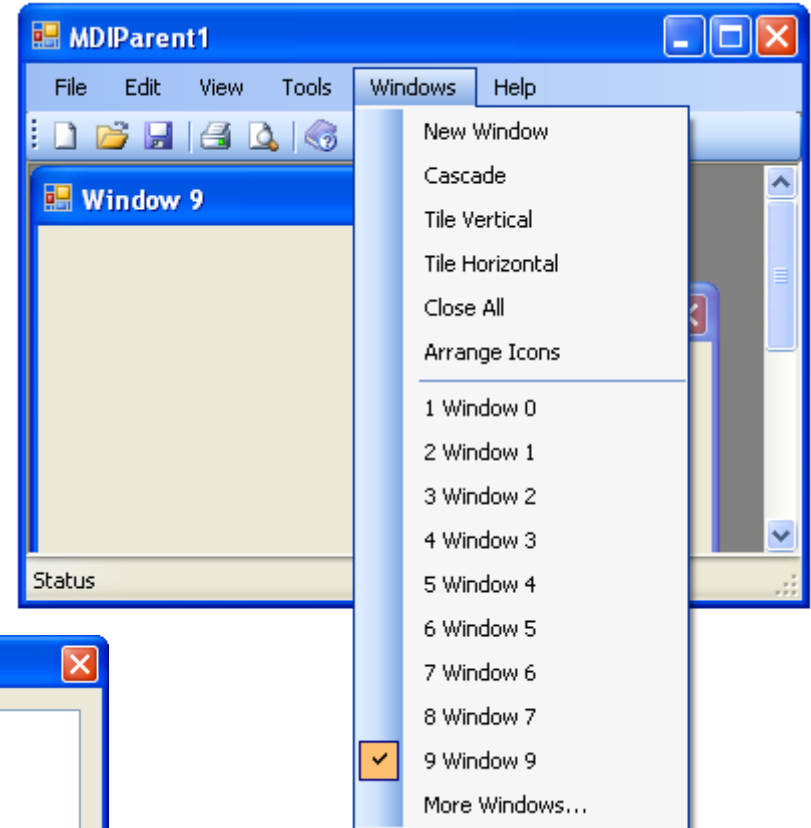
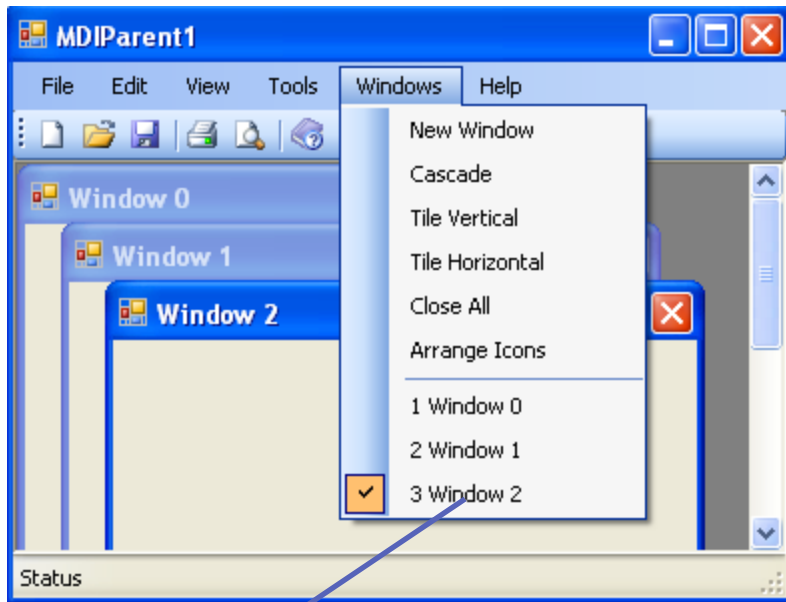
Parent's title bar displays maximized child



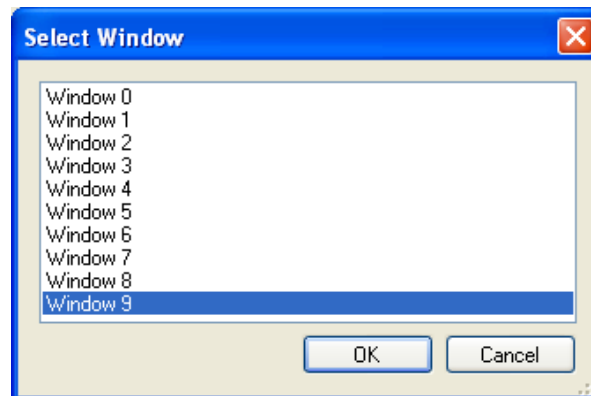
Maximized child's icons: restore, maximize and close

Minimized child's icons: restore, maximize and close

# MDI



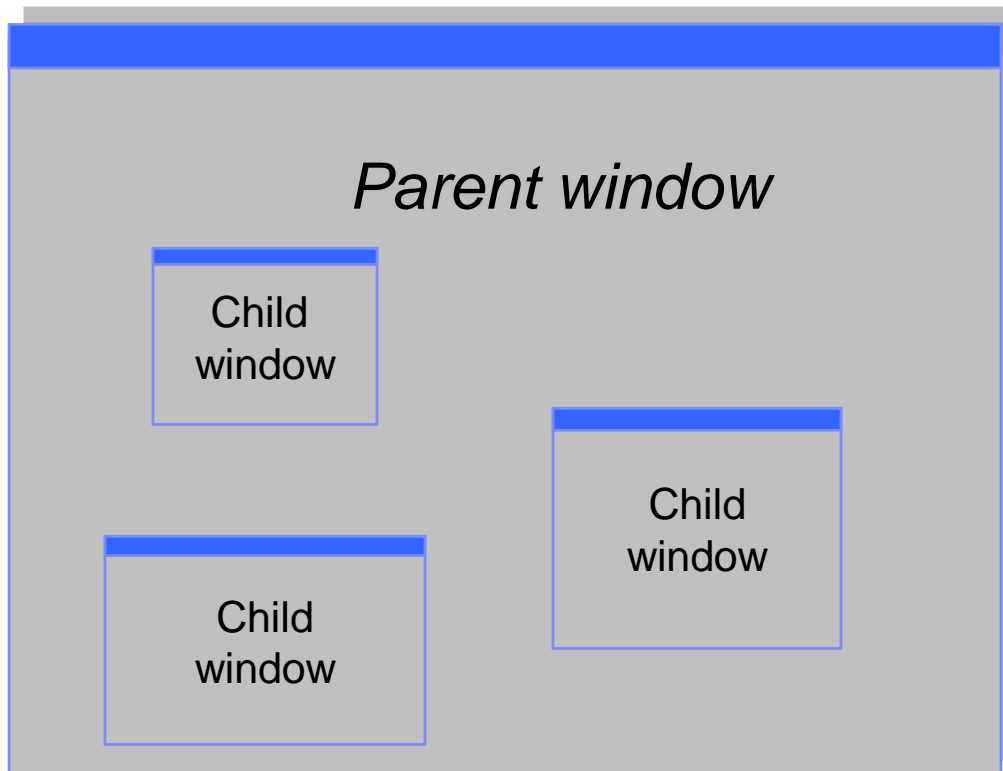
Child windows list



9 or more child windows  
enables the **More  
Windows...** option

# Multiple Document Interface

- **MDI Parent Window – MDI Child Window**  
MDI Application



Chỉ có duy nhất  
một parent  
windows

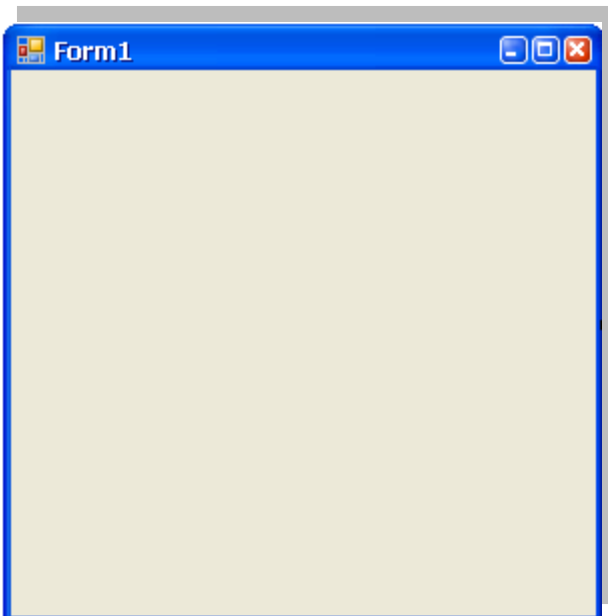
Có nhiều child  
windows

# Multiple Document Interface

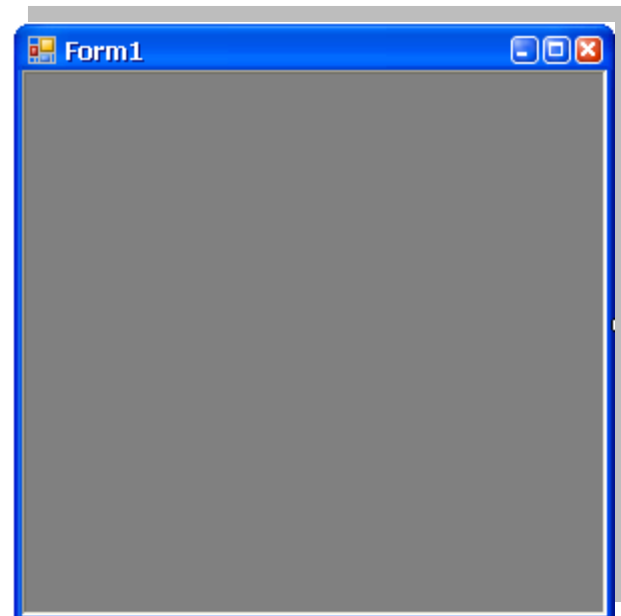
## ■ Cách tạo MDI Form

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

SDI Form

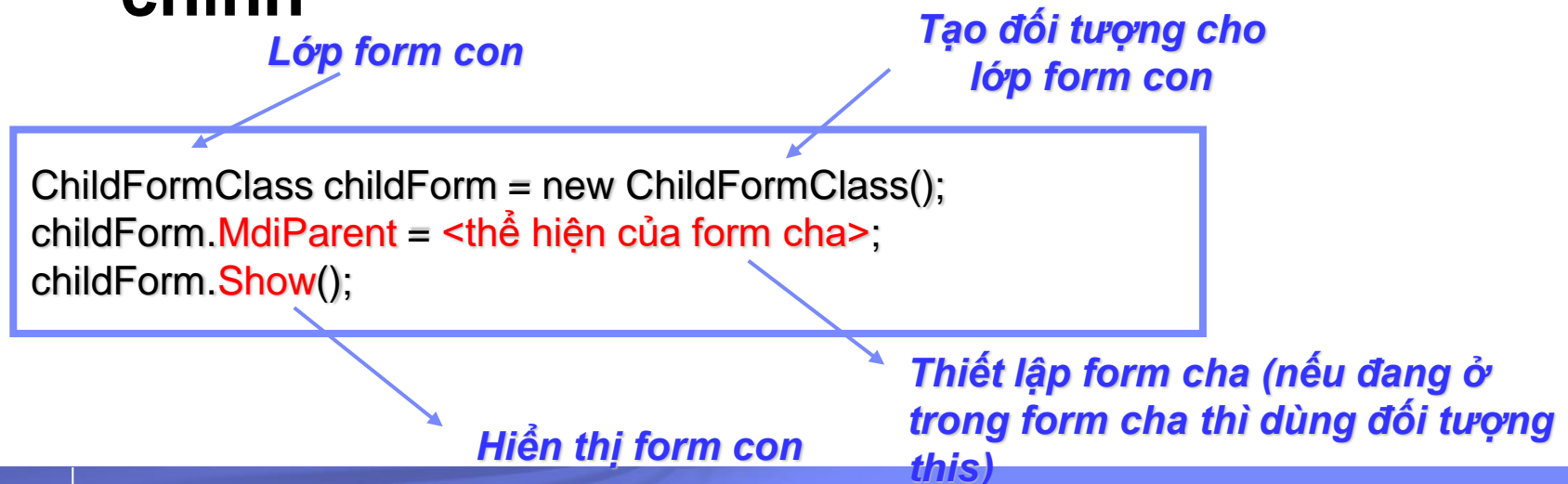


MDI Form



# Multiple Document Interface

- **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*

<b>IsMdiChild</b>	<b>Cho biết form là child hay không</b>
<b>MdiParent</b>	<b>Xác định form cha của form</b>

## *Thuộc tính của MDI parent*

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

## *Phương thức*

<b>LayoutMdi</b>	<b>Xác định cách hiển thị các form con</b>
------------------	--

## *Sự kiện*

<b>MdiChildActivate</b>	<b>Xuất hiện khi child form đóng hoặc active</b>
-------------------------	--

# Multiple Document Interface

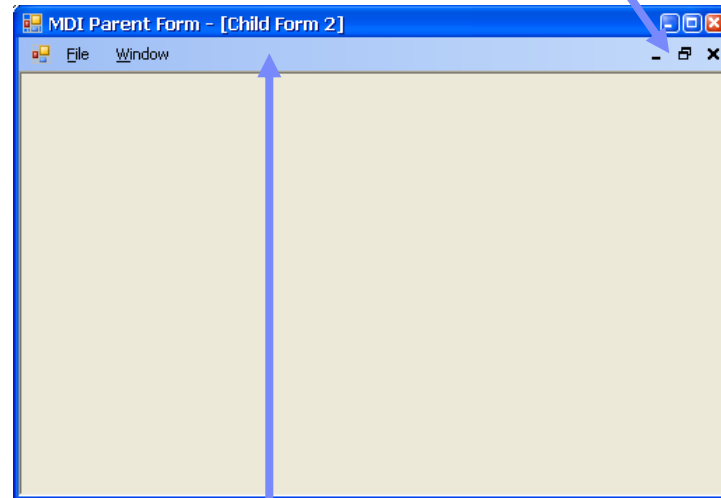
## ■ Trạng thái Minimize, Maximize và Close

Icon của **parent**: minimize, maximize, close



Icon của **child**: restore, maximize, close

Icon của **child**: minimize, restore, close



Title của parent chứa tên child đang maximize

## Sắp xếp các MDI

- Để tạo một danh sách các MDI Child, chỉ cần thêm một top-level menu item (thường mang tên Window) rồi cho thuộc tính **MdiList = true**.
- Bộ máy Windows.Forms sẽ tự động thêm một item vào cuối submenu cho mỗi cửa sổ MDI Child.
- Mỗi MDI container đều hỗ trợ hàm **LayoutMdi()** hàm này nhận giá trị từ Enumeration **MdiLayout** và sắp xếp tự động các cửa sổ.



- Đoạn chương trình sắp xếp theo kiểu Cascade

```
private void mnuCascade_Click ( Object sender, System.EventArgs e)  
{  
    this.LayoutMdi (MdiLayout.Cascade);  
}
```

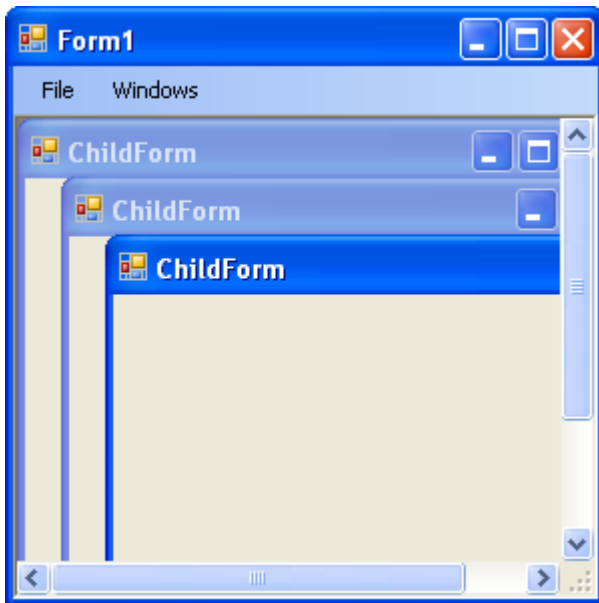
- Đoạn chương trình sắp xếp theo kiểu Tile Horizontal

```
private void mnuTile_Click ( Object sender , System.EventArgs e)  
{  
    this.LayoutMdi (MdiLayout.TileHorizontal);  
}
```

# Multiple Document Interface

- **Sắp đặt các child form**

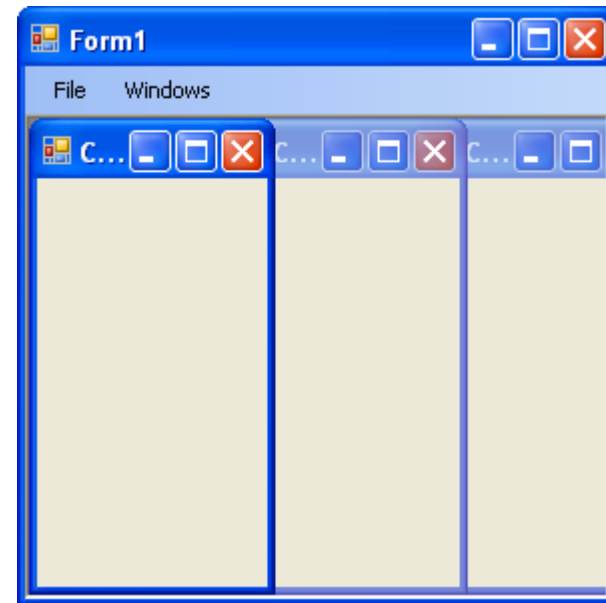
Cascade



TileHorizontal



TileVertical

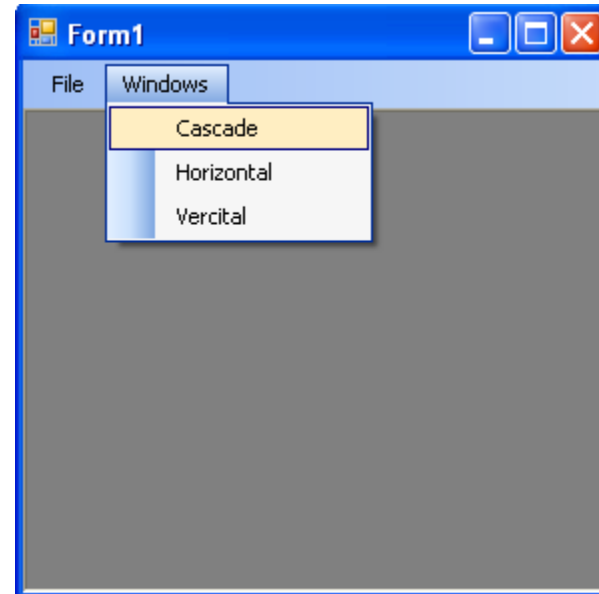
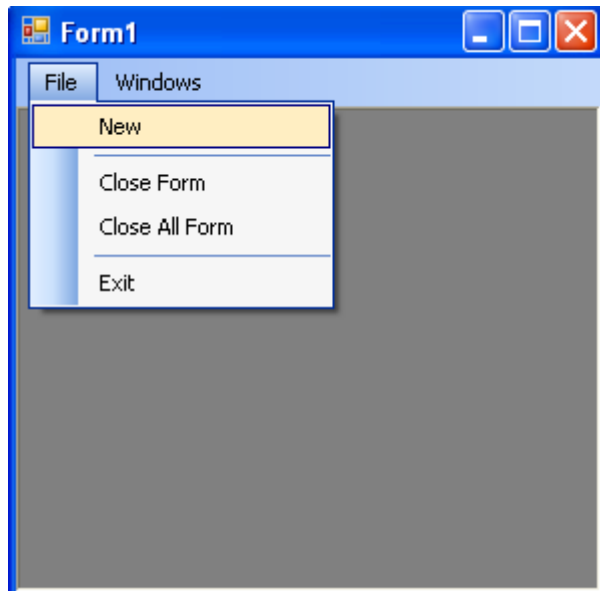


- Ngoài ra, có thể tạo ra các cách sắp xếp riêng tùy theo mỗi ứng dụng.
- Ví dụ, đoạn chương trình sau cho phép thu nhỏ lại tất cả các cửa sổ đang mở.

```
void mnuMinimizeAll_Click (Object sender,  
    System.EventArgs e)
```

```
{  
    foreach (Form frm in this.MdiChildren)  
        frm.WindowState = FormWindowState.Minimized;  
}
```

# Ví dụ MDI minh họa :



# Coding

**Viết sự kiện cho Menu New:**

```
frmChildForm frmChil = new frmChildForm();  
frmChil.MdiParent = this;  
frmChil.Show();
```

**Viết sự kiện cho Menu Close Form:**

```
if (this.ActiveMdiChild != null)  
    this.ActiveMdiChild.Close();
```

**Viết sự kiện cho Menu Close All Form:**

```
foreach (Form frm in this.MdiChildren)  
{  
    frm.Dispose();  
}
```

## Coding

***Viết sự kiện cho Menu Window Cascade:***  
***this.LayoutMdi(MdiLayout.Cascade);***

***Viết sự kiện cho Menu Window Horizontal:***  
***this.LayoutMdi(MdiLayout.TileHorizontal);***

***Viết sự kiện cho Menu Window Verital:***  
***this.LayoutMdi(MdiLayout.TileVertical);***

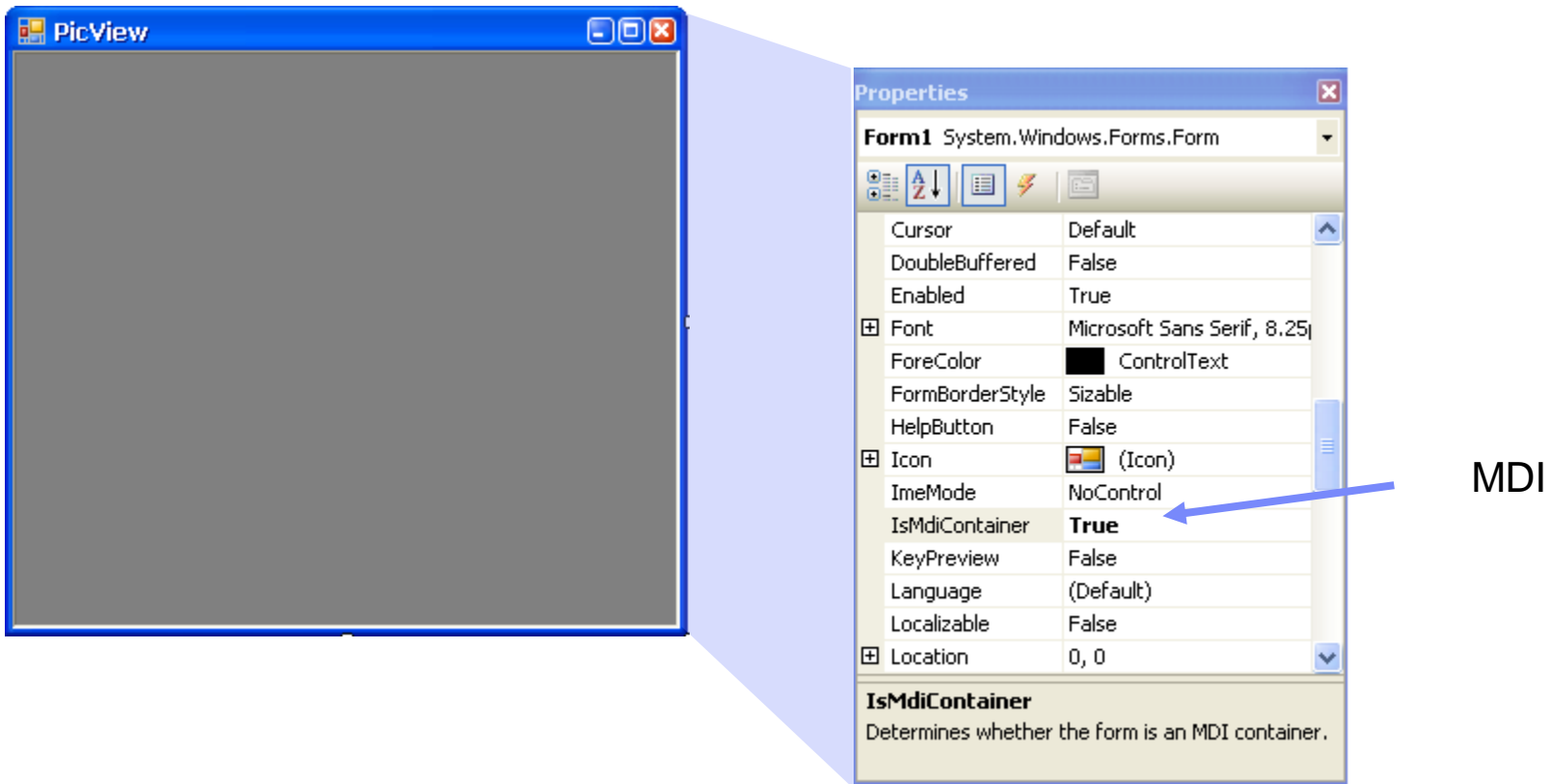
**Chạy và thử nghiệm ứng dụng**

## Bài tập MDI

- **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



The image shows a Windows Form titled 'PicView' with a gray background. To its right is the 'Properties' window for 'Form1' (System.Windows.Forms.Form). The 'IsMdiContainer' property is highlighted in yellow and set to 'True'. A blue arrow points from the text 'MDI' to this property.

Property	Value
Cursor	Default
DoubleBuffered	False
Enabled	True
Font	Microsoft Sans Serif, 8.25
ForeColor	ControlText
FormBorderStyle	Sizable
HelpButton	False
Icon	(Icon)
ImeMode	NoControl
<b>IsMdiContainer</b>	<b>True</b>
KeyPreview	False
Language	(Default)
Localizable	False
Location	0, 0

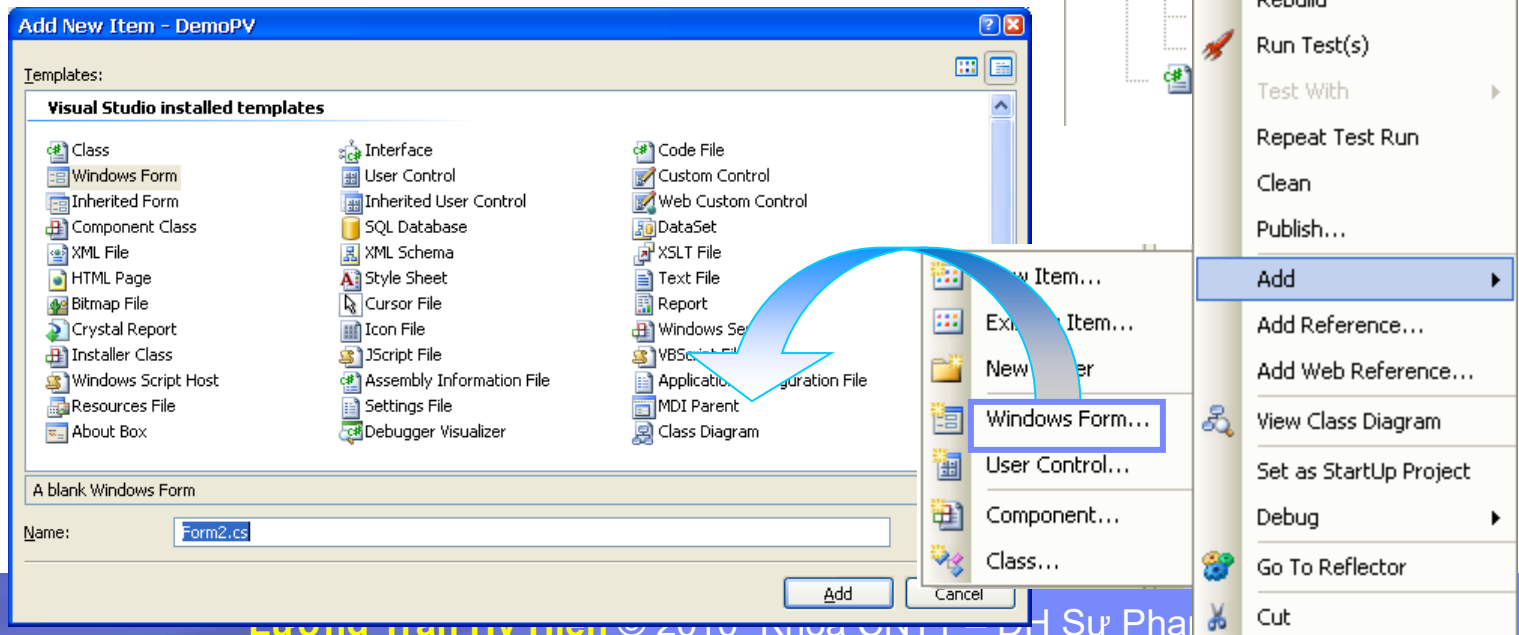
**IsMdiContainer**  
Determines whether the form is an MDI container.



# Multiple Document Interface

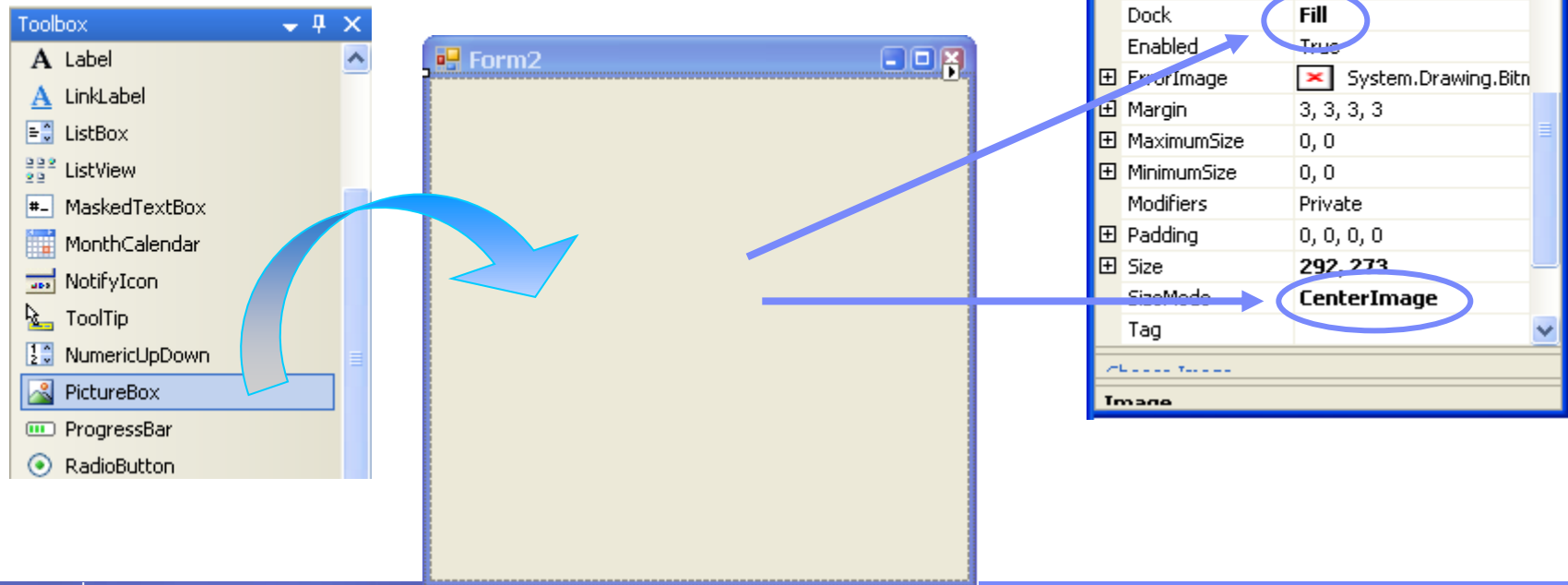
## ■ Bước 2: tạo form con

- 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

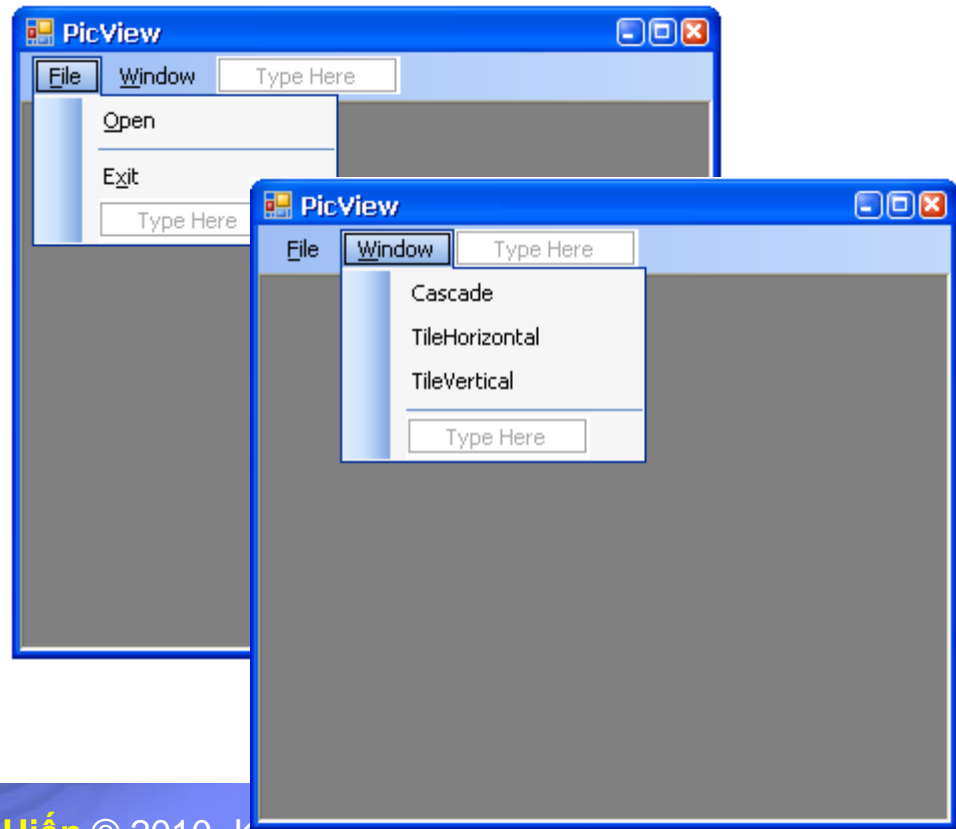
## ■ Bước 4: bổ sung constructor cho form2,

- 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 6: 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
    - Hiển 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





