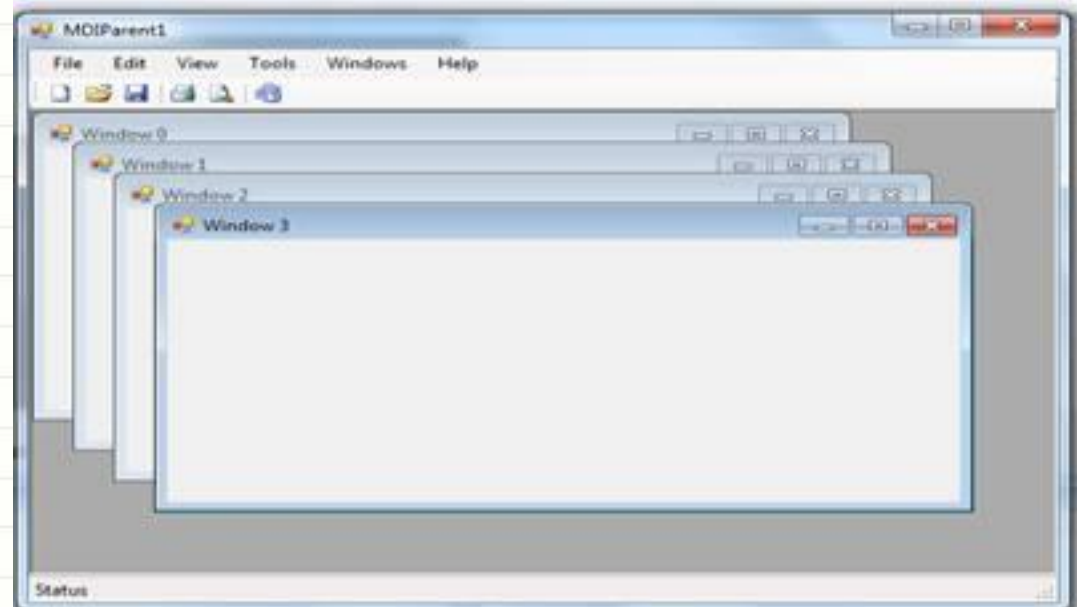


# LẬP TRÌNH ỨNG DỤNG MDI (MULTIPLE DOCUMENT INTERFACE)



# Nội dung

- 1 Giới thiệu
- 2 Cấu trúc ứng dụng MDI
- 3 Xây dựng ứng dụng MDI

# Giới thiệu

- Phần lớn các ứng dụng của Windows đều có giao diện thuộc một trong 3 loại giao diện sau:
  - Single Document Interface (SDI)
  - Explorer Interface
  - Multiple Document Interface (MDI)

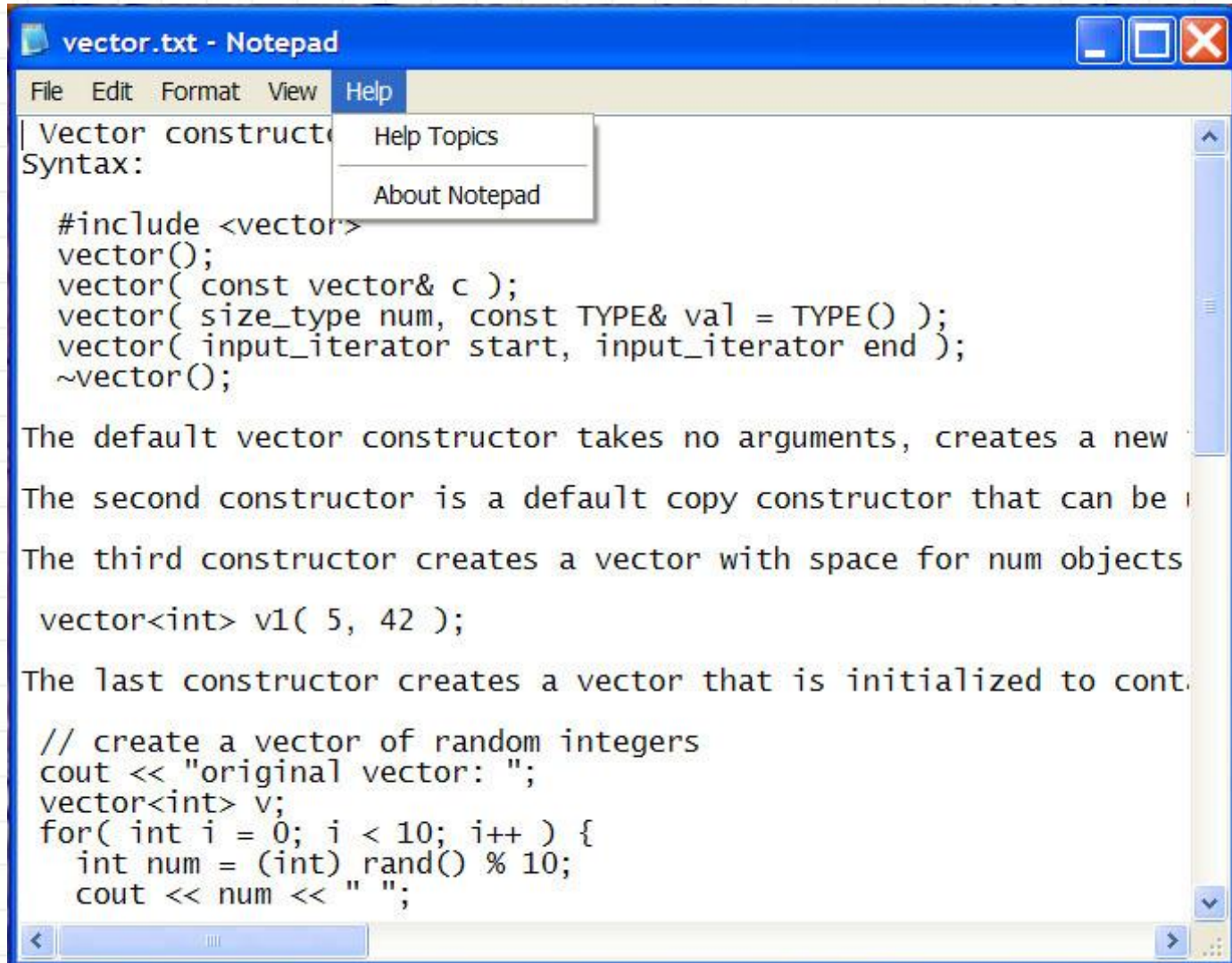
# Giới thiệu

## SDI:

- Mỗi tài liệu hay dữ liệu đóng gói sẽ được thể hiện ở một cửa sổ đơn.
- Trong Windows tiêu biểu cho loại giao diện này là Notepad hay Wordpad

# Giới thiệu

## SDI:



```
vector.txt - Notepad
File Edit Format View Help
Vector constructor
Syntax:
#include <vector>
vector();
vector( const vector& c );
vector( size_type num, const TYPE& val = TYPE() );
vector( input_iterator start, input_iterator end );
~vector();

The default vector constructor takes no arguments, creates a new
The second constructor is a default copy constructor that can be
The third constructor creates a vector with space for num objects
vector<int> v1( 5, 42 );

The last constructor creates a vector that is initialized to cont.

// create a vector of random integers
cout << "original vector: ";
vector<int> v;
for( int i = 0; i < 10; i++ ) {
    int num = (int) rand() % 10;
    cout << num << " ";
}
```



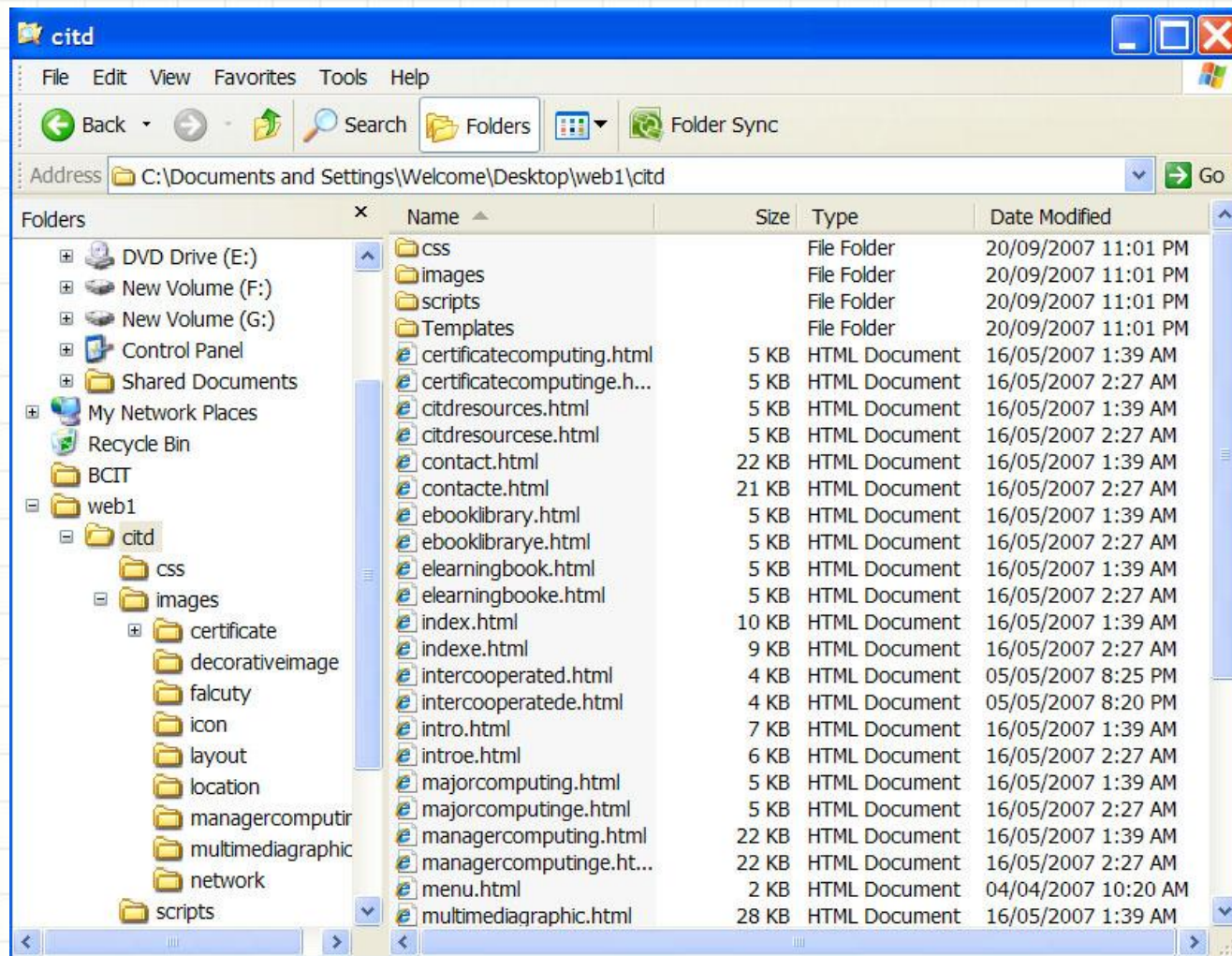
# Giới thiệu

## Explorer Interface:

- Đây là dạng form mà thông tin sẽ được hiển thị theo một hệ thống phân cấp.
- Thông thường một TreeView control sẽ dùng để hiển thị hệ thống phân cấp này.
- Thông tin chi tiết của mỗi nút trên TreeView được chọn sẽ hiển thị trong một ListView control

# Giới thiệu

## Explorer Interface:



# Giới thiệu

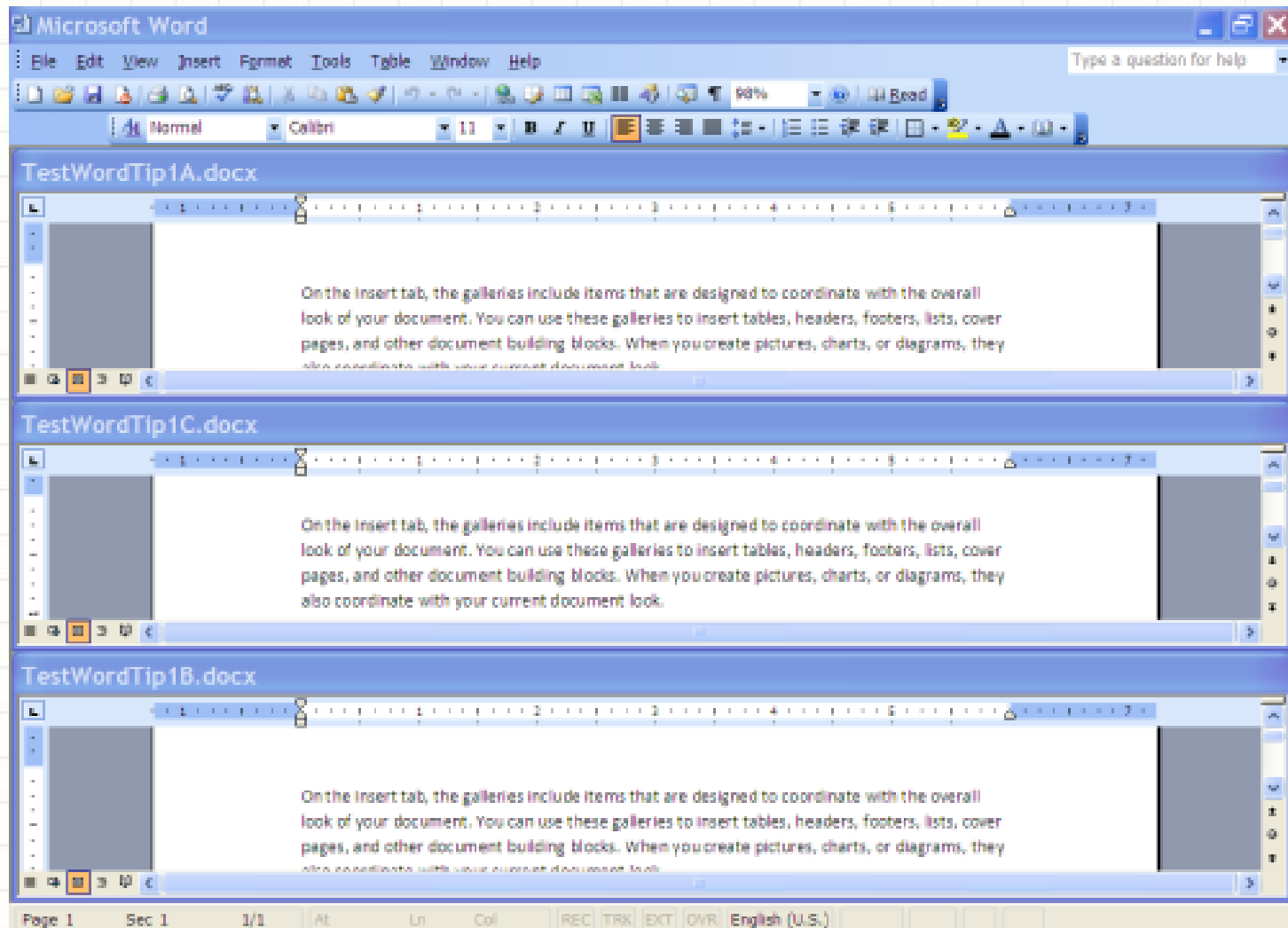
## MDI:

- 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
- Những ứng dụng tiêu biểu cho MDI: Word, Photoshop, ...



# Giới thiệu

## MDI:



# Giới thiệu

## So sánh MDI với SDI:

Tiêu chí	MDI	SDI
Cửa sổ	1 form cha chứa nhiều form con	Mỗi cửa sổ là một form riêng
Quản lý	Dễ quản lý nhiều tài liệu	Khó quản lý khi quá nhiều cửa sổ
Giao diện	Phức tạp hơn	Đơn giản
Ứng dụng	Vẽ, văn bản, hệ thống đa cửa sổ	Ứng dụng nhỏ

# Các trúc của ứng dụng MDI

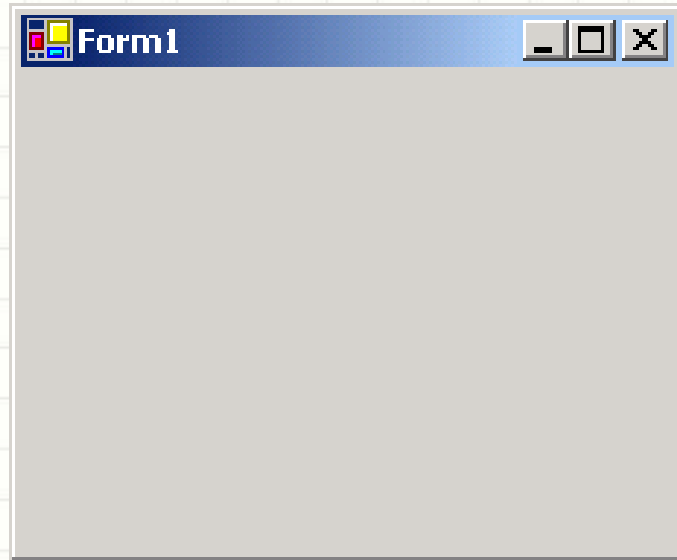
## Form cha (Parent / MDI Container)

- Là form chính
- Chứa menu, toolbar
- Chứa tất cả form con

## Form con (Child form)

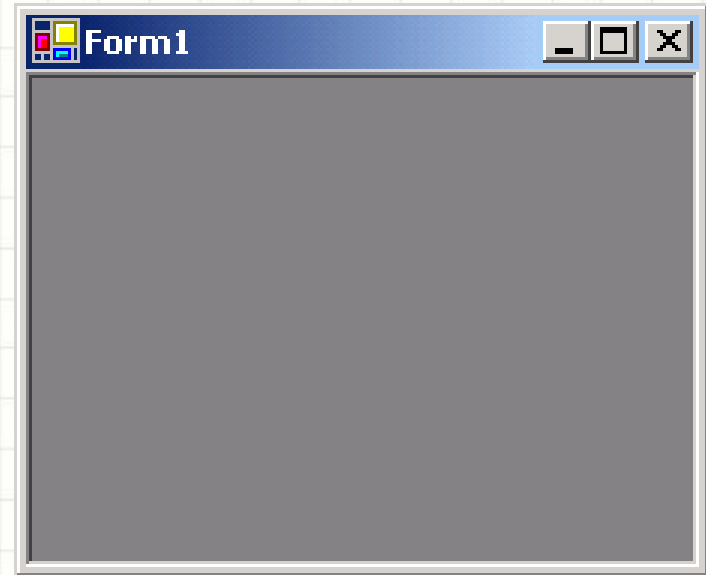
- Là các tài liệu hoặc cửa sổ chức năng
- Luôn nằm trong vùng làm việc của form cha
- Không thể đi ra ngoài

# Các trúc của ứng dụng MDI



SDI

Single Document Interface



MDI

Multiple Document Interface



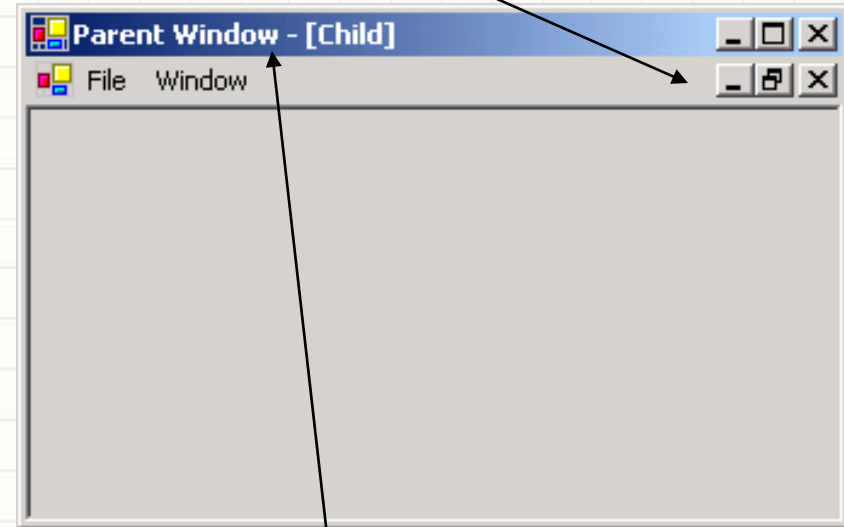
# Các trúc của ứng dụng MDI

Parent's icons: minimize, maximize and close



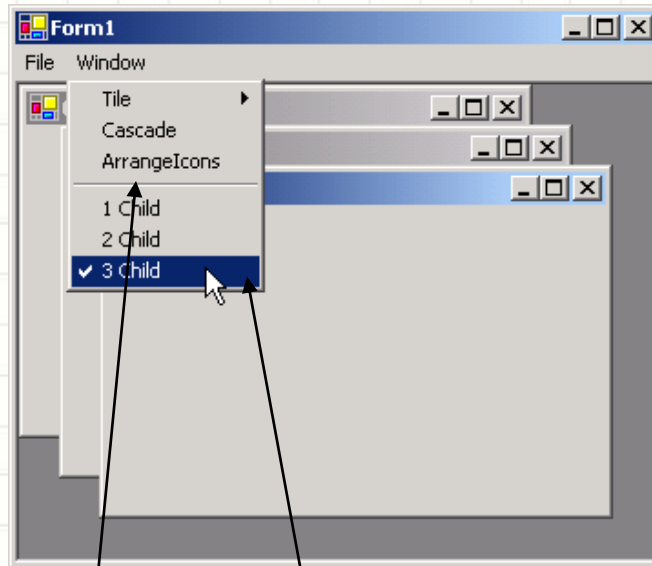
Minimized child's icons: restore, maximize and close

Maximized child's icons: minimize, restore and close



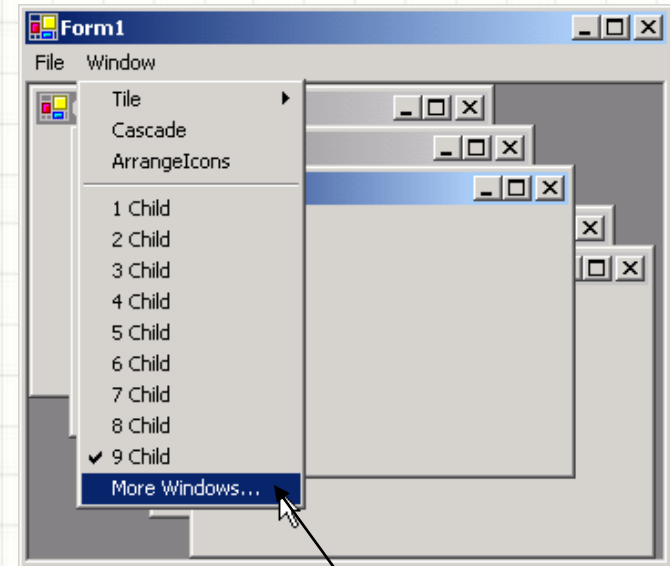
Parent's title bar displays maximized child

# Các trúc của ứng dụng MDI

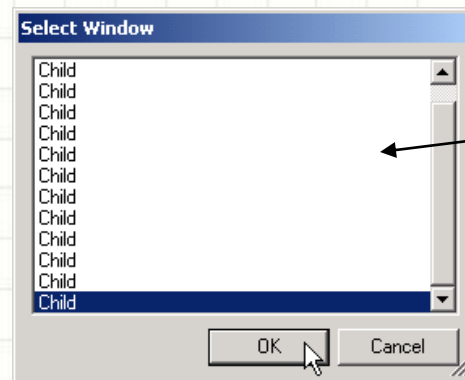


Separator bar and  
child windows

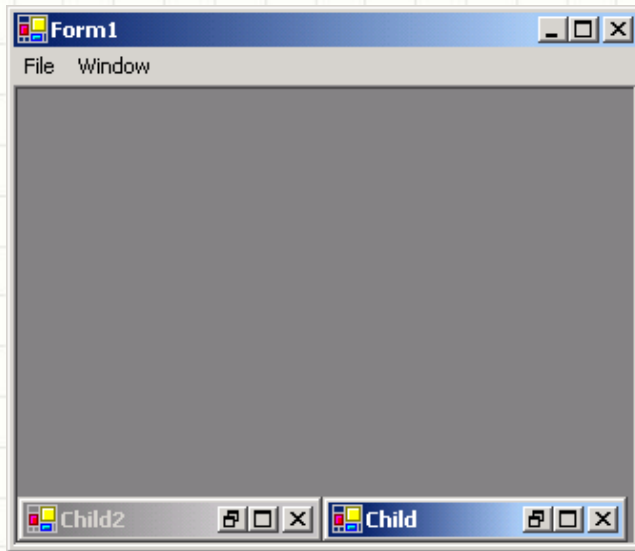
Child windows list



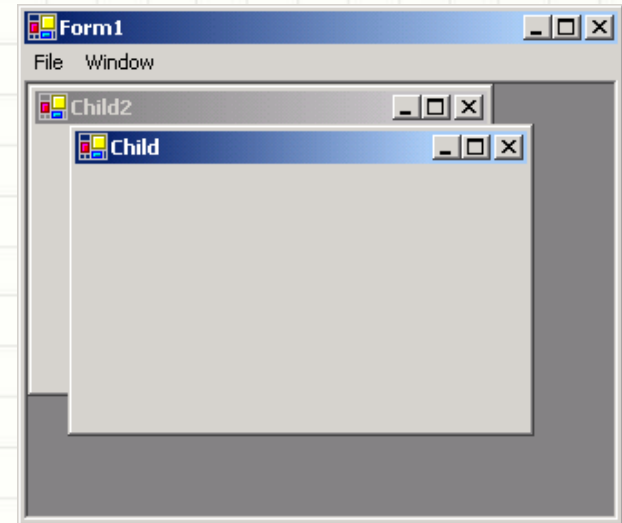
9 or more child windows  
enables the **More  
Windows...** option



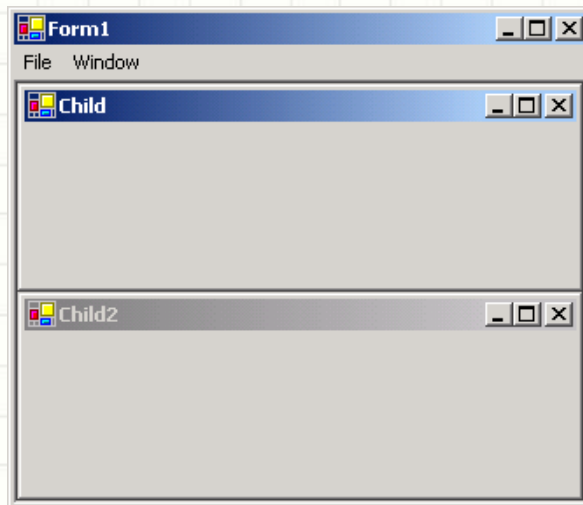
# Các trúc của ứng dụng MDI



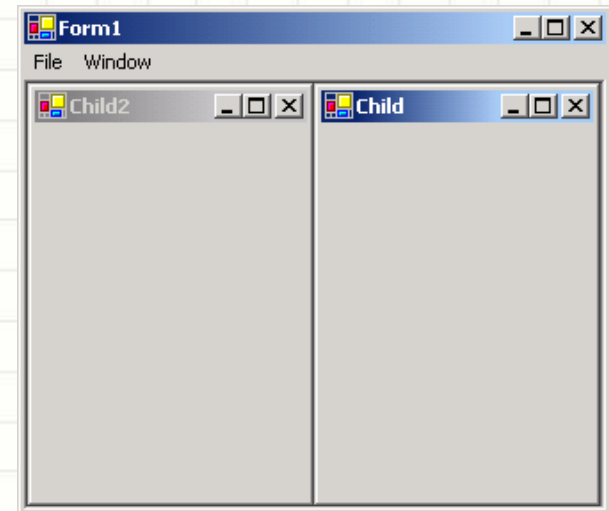
**ArrangeIcons**



**Cascade**



**TileHorizontal**



**TileVertical**

# Xây dựng ứng dụng MDI

## Các lớp thường dùng

Lớp	Loại	Tên	Mô tả
Form	Properties	ActiveMdiChild	Gets the MDI child window that is currently active.
		IsMdiChild	Gets whether the form is an MDI child.
		IsMdiContainer	Gets whether the form is an MDI container form
		MdiChildren	Gets the set of MDI children contained by this form as an array of <i>Form</i> object.
		MdiParent	Gets or sets the MDI container for this form. If set, then this form is an MDI child form.
		MergedMenu	Gets the <i>MainMenu</i> object representing the current merged menu for an MDI container form
	Methods	LayoutMdi	Arranges the MDI children within this form using a given layout style.



# Xây dựng ứng dụng MDI

Lớp	Loại	Tên	Mô tả
	Events	MdiChildActive	Occurs when an MDI child form is activated or deactivated with an MDI application . Note that MDI children do not receive the Activated and Deactivated events.
<b>Menu</b>	Properties	MdiListItem	Gets the <i>MenuItem</i> object contained by this menu that displays a list of MDI child forms for the associated form object.
	Method	MergeMenu	Merges the <i>MenuItem</i> objects in a given menu with those contained by this menu.
<b>MenuItem</b>	Properties	MdiList	Gets or sets whether this menu should be populated with a list of MDI child forms contained by the associated form.
		MergeOrder	Gets or sets the relative position of this menu item when it is merged with another menu.
		MergeType	Gets or sets how this menu should be merged with other menus. The default is <i>MergeType.Add</i>

# Xây dựng ứng dụng MDI

## Tạo Parent form:

- Đặt thuộc tính **isMdiContainer** là **True**
- Đặt thuộc tính **WindowState** là **Maximized**
- Tạo menu với các mục cơ bản:

### File

- New
- Open
- Close

### Windows

- Cascade
- Tile Horizontal
- Tile Vertical
- Arrange Icons

### Help

# Xây dựng ứng dụng MDI

Tạo Child form: (dựa vào sự kiện click trên menu New của Parent form)

```
Form newMDIChild = new Form();  
// Set the Parent Form of the Child window.  
newMDIChild.MdiParent = this;  
// Display the new form.  
newMDIChild.Show();
```

Thực tế thường ta tạo một Form template cho child form, để khi tạo child form mới ta sẽ dựa trên template này

# Xây dựng ứng dụng MDI

Đóng child form:

```
this.ActiveMdiChild.Close();
```

Đóng tất cả child form:

```
foreach (Form f in this.MdiChildren)  
    f.Close();
```



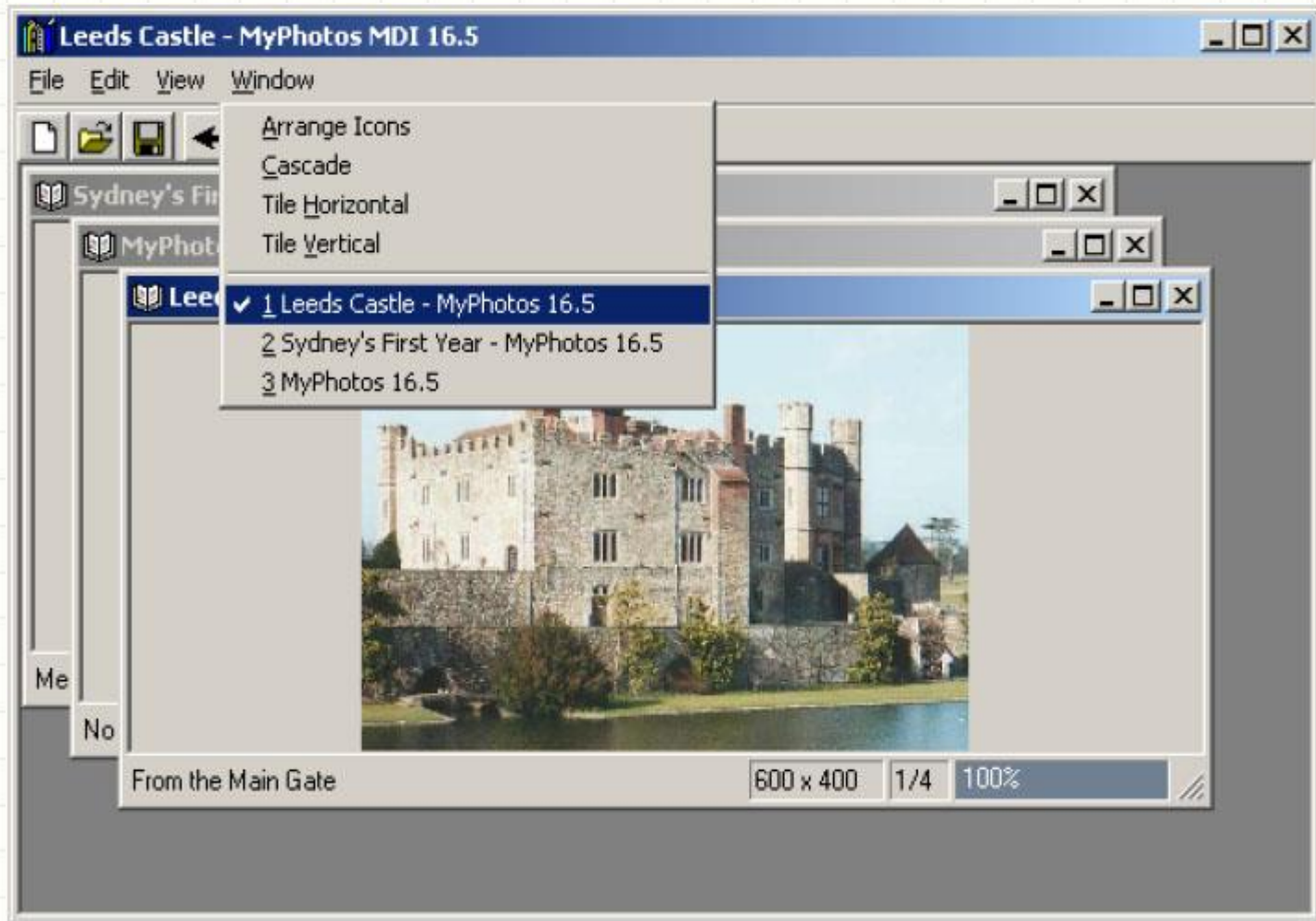
# Xây dựng ứng dụng MDI

Tạo danh sách child form để truy cập nhanh :

- 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

# Xây dựng ứng dụng MDI

Tạo danh sách child form để truy cập nhanh:



# Xây dựng ứng dụng MDI

## Sắp xếp các child form:

- Nếu muốn sắp xếp các child form, theo kiểu Cascade hoặc Tile chẳng hạn, cần thêm các chức năng này vào menu và gọi hàm sắp xếp **LayoutMdi**
- Mỗi MDI container đều hỗ trợ hàm **LayoutMdi()**, hàm này nhận giá trị từ Enumeration **MdiLayout** và sắp xếp tự động các cửa sổ

# Xây dựng ứng dụng MDI

Sắp xếp các child form:

- **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 mnuTileH_Click ( Object sender ,  
    System.EventArgs e)  
{  
    this.LayoutMdi (MdiLayout.TileHorizontal)  
}
```



# Xây dựng ứng dụng MDI

Sắp xếp các child form:

LayoutMdi(MdiLayout.Cascade);

LayoutMdi(MdiLayout.TileHorizontal);

LayoutMdi(MdiLayout.TileVertical);

LayoutMdi(MdiLayout.ArrangeIcons);