

CHƯƠNG 2: WINDOWS APPLICATION (3)

1

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

Xử lý các sự kiện nhập liệu bàn phím, chuột

2

XỬ LÝ SỰ KIỆN BÀN PHÍM (KEYBOARD)
XỬ LÝ SỰ KIỆN CHUỘT (MOUSE)

Giới thiệu

3

- Tìm hiểu thông điệp phát sinh từ bàn phím hay thiết bị chuột để viết các xử lý tương ứng
- Bộ định thời gian: windows cung cấp cơ chế này để truyền thông với ứng dụng theo định kỳ.
 - Ứng dụng cần khai báo bộ định thời gian với 1 khoảng thời gian cho trước
 - Khi ứng dụng hoạt động thì hệ thống sẽ truyền 1 tín hiệu cho ứng dụng theo từng khoảng thời gian định kỳ đã được khai báo
- Bàn phím và chuột được xử lý qua cơ chế thông điệp của windows.
- Mọi sự kiện đối với bàn phím và chuột được Windows gửi đến chương trình thông qua các thông điệp

Bàn phím

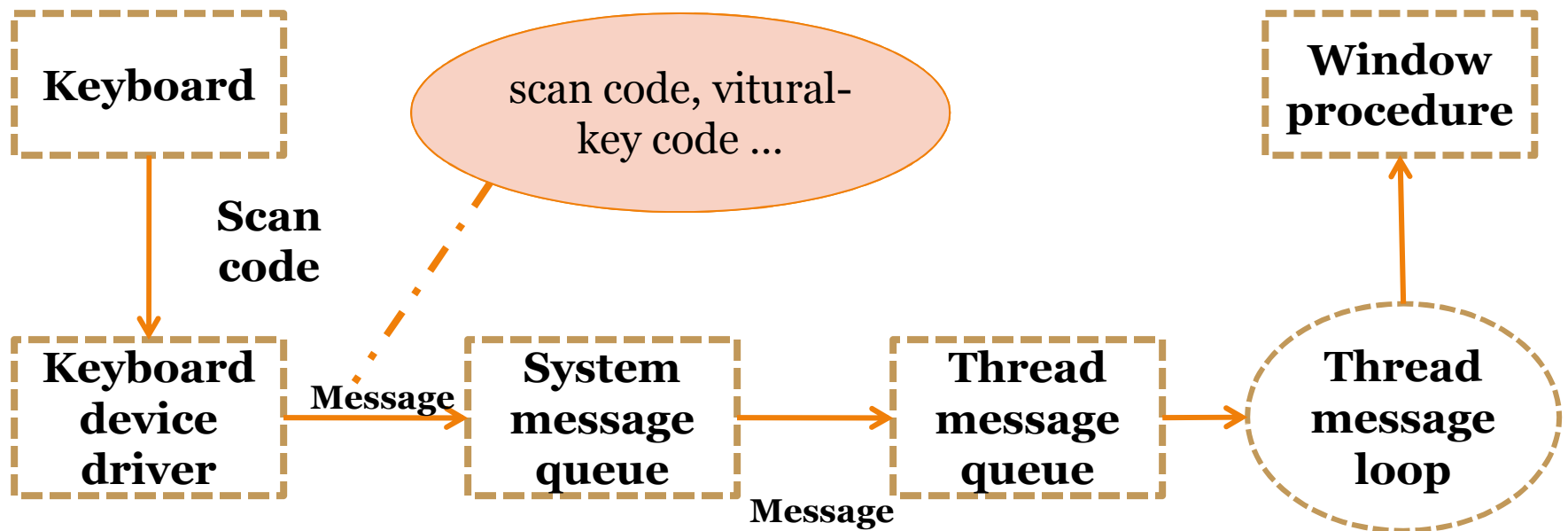
4

- Khi nhấn phím có thể xảy ra trường hợp sau:
 - Nhấn 1 phím ký tự
 - Nhấn 1 phím điều khiển (Enter, Esc, F1→F12)
 - Nhấn Shift hoặc Ctrl hoặc Alt hoặc tổ hợp nào đó của 3 phím này với các ký tự.
- Khi phím nào đó trên bàn phím được gõ, nhả hay giữ thì các thông điệp tương ứng sẽ được gửi đến cửa sổ đang được focus
- Các phím được nhấn được phân thành 2 nhóm chính:
 - Nhóm các phím hệ thống (System keys): là các phím được nhấn với phím Alt.
 - Nhóm các phím thường (nonsystem keys): Khi phím Alt không được nhấn.

Xử lý sự kiện bàn phím (Event Keyboard)

5

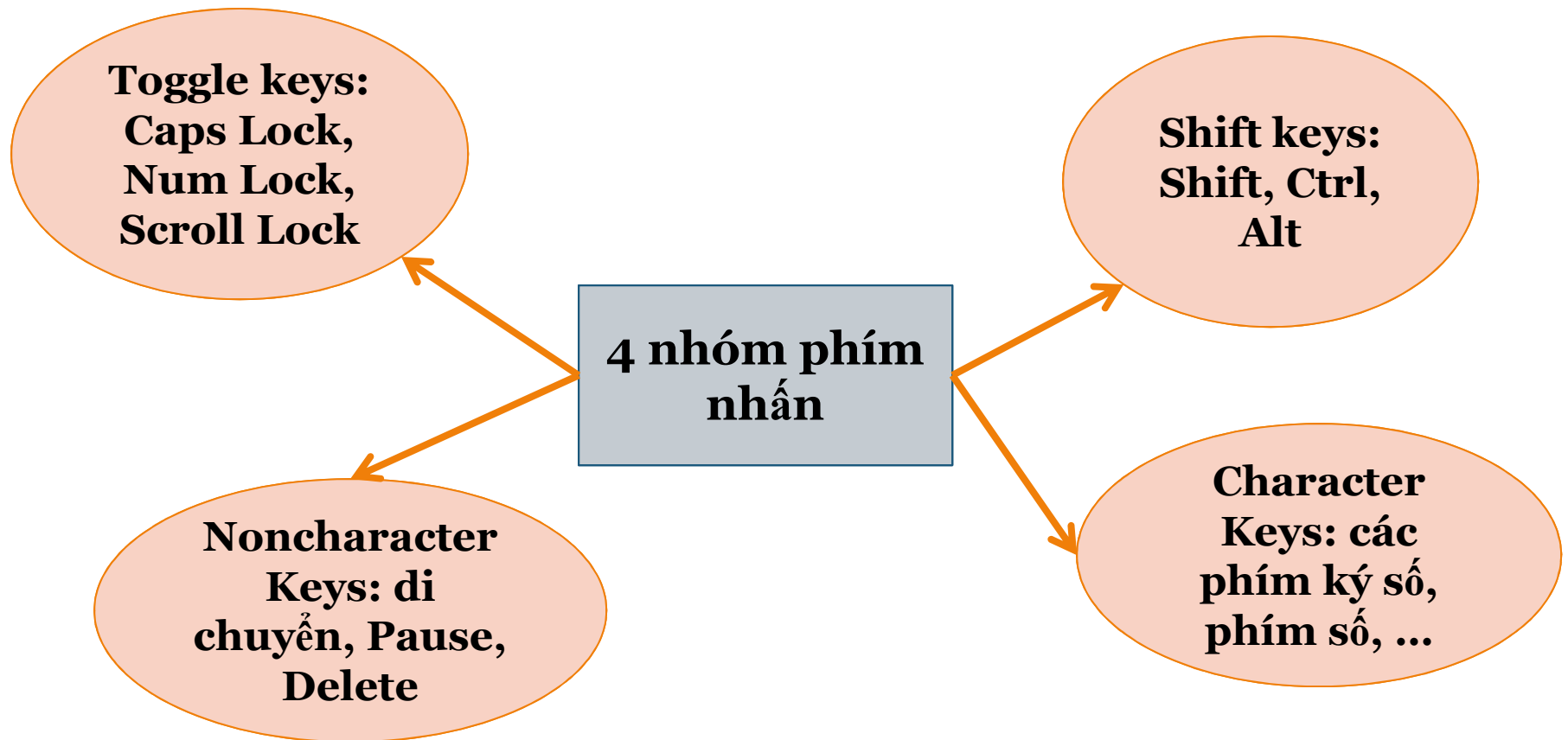
- Mô hình xử lý sự kiện bàn phím của Windows



Xử lý sự kiện bàn phím (Event Keyboard)

6

- Các phím được nhấn được chia làm 4 nhóm:



Keyboard Event

7

- Phát sinh khi một phím được nhấn hoặc thả

KeyPress

Phát sinh kèm theo mã ASCII của phím được nhấn

KeyUp

Phát sinh không cho biết trạng thái các phím bổ sung (Shift, Alt, Ctrl,...)

KeyDown

Sử dụng KeyUp & KeyDown để xác định trạng thái bổ sung

Keyboard Event

Sự kiện với tham số kiểu KeyEventArgs

KeyDown	Phát sinh khi phím được nhấn
KeyUp	Phát sinh khi phím được thả
Thuộc tính của lớp KeyEventArg	
Alt, Control, Shift	Trạng thái phím bổ sung
Handled	Cho biết sự kiện đã xử lý
KeyCode	Trả về mã ký tự được định nghĩa trong Keys enumeration
KeyData	Chứa mã ký tự với thông tin phím bổ sung
KeyValue	Trả về số int, đây chính là mã Windows Virtual Key Code
Modifier	Trả về giá trị của phím bổ sung

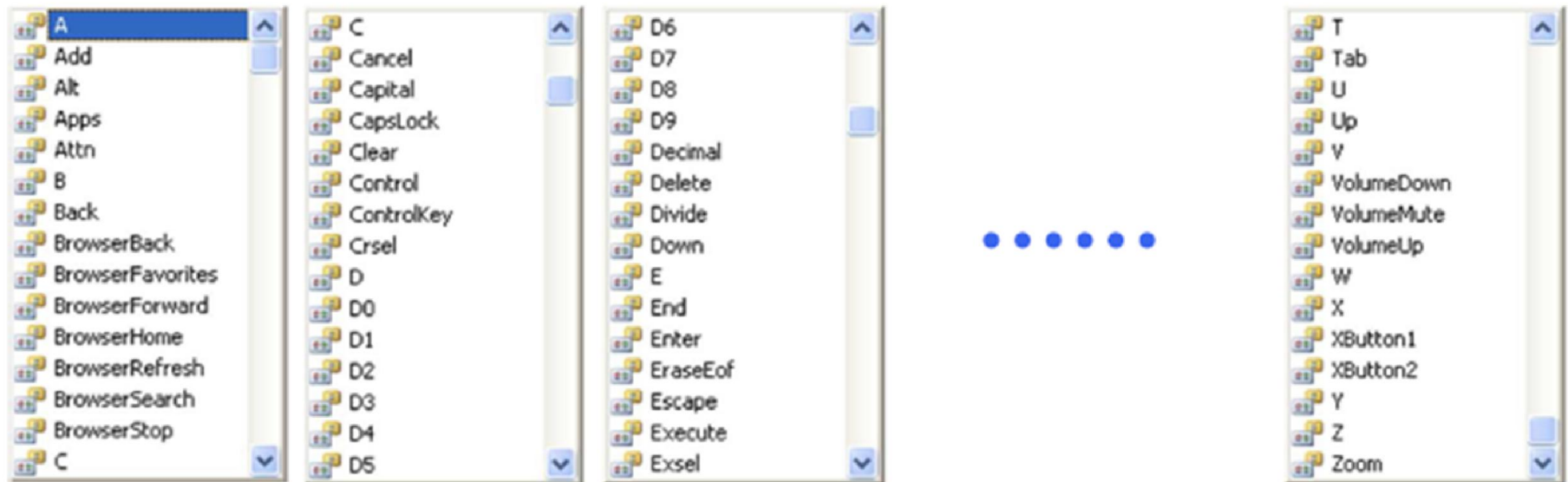
Sự kiện với tham số kiểu KeyPressEventArgs

KeyPress	Khởi tạo khi phím được nhấn
Thuộc tính của lớp KeyPressEventArgs	
KeyChar	Chứa ký tự ASCII của phím được nhấn
Handled	Cho biết sự kiện KeyPress có được xử lý chưa

Keyboard Event

9

- Keys Enumeration:



Key Enumeration

10

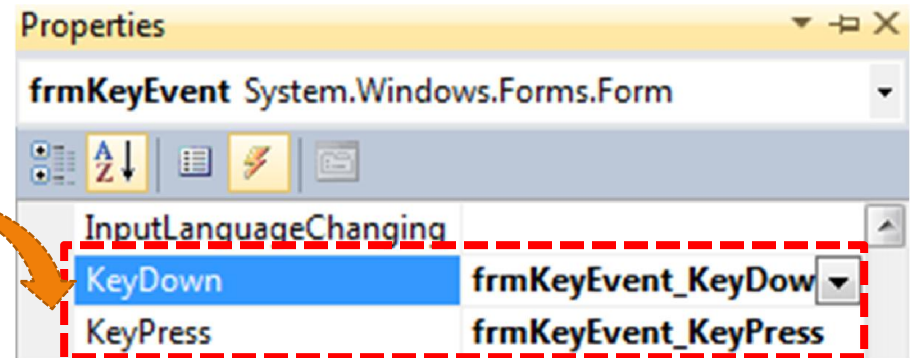
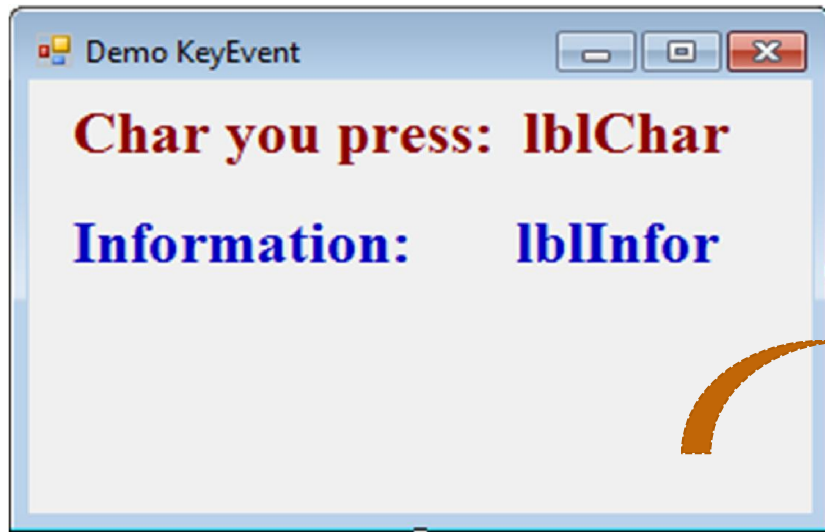
Keys Enumeration (letters)

Member	Value	Member	Value
<i>A</i>	65	<i>N</i>	78
<i>B</i>	66	<i>O</i>	79
<i>C</i>	67	<i>P</i>	80
<i>D</i>	68	<i>Q</i>	81
<i>E</i>	69	<i>R</i>	82
<i>F</i>	70	<i>S</i>	83
<i>G</i>	71	<i>T</i>	84
<i>H</i>	72	<i>U</i>	85
<i>I</i>	73	<i>V</i>	86
<i>J</i>	74	<i>W</i>	87
<i>K</i>	75	<i>X</i>	88
<i>L</i>	76	<i>Y</i>	89
<i>M</i>	77	<i>Z</i>	90

Keys Enumeration (function keys)

Member	Value	Member	Value
<i>F1</i>	112	<i>F13</i>	124
<i>F2</i>	113	<i>F14</i>	125
<i>F3</i>	114	<i>F15</i>	126
<i>F4</i>	115	<i>F16</i>	127
<i>F5</i>	116	<i>F17</i>	128
<i>F6</i>	117	<i>F18</i>	129
<i>F7</i>	118	<i>F19</i>	130
<i>F8</i>	119	<i>F20</i>	131
<i>F9</i>	120	<i>F21</i>	132
<i>F10</i>	121	<i>F22</i>	133
<i>F11</i>	122	<i>F23</i>	134
<i>F12</i>	123	<i>F24</i>	135

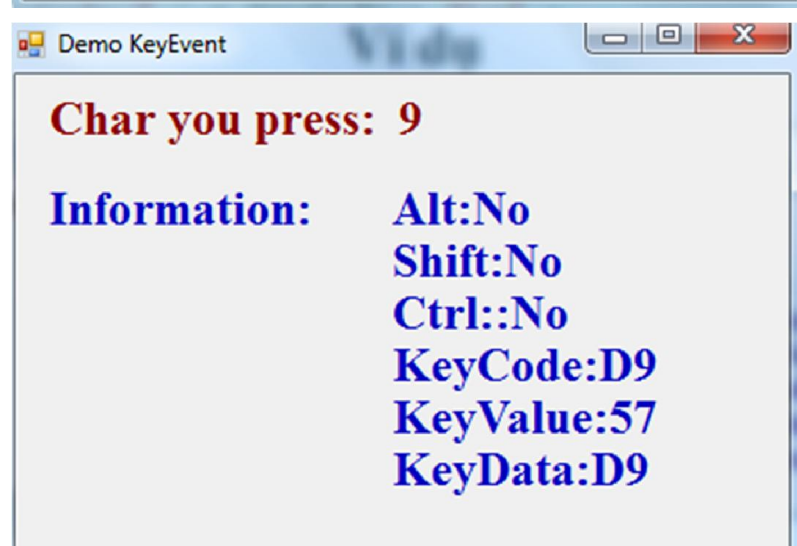
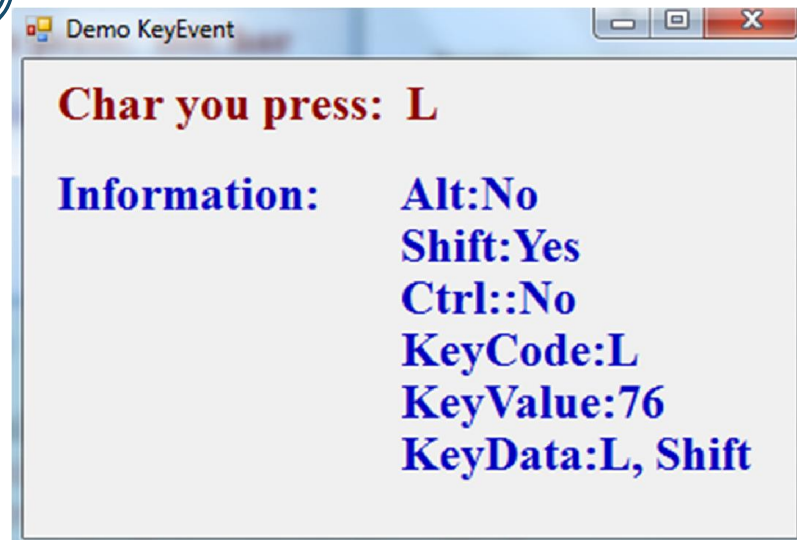
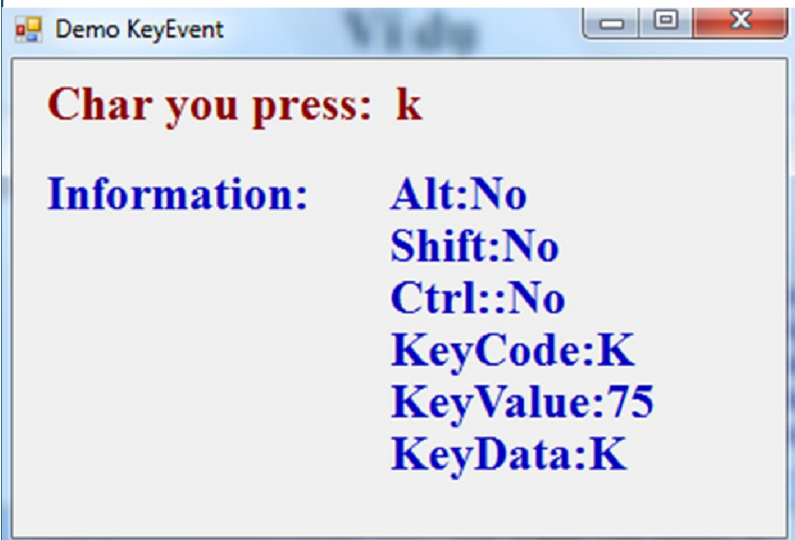
Ví dụ minh họa



```
private void frmKeyEvent_KeyDown(object sender, KeyEventArgs e)
{
    this.lblInfor.Text = "Alt:"+(e.Alt ? "Yes" : "No")+'\n'+
        "Shift:" + (e.Shift ? "Yes" : "No") + '\n' +
        "Ctrl::" + (e.Control ? "Yes" : "No") + '\n' +
        "KeyCode:" + e.KeyCode + "\n" +
        "KeyValue:" + e.KeyValue + "\n" +
        "KeyData:" + e.KeyData;
}
private void frmKeyEvent_KeyPress(object sender, KeyPressEventArgs e)
{
    this.lblChar.Text = e.KeyChar.ToString();
}
```

Ví dụ

12



Mouse Event

13

- Các thông điệp tạo từ chuột:
 - Di chuyển chuột qua cửa sổ
 - Click vào trong cửa sổ
 - Thậm chí cả trong trường hợp cửa sổ không được kích hoạt hoặc không nhận được sự quan tâm.
- Các sự kiện chuột gửi đến cho:
 - Cửa sổ hiện tạo đang chứa con trỏ chuột
 - Hoặc cửa sổ đang “capture” chuột.
- Có 2 loại sự kiện chuột:
 - Client area message: các sự kiện xảy ra khi chuột đang ở vùng client của cửa sổ
 - Non-client area message: các sự kiện chuột xảy ra khi chuột đang ở các vùng như border, menu bar, title bar, scroll bar, window menu, minimize button và maximize button

Mouse Event

14

MouseEventArgs

- Thao tác phát sinh từ mouse:
 - Di chuyển
 - Kích chuột
- Lớp MouseEventArgs chứa thông tin truyền vào cho trình xử lý sự kiện mouse.(hoặc EventArgs)

Sự kiện chuột với tham số kiểu EventArgs

- **MouseEnter**: xuất hiện khi chuột trở vào vùng biên của control
- **MouseLeave**: xuất hiện khi con trỏ rời khỏi vùng biên của control

Sự kiện chuột với tham số kiểu MouseEventArgs

- **MouseDown/MouseUp**: Xuất hiện khi button nhấn/thả và con trỏ đang ở trong vùng biên của control
- **MouseMove**: xuất hiện khi chuột di chuyển và con trỏ ở trong vùng biên của control

MouseEvent

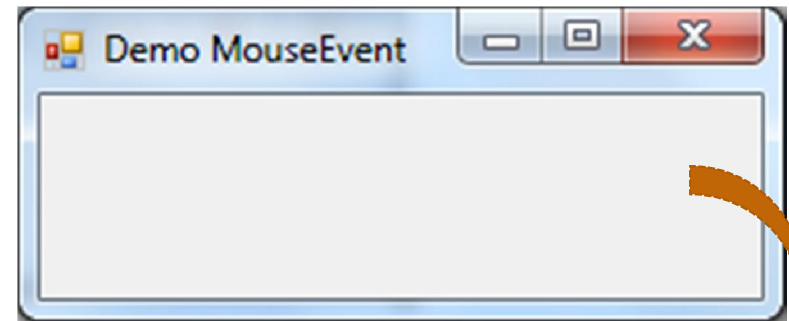
15

Thuộc tính của lớp MouseEventArgs

Button	Button được nhấn {Left, Right, Middle, None} có kiểu MouseButton
Clicks	Số lần button được nhấn
x	Tọa độ x của con trỏ chuột trong control
y	Tọa độ y của con trỏ chuột trong control

Ví dụ

```
protected override void OnMouseClicked(MouseEventArgs e)
{
    string s = "Bạn đang nhấn chuột ";
    switch (e.Button)
    {
        case MouseButton.Left:
            s += "Trái";
            break;
        case MouseButton.Middle:
            s += "Giữa";
            break;
        case MouseButton.Right:
            s += "Phải";
            break;
        default:
            break;
    }
    MessageBox.Show(s);
}
```



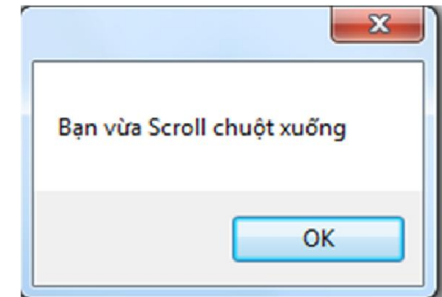
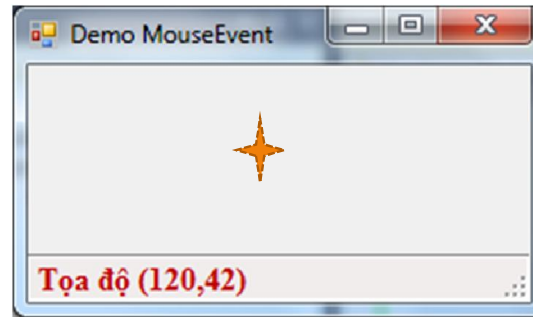
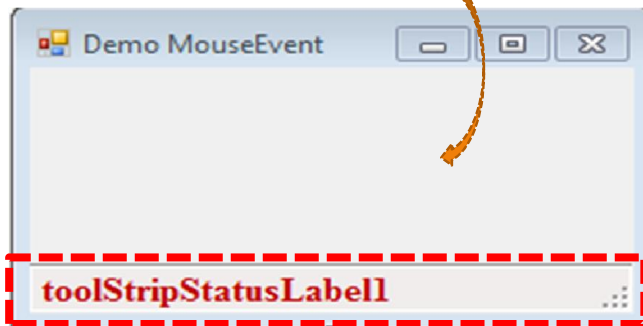
override phương thức
OnMouseClicked Hoặc
OnMouseDown

```
protected override void OnMouseDown(MouseEventArgs e)
{
    MessageBox.Show("Bạn vừa nhấn chuột " + e.Button);
}
```

```
protected override void OnMouseUp(MouseEventArgs e)
{
    MessageBox.Show("Bạn vừa nhả chuột " + e.Button);
}
```


MouseMove và MouseWheel

Thiết kế



```
protected override void OnMouseWheel(MouseEventArgs e)
{
    if (e.Delta > 0)
        MessageBox.Show("Bạn vừa Scroll chuột lên");
    else
        MessageBox.Show("Bạn vừa Scroll chuột xuống");
}
protected override void OnMouseMove(MouseEventArgs e)
{
    string s;
    s = "Tọa độ (" + e.X + "," + e.Y + ")";
    this.toolStripStatusLabel1.Text=s;
}
```

Sự kiện Click và DoubleClick

18

- Được phát sinh khi một button bất kỳ của chuột được nhấn
- Khi DoubleClick sẽ phát sinh 1 dãy các sự kiện sau:
 - MouseDown
 - Click
 - MouseUp
 - MouseMove
 - DoubleClick

MDI

(Multiple Document Interface)

19

1. **GIỚI THIỆU**
2. **CÁC THÀNH PHẦN CỦA MDI**
3. **CÁCH TẠO MDI**
4. **THUỘC TÍNH**
5. **MỘT SỐ THAO TÁC TRÊN MDI**

MDI

(Multiple Document Interface)

20

1. **GIỚI THIỆU**
2. **CÁC THÀNH PHẦN CỦA MDI**
3. **CÁCH TẠO MDI**
4. **THUỘC TÍNH**
5. **MỘT SỐ THAO TÁC TRÊN MDI**

Giới thiệu

21

- Windows form có 2 loại tài liệu cơ bản:

Document Interface

Single Document Interface (SDI)

Chỉ hỗ trợ 1 document hoặc 1 cửa sổ

VD như NotePad hoặc MS Paint

Để làm việc nhiều tài liệu thì phải mở nhiều Instance

Quản lý SDI đơn giản

Multiple Document Interface (MDI)

MDI là 1 đặc tả quản lý tài liệu trong Microsoft Windows

Mô tả cấu trúc cửa sổ và giao diện người dùng sử dụng làm việc với nhiều tài liệu trong 1 ứng dụng đơn

MDI xuất hiện từ Windows 2.0 nhưng rất khó viết

Từ Windows 3.0 trở đi đã có nhiều chức năng được mở rộng

Các thành phần MDI

22

Thanh tiêu đề

1 trình đơn

Một đường viền thay đổi kích thước

Một Icon trình đơn hệ thống

Các nút minimize/maximize/close.

Vùng client thường gọi là “Workspace”

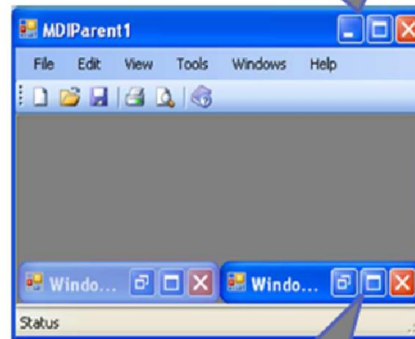
Workspace có thể không chứa, hoặc chứa nhiều cửa sổ con

Tại 1 thời điểm chỉ 1 cửa sổ được kích hoạt

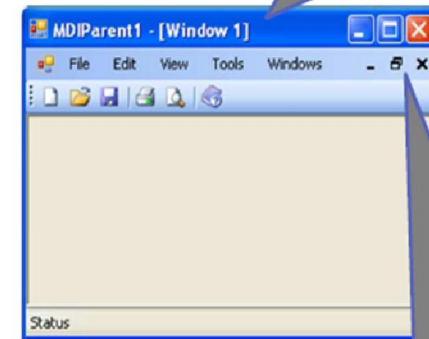
Tất cả các cửa sổ con đều được giới hạn trong workspace

Parent's icons: minimize, maximize and close

Parent's title bar displays maximized child



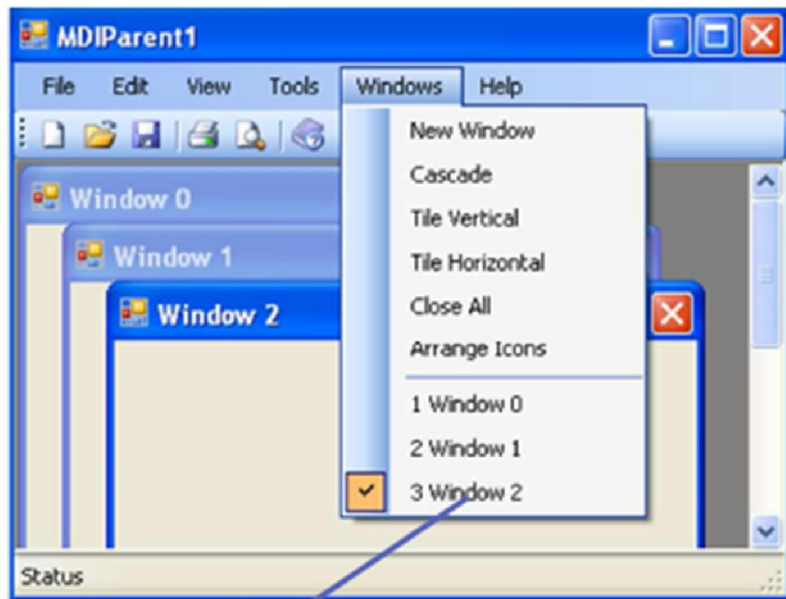
Minimized child's icons: restore, maximize and close



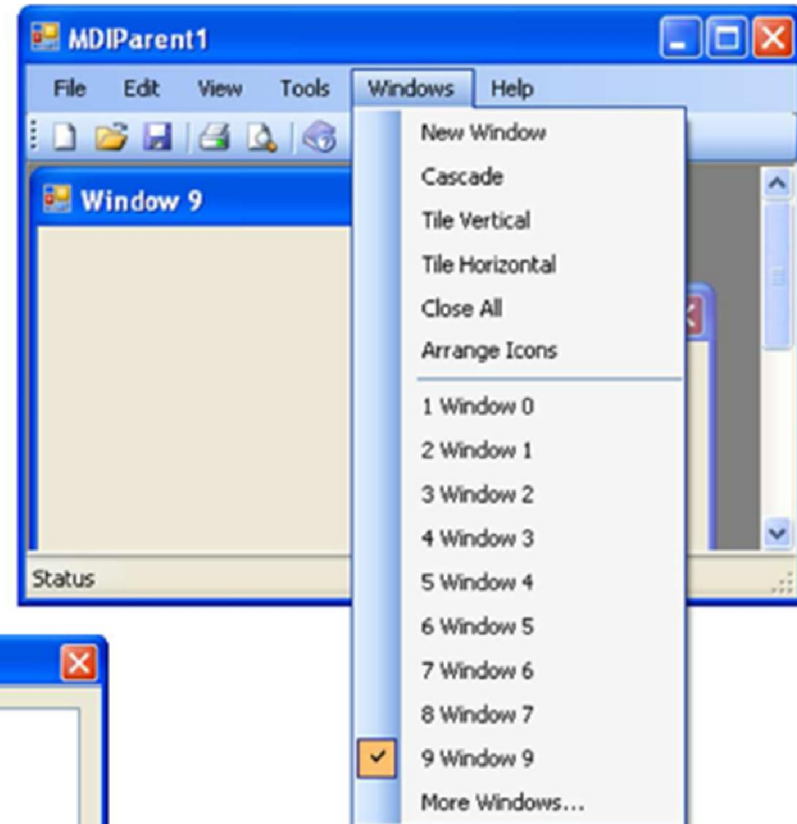
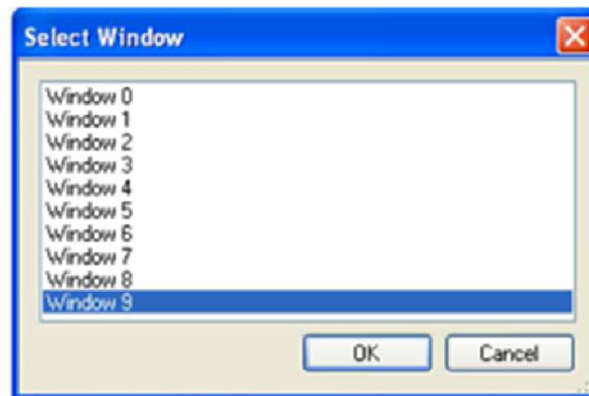
Maximized child's icons: restore, maximize and close

Các thành phần của MDI

23



Child windows list

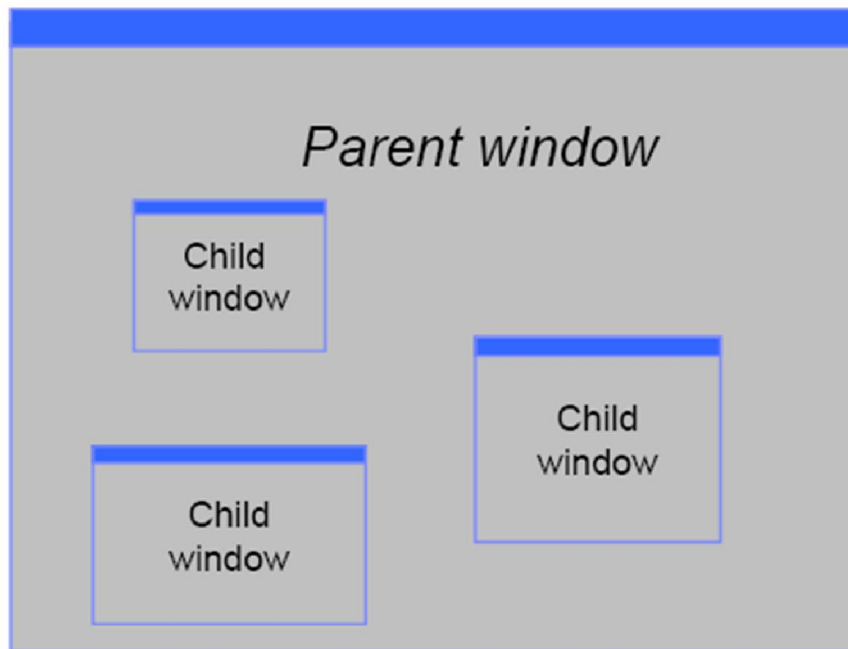


9 or more child windows
enables the More
Windows... option

Các thành phần MDI

24

- MDI parent window – MDI child window



Chỉ có duy nhất
1 parent window

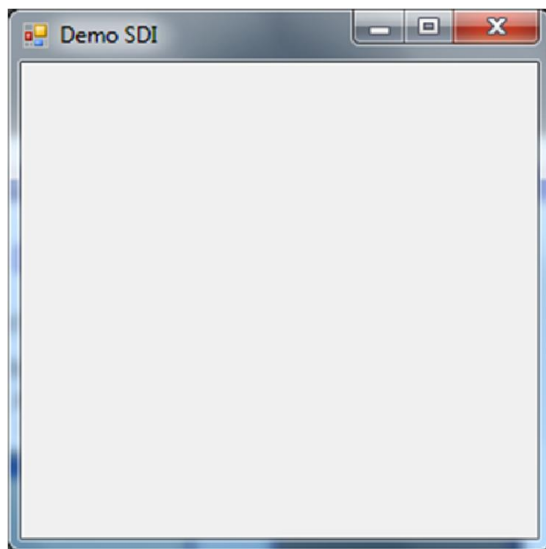
Có nhiều child
windows

Cách tạo MDI

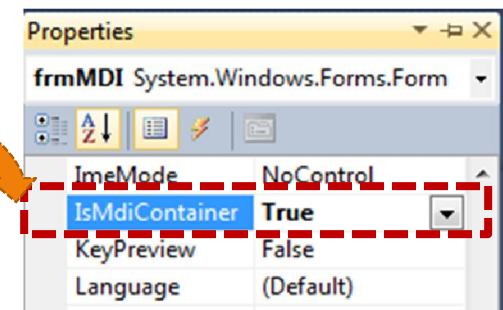
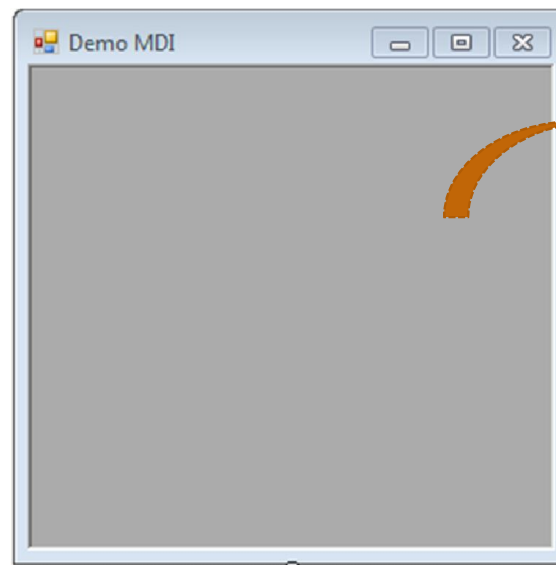
25

- Tạo Form và thiết lập thuộc tính IsMdiContainer=true.

SDI



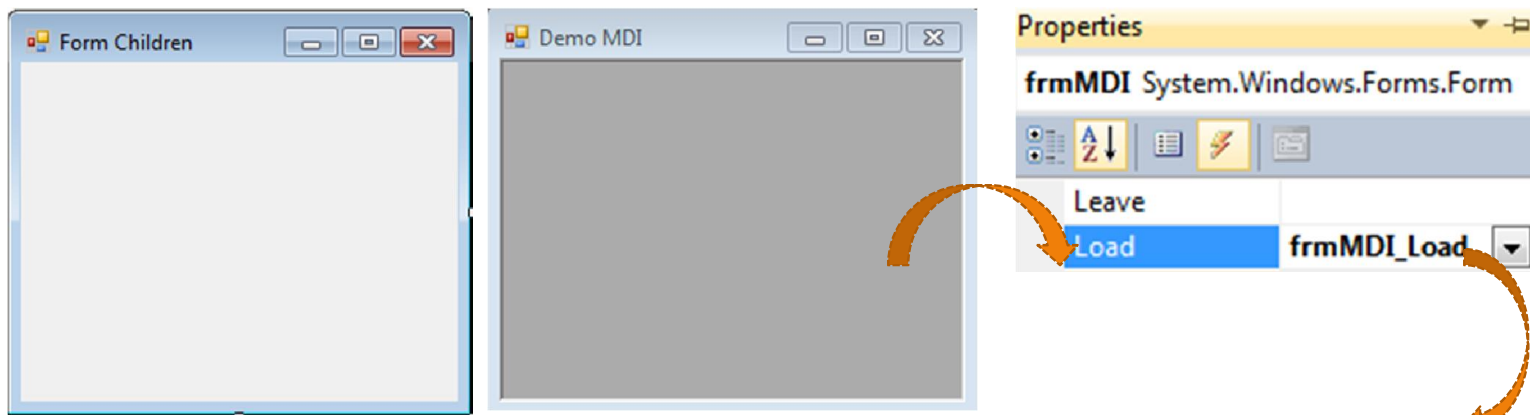
MDI



Cách tạo MDI

26

- 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.



frmChild

```
private void frmMDI_Load(object sender, EventArgs e)
{
    frmChild frm = new frmChild();
    frm.MdiParent = this;
    frm.Show();
}
```

Thuộc tính MDI child và MDI parent

27

Thuộc tính MDI child

IsMdiChild	Cho biết form là child hay không?
------------	-----------------------------------

MdiParent	Xác định form cha của Form
-----------	----------------------------

Thuộc tính 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 của con
-----------	--------------------------------

Sự kiện

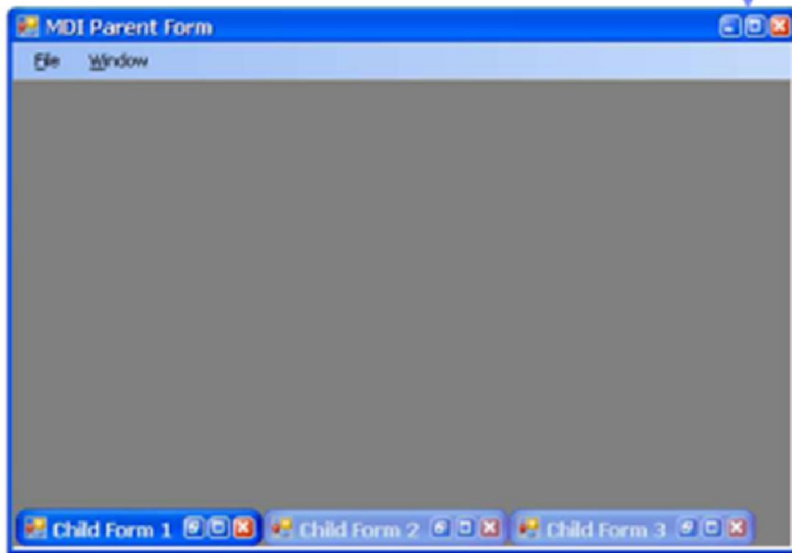
MdiChildActivate	Xác định khi Child form đóng hay Active
------------------	---

Thuộc tính MDI child và MDI parent

28

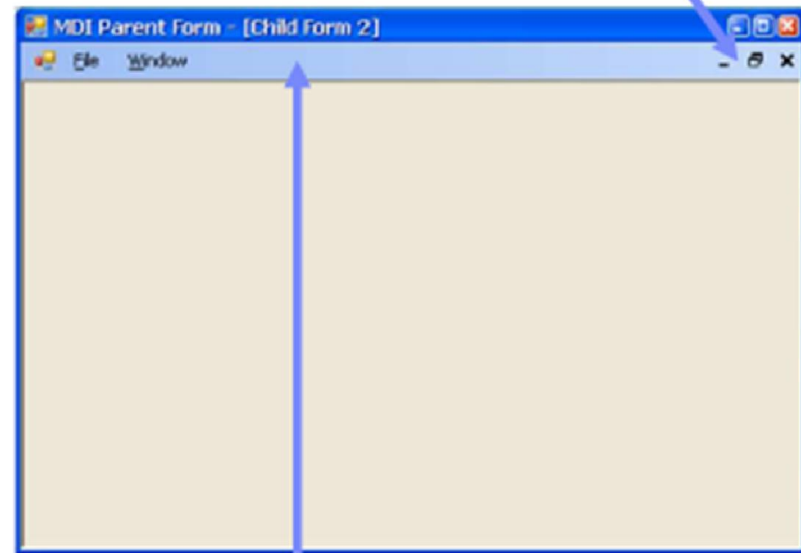
- Trạng thái Minimize, Maximize hay Close

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



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

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



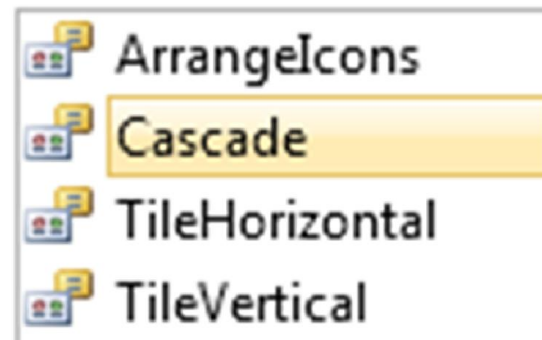
Title của parent chứa tên child đang maximize

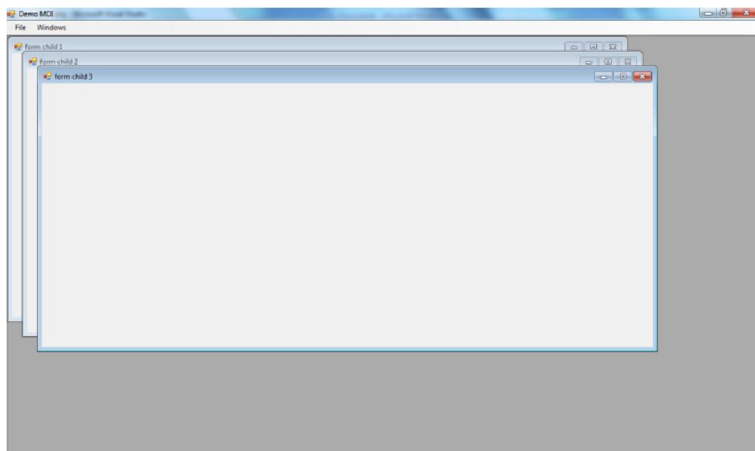
Thao tác trên MDI

29

- Sắp xếp các MDI
 - Để tạo 1 danh sách các MDI child, chỉ cần thêm 1 top-level menu item (thường mang tên Window).
 - Mỗi MDI container đều hỗ trợ hàm `LayoutMdi()` hàm này nhận giá trị từ Enumeration `MdiLayout` và sắp xếp tự động các cửa sổ.

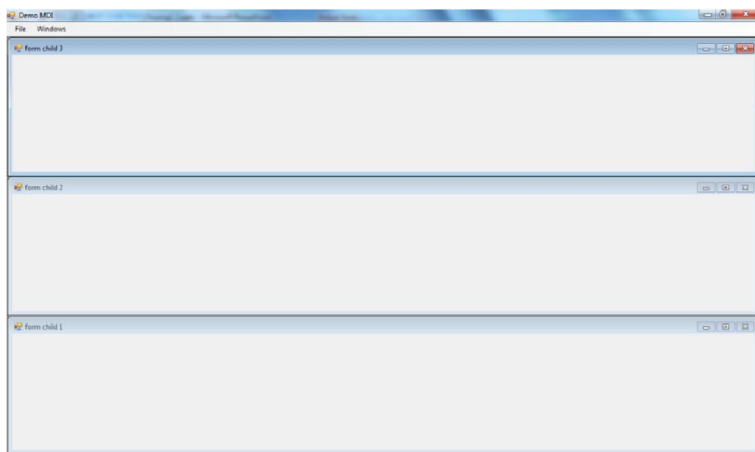
```
LayoutMdi(MdiLayout.);
```





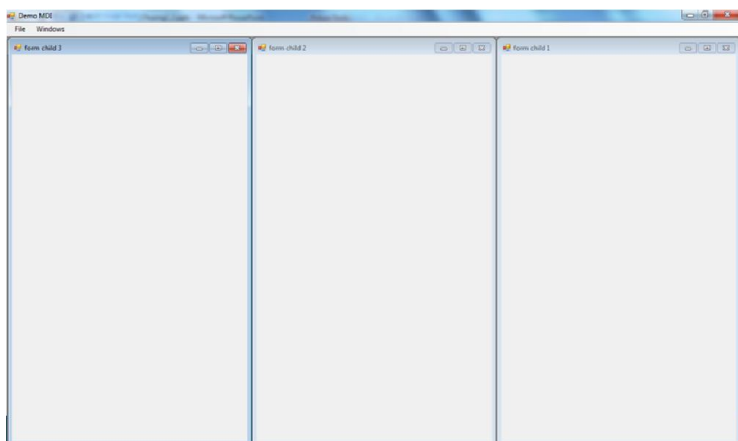
Kiểu sắp xếp: cascade

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



Kiểu sắp xếp: Tile Horizontal

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



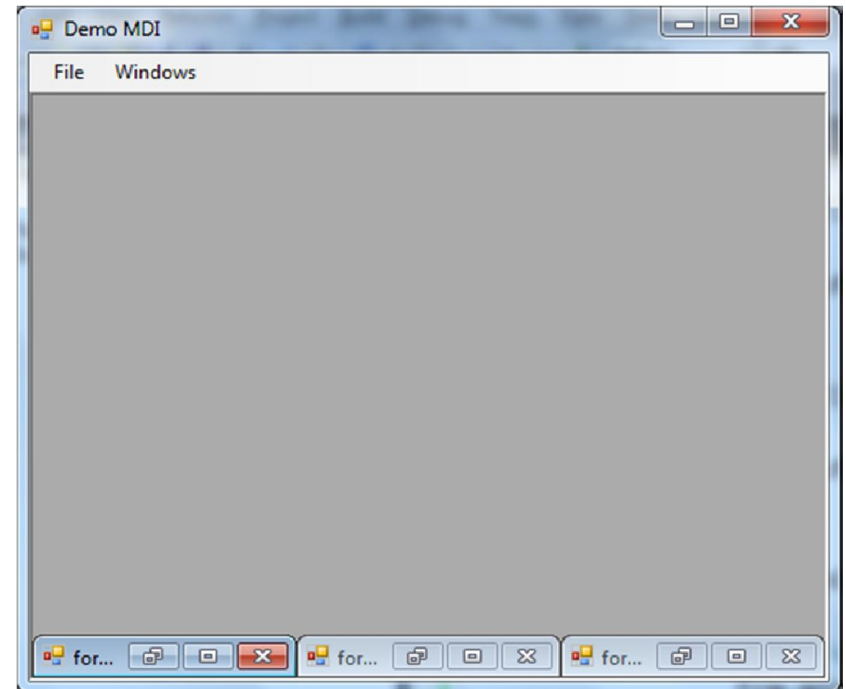
Kiểu sắp xếp: Tile Vertical

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

Thao tác trên MDI

- Duyệt qua từng MDI child. Sử dụng thuộc tính **MdiChildren** của MDI Parent.
- **Ví dụ:** Thu nhỏ các MDI child đang mở

Duyệt qua các MDI child



```
private void minimizeAllToolStripMenuItem_Click(object sender, EventArgs e)
{
    foreach (Form frm in this.MdiChildren)
        frm.WindowState = FormWindowState.Minimized;
}
```

Q&A