

Bài 6: Windows Form (tt)

Xử lý các sự kiện nhập liệu bàn phím, chuột

Lương Trần Hy Hiến

FIT, HCMUP

Lập trình Windows Form với C#

NỘI DUNG

- **Xử lý sự kiện bàn phím (Keyboard)**
- **Xử lý sự kiện chuột (Mouse)**

Giới thiệu

- **Tìm hiểu các 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 một khoảng thời gian cho trước
 - Khi ứng dụng hoạt động thì hệ thống sẽ truyền một tín hiệu cho ứng dụng theo từng khoảng thời gian định kỳ đã được khai báo.

Giới thiệu

- Bàn phím và chuột là 2 thiết bị nhập liệu quan trọng nhất của máy tính.
- Hầu hết các chức năng Windows đều hỗ trợ dùng bàn phím và chuột
- 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.

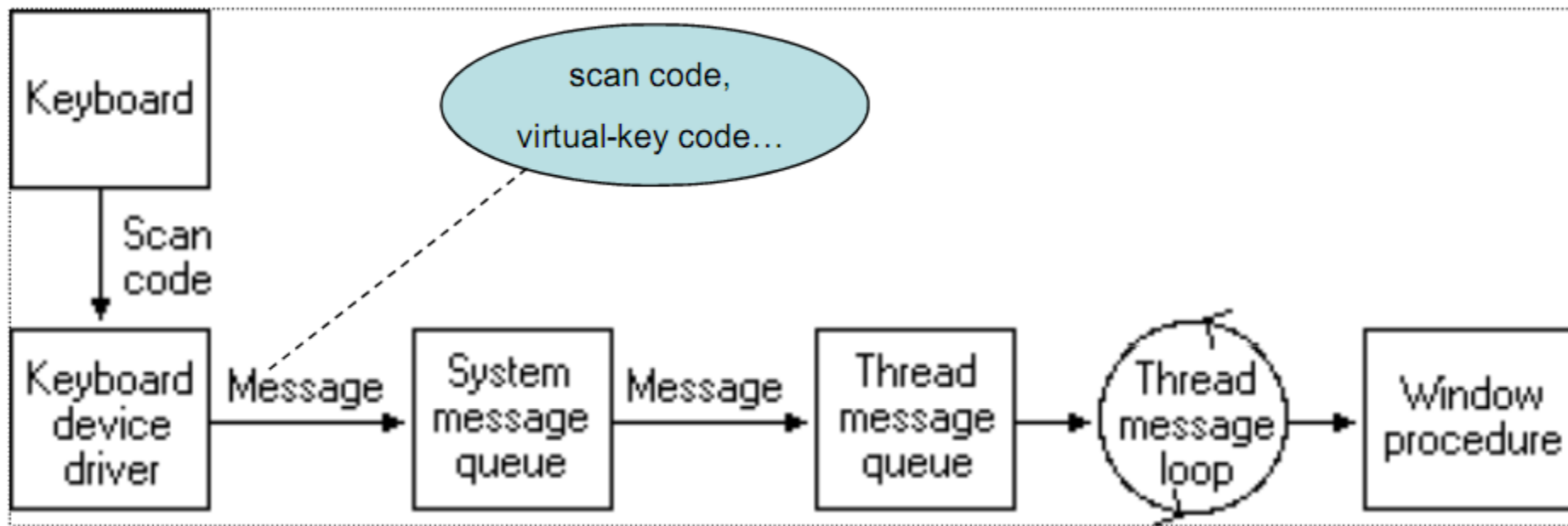
Bàn phím

- **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 hai 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.
- **Thường thì các phím hệ thống được Windows xử lý và dịch thành các sự kiện tương ứng.**

Xử lý sự kiện bàn phím

- Mô hình xử lý sự kiện bàn phím của Windows



Xử lý sự kiện bàn phím

- Khi người dùng nhấn hoặc nhả một phím bất kỳ từ bàn phím các driver bàn phím sẽ nhận được mã bàn phím và mã quét (scan code) của phím tương ứng.
- Mã quét này sẽ được chuyển thành mã phím ảo (Virtual keycode) và một thông điệp bàn phím tương ứng (bao gồm cả scan code, virtual keycode và một số thông tin khác) sẽ được gửi đến cho System message queue.
- Các sự kiện bàn phím chỉ được gửi đến cho cửa sổ đang giữ focus hiện hành.
- Hệ thống gửi hai sự kiện bàn phím khác nhau khi người dùng nhấn phím và nhả phím.

Xử lý sự kiện bàn phím

- **Các phím được nhấn được chia làm 4 nhóm sau:**
 - Toggle keys: Caps Lock, Num Lock, Scroll Lock
 - Shift keys: Shift, Ctrl, Alt
 - Noncharacter keys: các phím chức năng như các phím di chuyển, Pause, Delete
 - Character keys: các phím ký tự, phím số,...

Keyboard Event

- Phát sinh khi một phím được nhấn hoặc thả
- Có 3 sự kiện
 - KeyPress
 - KeyUp
 - KeyDown
- KeyPress phát sinh kèm theo với mã ASCII của phím được nhấn
- KeyPress không cho biết trạng thái các phím bổ sung {*Shift, Alt, Ctrl...*}
- Sử dụng KeyUp & KeyDown để xác định trạng thái các phím 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

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

Thuộc tính của lớp KeyEventArgs

Alt, Control, Shift Trạng thái các phím bổ sung

Handled Cho biết sự kiện đã xử lý

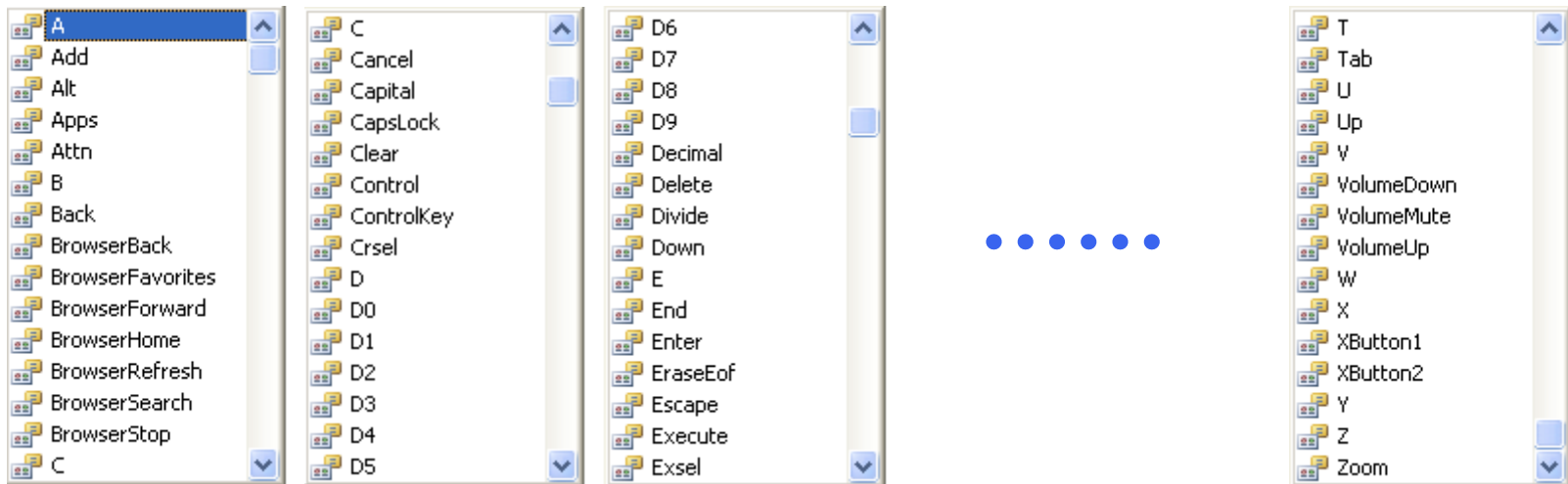
Keyboard Event

Thuộc tính của lớp EventArgs (tt)

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

Keyboard Event

■ Keys Enumeration



Key Enumeration

- Kiểu **Keys** được định nghĩa để liệt kê tất cả các phím. Bảng liệt kê giá trị 26 ký tự Latin được mô tả trong bảng sau:

<i>Keys Enumeration (letters)</i>			
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

Key Enumeration

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

Keyboard Event

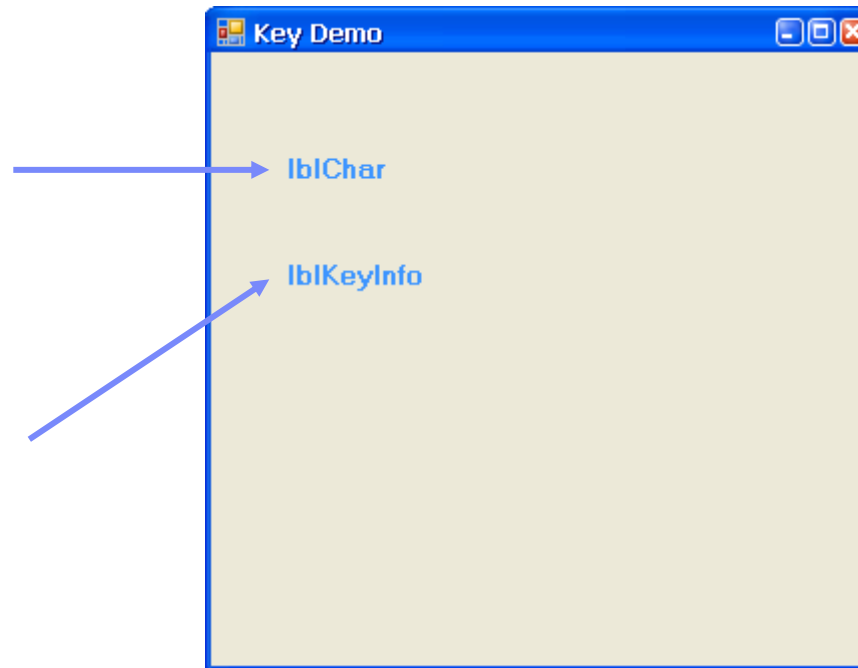
- **Minh họa các sự kiện: KeyPress, KeyDown, KeyUp**
 - Khi user nhấn một phím
 - Bắt sự kiện KeyPress: xuất ra phím được nhấn
 - Bắt sự kiện KeyDown: xuất ra các tham số trong KeyEventArgs
 - Khi user thả phím
 - Xóa các thông tin mô tả phím được nhấn trong các label
- **Cách thực hiện**
 - Tạo một form minh họa
 - Thiết kế trên form có 2 Label:
 - lblChar: hiển thị ký tự được nhấn trong KeyPress
 - lblKeyInfo: hiển thị các thông tin của KeyEventArgs khi KeyDown

Keyboard Event

■ Bước 1: tạo Windows Form như hình mô tả

Label chứa ký tự
được nhấn trong sự
kiện KeyPress

Label chứa thông tin
mã ký tự được nhấn
trong sự kiện
KeyDown



Keyboard Event

■ Bước 2:

– Tạo KeyPress Event Handling cho form

The screenshot illustrates the process of adding a KeyPress event handler to a Windows Form. On the left, a form titled 'Key Demo' contains two labels: 'lblChar' and 'lblKeyInfo'. A yellow lightning bolt icon points to the 'KeyPress' event in the 'Properties' window. The 'Properties' window shows the 'Form1' object with a list of events. 'KeyPress' is selected, and the handler 'Form1_KeyPress' is assigned. A blue arrow points from the 'Form1_KeyPress' entry in the Properties window to a code block on the right. The code block contains the following C# code:

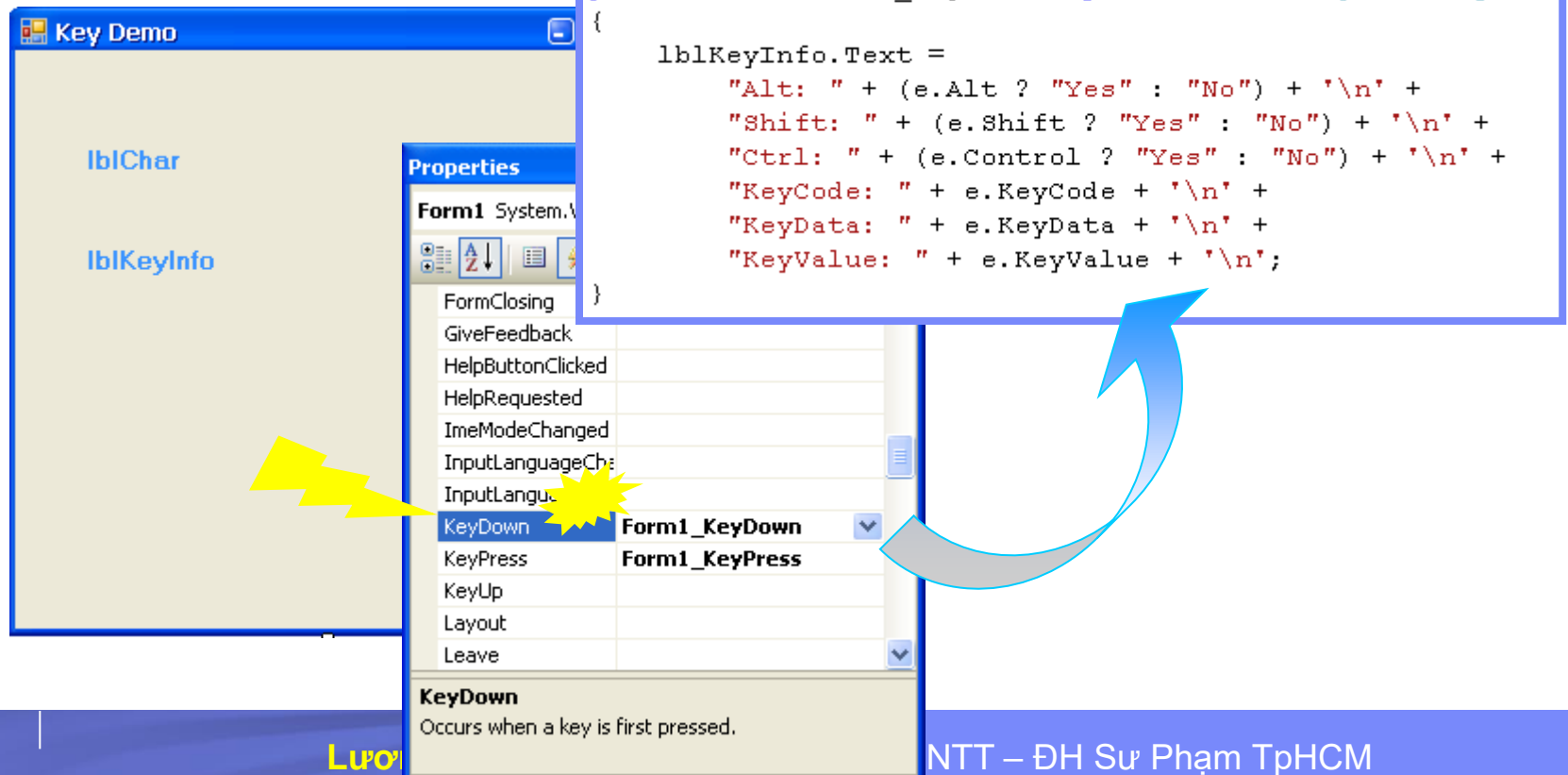
```
private void Form1_KeyPress(object sender,
    KeyPressEventArgs e)
{
    // hiển thị ký tự trên label
    lblChar.Text = "Key pressed: " + e.KeyChar;
}
```

Below the event list, the 'KeyPress' event is described: 'Occurs when the control has focus and the user presses and releases a key.'

Keyboard Event

■ Bước 3:

– Tạo KeyDown Event Handling cho form

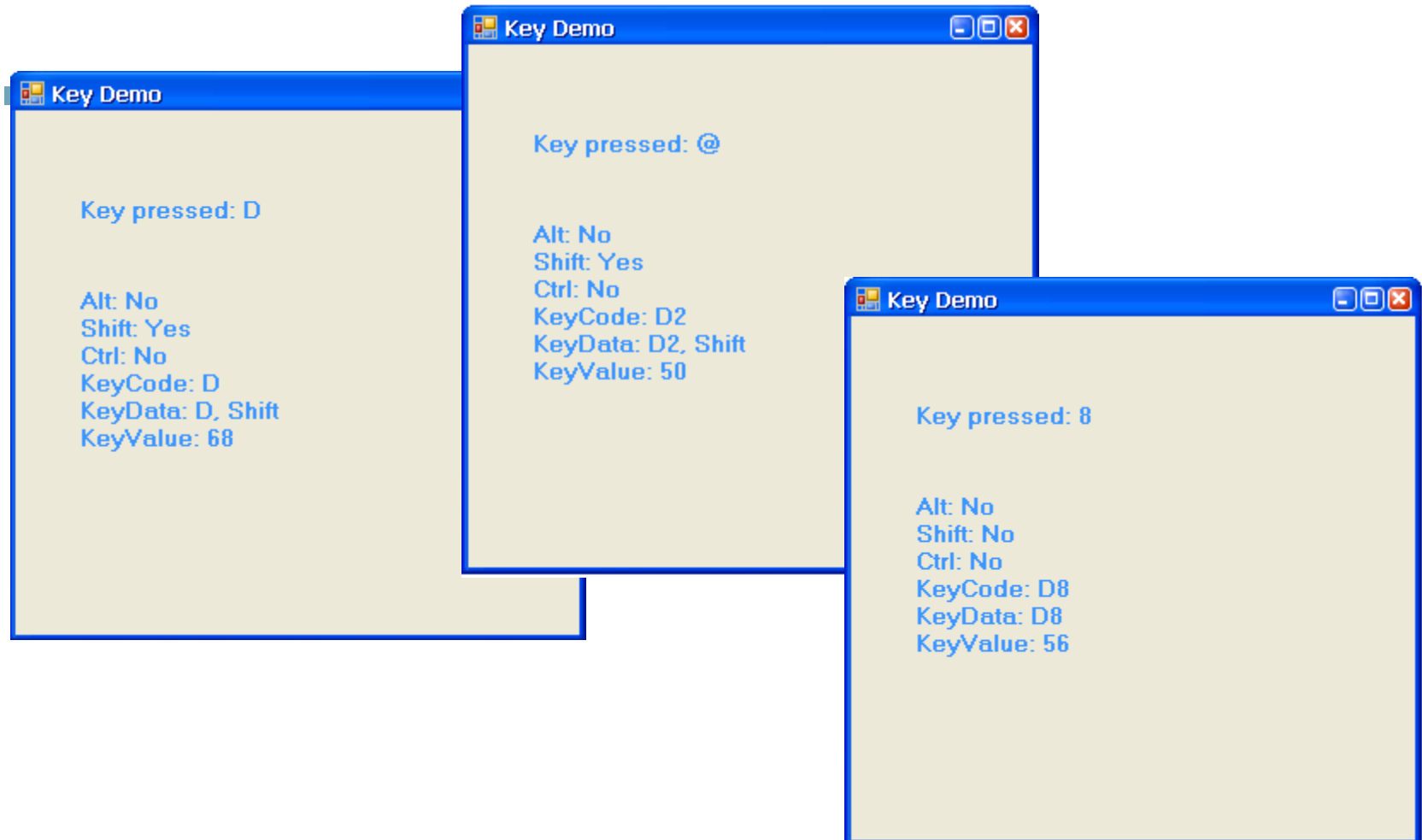


The screenshot shows a Windows Forms application titled "Key Demo". The form contains two labels: "lblChar" and "lblKeyInfo". The Properties window is open, showing the "Form1" properties. The "KeyDown" event is selected in the "Events" tab, and the "Form1_KeyDown" handler is assigned. A yellow lightning bolt points to the "KeyDown" event in the Properties window. A blue arrow points from the "KeyDown" event to a code block showing the event handler implementation.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    lblKeyInfo.Text =
        "Alt: " + (e.Alt ? "Yes" : "No") + '\n' +
        "Shift: " + (e.Shift ? "Yes" : "No") + '\n' +
        "Ctrl: " + (e.Control ? "Yes" : "No") + '\n' +
        "KeyCode: " + e.KeyCode + '\n' +
        "KeyData: " + e.KeyData + '\n' +
        "KeyValue: " + e.KeyValue + '\n';
}
```

KeyDown
Occurs when a key is first pressed.

Keyboard Event



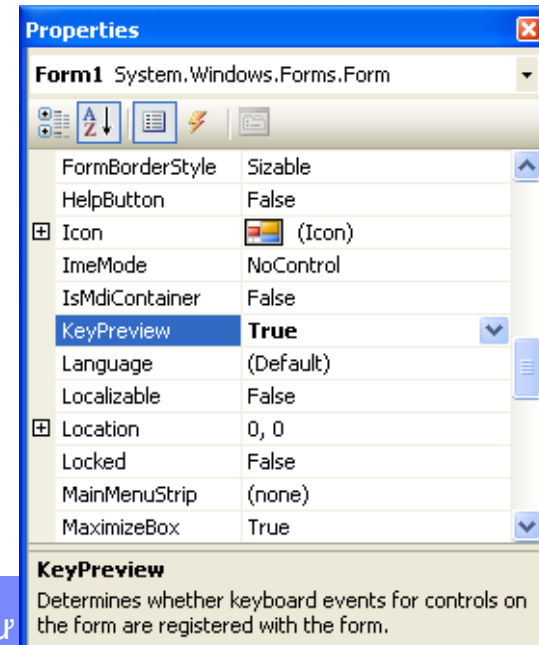
Keyboard Event

■ Mở rộng chương trình Calculator mở rộng cho phép xử lý các phím

- Form nhận xử lý thông điệp KeyDown
 - Xác định các phím tương ứng rồi gọi sự kiện click của button
 - VD: user gõ phím 1, tương tự như button “1” được nhấn

■ Cách thực hiện

- Khai báo trình xử lý sự kiện KeyDown cho Form chính
- Thiết lập thuộc tính *KeyPreview* cho Form để nhận sự kiện bàn phím.



Keyboard Event

- **Viết phần xử lý cho sự kiện KeyDown**
 - Xác định các phím tương ứng để gọi sự kiện click của các button.

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    switch (e.KeyCode)
    {
        case Keys.D1:
            btnNum1.PerformClick();
            break;
        case Keys.D2:
            btnNum2.PerformClick();
            break;
        //.....
        case Keys.Oemplus:
            if (!e.Shift) // phím '='
                btnEqual.PerformClick();
            else // phím '+'
                btnPlus.PerformClick();
            break;
        // còn tiếp cho các phím khác.....
    }
}
```

Gọi event Click
của button “1”

Phím '=' được nhấn

Phím '+' được nhấn

Mouse Event

Lập trình Windows Form với C#

Giới thiệu

- Về cơ bản Windows hỗ trợ các loại thiết bị chuột có một nút, hai và ba nút, ngoài ra Windows còn có thể dùng thiết bị khác như joystick hay bút vẽ để bắt chước thiết bị chuột.
- Các thông điệp được tạo từ chuột rất khác với thông điệp của bàn phím:
 - Chuột di chuyển qua cửa sổ
 - hay kích vào trong cửa sổ,
 - Thậm chí cả trong trường hợp cửa sổ không được kích hoạt hay không nhận được sự quan tâm.

Giới thiệu

- **Các sự kiện chuột sẽ được gửi đến cho:**
 - Cửa sổ hiện đang chứa con trỏ chuột.
 - Hoặc cửa sổ đang “capture” chuột.
- **Có hai loại sự kiện về chuột:**
 - **Client area messages:** Các sự kiện chuột xảy ra khi chuột đang ở vùng client của cửa sổ.
 - **Non-client area messages:** 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.

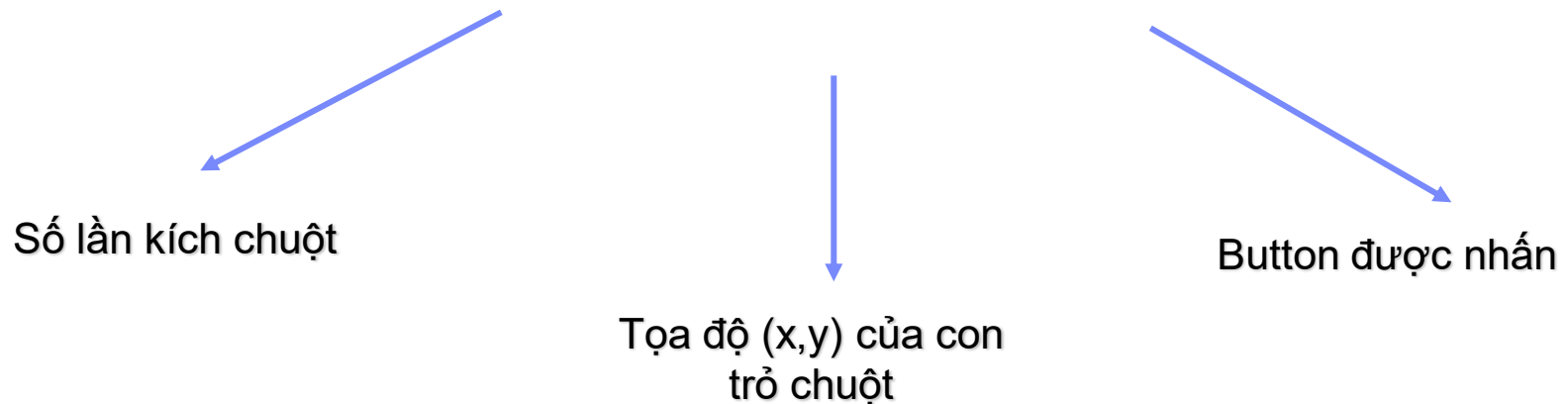
Mouse Event

- **Một số các thao tác phát sinh từ mouse**
 - Di chuyển
 - Kích chuột
- **Ứng dụng cần xử lý sự kiện chuột nào sẽ khai báo trình xử lý tương ứng**
- **Lớp `MouseEventArgs` được sử dụng để chứa thông tin truyền vào cho trình xử lý sự kiện mouse.**
- **Mỗi trình xử lý sự kiện sẽ có tham số là đối tượng object và đối tượng `MouseEventArgs` (hoặc `EventArgs`)**

Mouse Event

- Tham số cho sự kiện liên quan đến mouse

MouseEventArgs



Mouse Event

Sự kiện chuột với tham số kiểu EventArgs

MouseEnter	Xuất hiện khi con trỏ chuột đi vào vùng biên của control
MouseLeave	Xuất hiện khi con trỏ chuột rời khỏi biên của control

Sự kiện chuột với tham số kiểu MouseEventArgs

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

Mouse Event

Thuộc tính của lớp MouseEventArgs

Button	Button được nhấn {Left, Right, Middle, none} có kiểu là 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 Test Mouse Button

```
protected override void OnMouseClicked(MouseEventArgs  
mea)  
{  
    base.OnMouseClicked(mea);  
    if (mea.Button == MouseButtons.Left)  
        MessageBox.Show("Nhan chuot trai");  
    if (mea.Button == MouseButtons.Right)  
        MessageBox.Show("Nhan chuot phai");  
    if (mea.Button == MouseButtons.Middle)  
        MessageBox.Show("Nhan chuot giua");  
}
```

Sự kiện MouseDown

- Sự kiện MouseDown được phát sinh khi người dùng nhấn một nút của chuột.
- Để xử lý sự kiện MouseDown ta override phương thức OnMouseDown
- Ví dụ:

```
protected override void OnMouseDown(MouseEventArgs mea)  
{  
    MessageBox.Show("Ban vua nhan chuot" + mea.Button);  
}
```

Sự kiện MouseUp

- Sự kiện MouseUp được phát sinh khi người dùng nhả một nút của chuột.
- Để xử lý sự kiện MouseUp ta override phương thức OnMouseUp
- Ví dụ:

```
protected override void OnMouseUp(MouseEventArgs mea)  
{  
    MessageBox.Show("Ban vua nha chuot " + mea.Button);  
}
```


Sự kiện MouseEventArgs

- Sự kiện MouseEventArgs được phát sinh khi người dùng di chuyển chuột.
- Để xử lý sự kiện MouseEventArgs ta override phương thức OnMouseEventArgs
- Ví dụ:

```
protected override void OnMouseEventArgs(MouseEventArgs e)  
{  
    //Vẽ một đường thẳng từ toạ độ (0,0) đến toạ độ chuột di chuyển  
    Graphics g = CreateGraphics();  
    Pen pen = new Pen(System.Drawing.Color.Blue);  
    g.DrawLine(pen, 0, 0, mea.X, mea.Y);  
}
```

Sự kiện MouseWheel

- Sự kiện MouseWheel được phát sinh khi người dùng scroll chuột scroll chuột.
- Để xử lý sự kiện MouseWheel ta override phương thức OnMouseWheel
- Ví dụ:

```
protected override void OnMouseWheel(MouseEventArgs e)  
{  
    if (mea.Delta>0)  
        MessageBox.Show("Ban vua scroll chuot len",  
        "Thong bao");  
    else  
        MessageBox.Show("Ban vua scroll chuot xuong",  
        "Thong bao");  
}
```

Sự kiện Click

- Sự kiện Click phát sinh khi một phím bất kỳ của chuột được nhấn.
- Sự kiện này phát sinh kèm theo tham số EventArgs, tham số này không chứa thông tin về trạng thái của nút chuột được nhấn cũng như vị trí của con trỏ chuột khi nhấn.

```
protected override void OnClick(EventArgs ea)  
{  
  
...  
  
}
```

Sự kiện DoubleClick

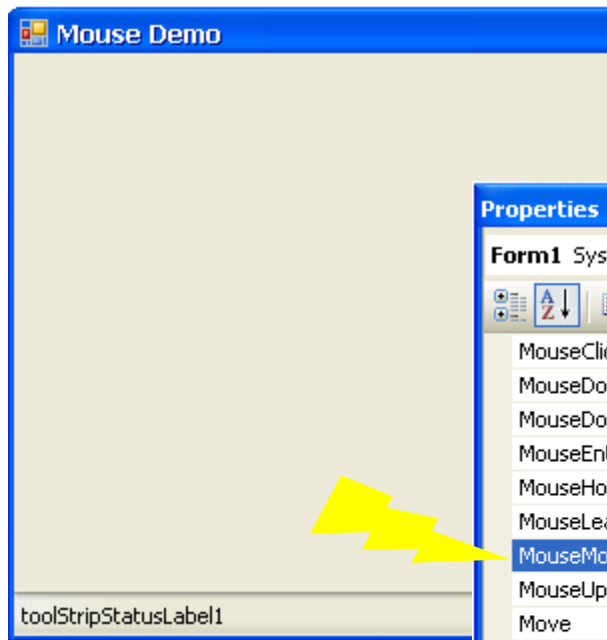
Khi DoubleClick sẽ phát sinh một dãy các sự kiện sau:

- MouseDown
- Click
- MouseUp
- MouseMove
- MouseDown
- DoubleClick
- MouseUp
- MouseMove

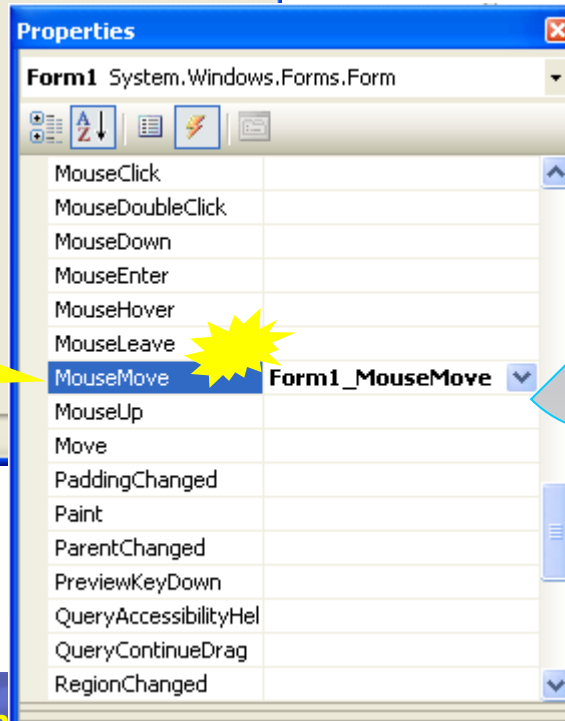
```
protected override void OnDoubleClick(EventArgs ea)  
{  
...  
}
```

Mouse Event

■ MouseMove

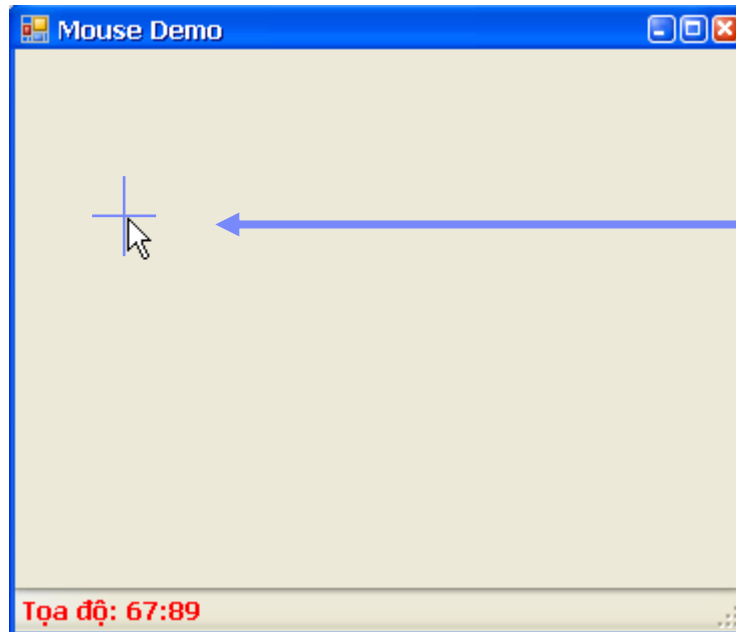


```
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    // hiển thị tọa độ con trỏ trên Status Bar
    string s;
    s = "Tọa độ: " + e.X.ToString() + ":" + e.Y.ToString();
    toolStripStatusLabel1.Text = s;
}
```



Mouse Event

■ Demo



Vị trí hiện tại của con
trỏ chuột

Hiển thị tọa độ hiện tại của
con trỏ chuột

Mouse Event

- **Demo thao tác: kích chuột trái tại một điểm A, giữ chuột trái và di chuyển chuột, chương trình sẽ vẽ đường thẳng từ điểm A đến vị trí hiện tại chuột.**
- **Các sự kiện cần xử lý**
 - MouseDown:
 - Xác định điểm A ban đầu
 - MouseMove
 - Kiểm tra nếu Left button của chuột đang giữ
 - Sử dụng Graphics để vẽ đường thẳng từ A đến vị trí hiện tại

Mouse Event

■ Bước 1:

- Tạo biến lưu trữ điểm A khi user kích chuột trái
- Biến pA có kiểu Point là biến thành viên của Form1

Lớp Form1

```
public partial class Form1 : Form
{
    private Point pA; // biến lưu giữ tọa độ A

    public Form1()
    {
        InitializeComponent();
    }
}
```

Biến pA lưu giữ tọa độ khi
chuột trái được click

Mouse Event

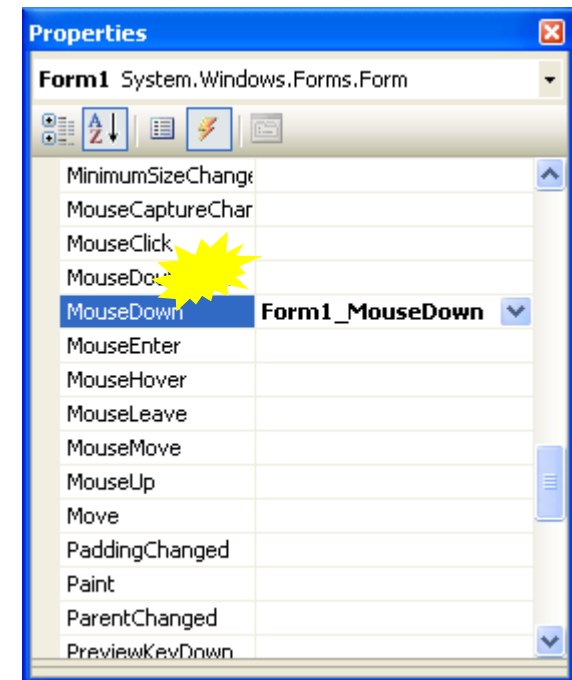
■ Bước 2

– Khai báo xử lý sự kiện MouseDown trong Form1

- Trong cửa sổ event của Form1, kích đúp vào sự kiện MouseDown

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    // lưu lại điểm thứ 1
    pA = e.Location;
}
```

Lưu lại điểm được nhấn chuột



Mouse Event

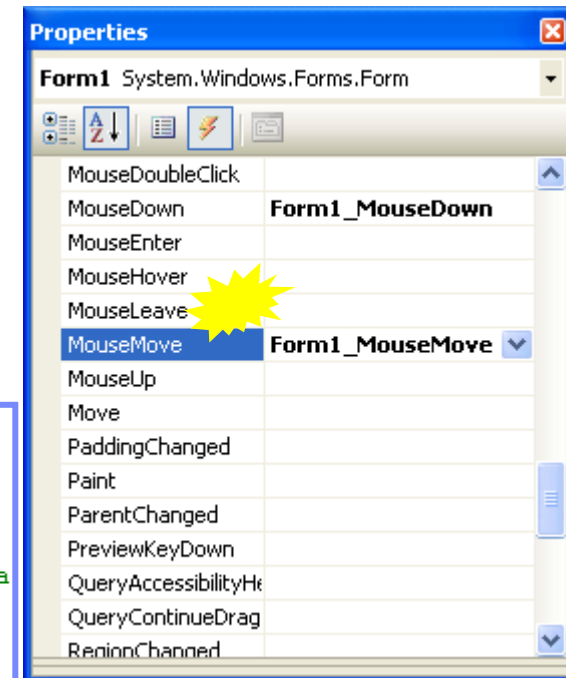
■ Bước 3

– Cài đặt xử lý sự kiện MouseMove

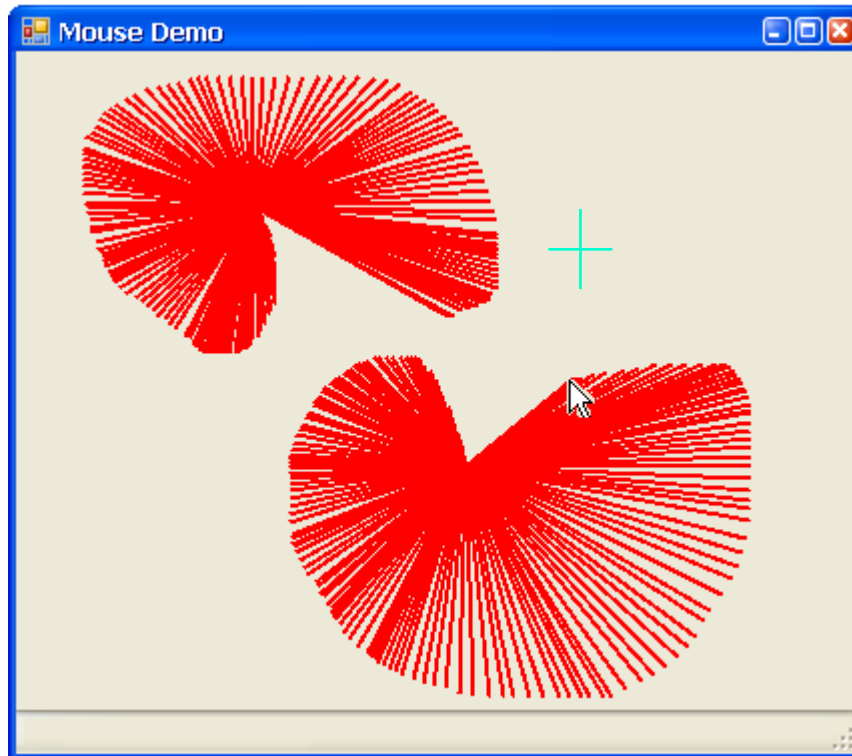
- Kiểm tra nếu LeftButton được nhấn
 - Vẽ đường thẳng từ pA đến vị trí hiện tại

```
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left) // LeftButton được nhấn
    {
        Graphics g = this.CreateGraphics(); // lấy thiết bị đồ họa
        Pen pen = new Pen(Color.Red, 2f); // tạo bút vẽ

        g.DrawLine(pen, pA, e.Location); // vẽ đường thẳng
    }
}
```



Mouse Event



Bài tập

- Chương trình đồ họa đơn giản: Vẽ đường thẳng, hình chữ nhật, ellipses

