



Windows Controls

GV: BÙI NGỌC LÊ

Nội Dung

- **Mouse Event handling**
- **Keyboard event handling**

Mouse Event

Mouse Event

- Mouse là thiết bị tương tác thông dụng trên GUI
- 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

Số lần kích chuột

Tọa độ (x,y) của
con trỏ chuột

Button được nhấn

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

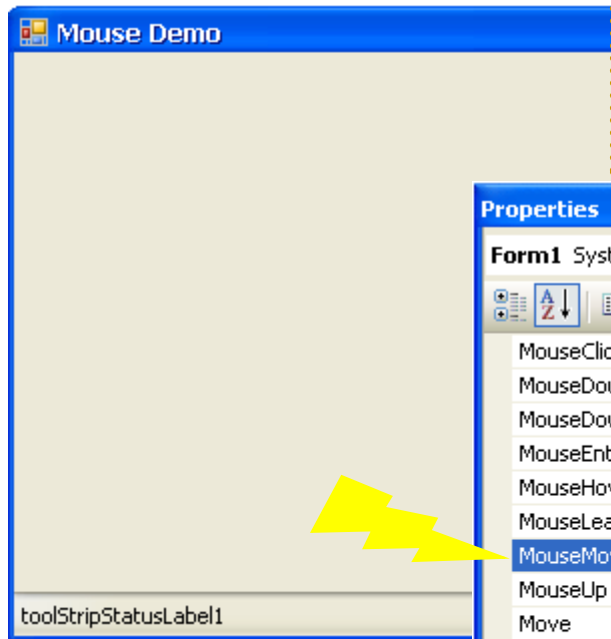
Mouse Event

Thuộc tính của lớp MouseEventArgs

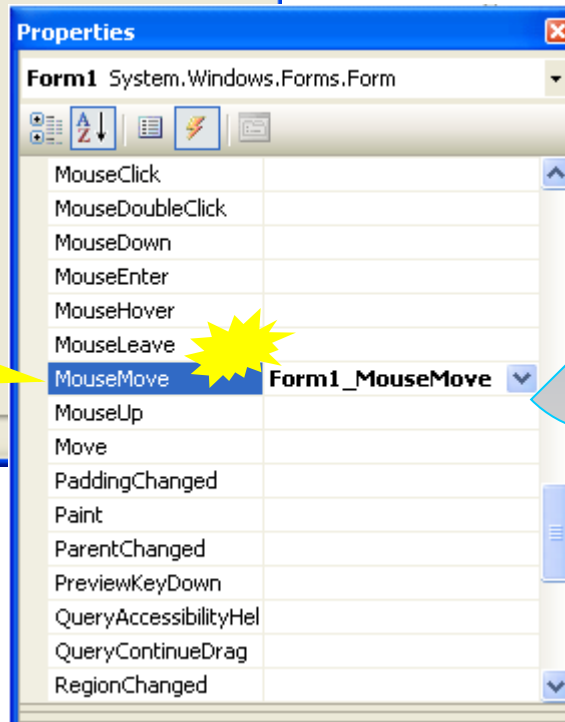
Button	Button được nhấn {Left, Right, Middle, none} có kiểu là MouseButton
Clicks	Số lần button được nhấn
X	Tọa độ x của con trỏ chuột trong control
Y	Tọa độ y của con trỏ chuột trong control

Mouse Event

■ MouseMove

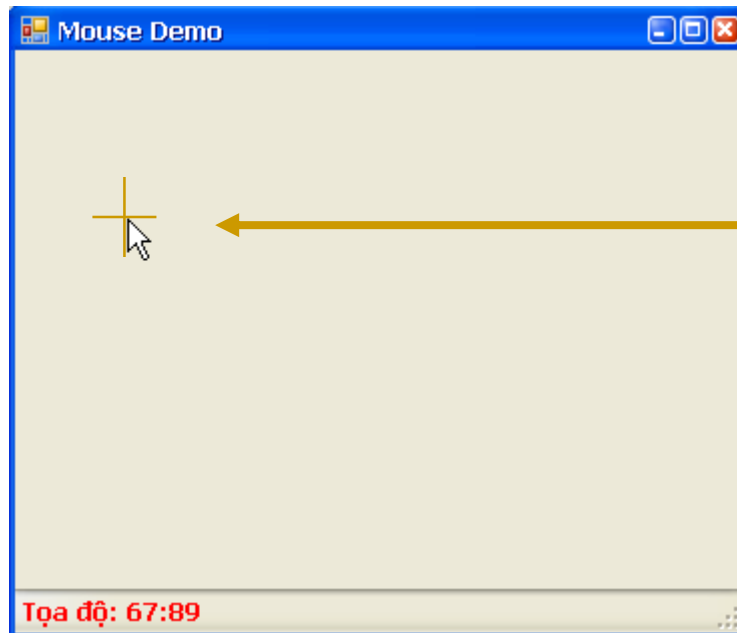


```
private void Form1_MouseMove(object sender, MouseEventArgs e)
{
    // hiển thị tọa độ con trỏ trên Status Bar
    string s;
    s = "Tọa độ: " + e.X.ToString() + ":" + e.Y.ToString();
    toolStripStatusLabel1.Text = s;
}
```



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

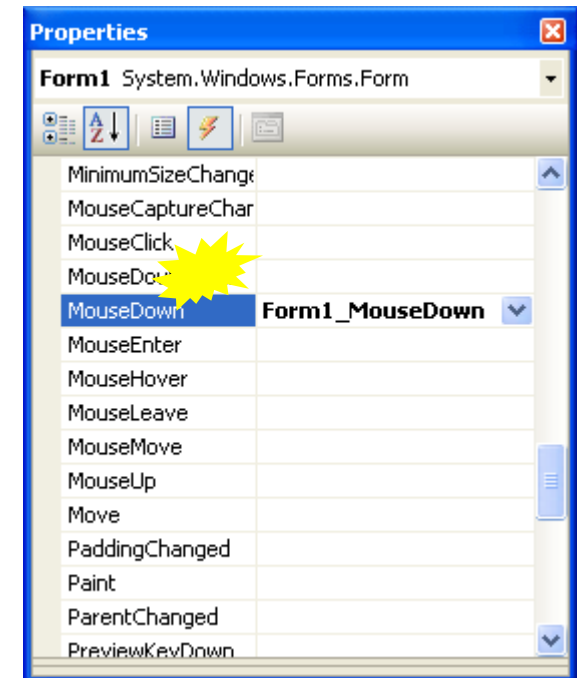
Mouse Event

■ 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



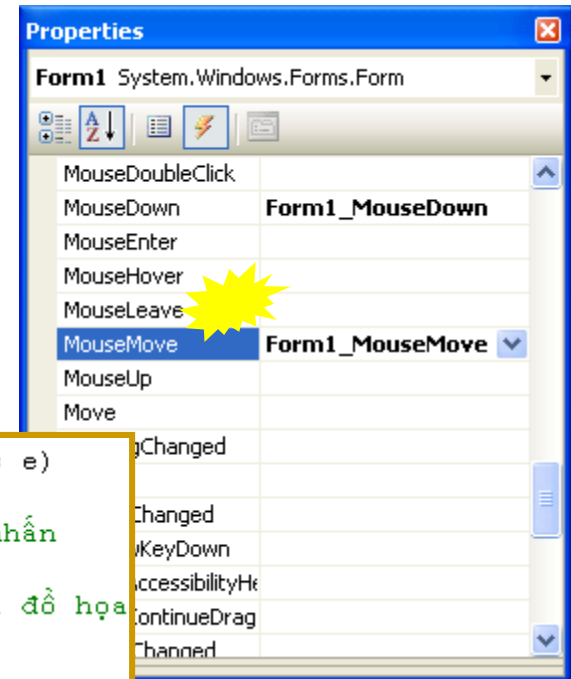
Mouse Event

■ 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
    }
}
```



Keyboard Event

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

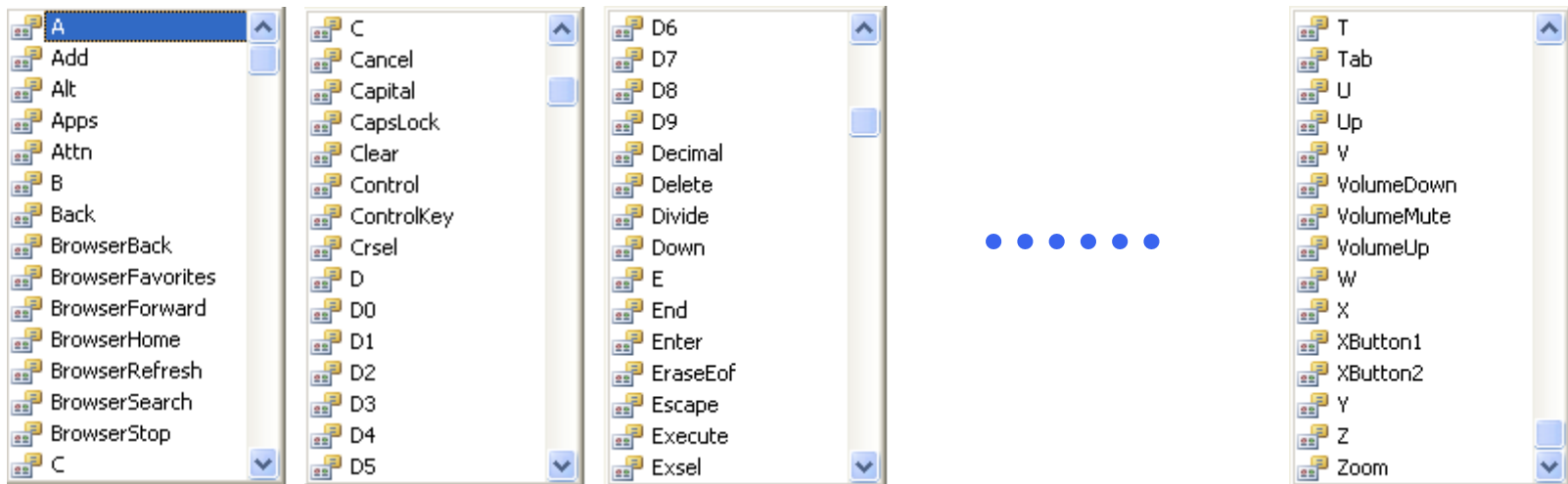
Keyboard Event

Thuộc tính của lớp KeyEventArgs (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



Keyboard Event

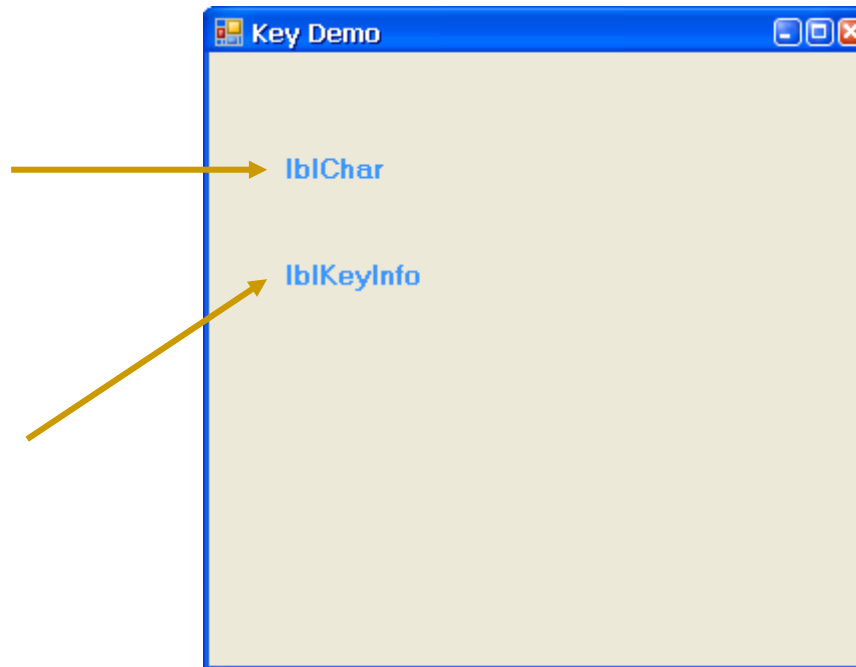
- **Minh họa các sự kiện: KeyPress, KeyDown, KeyUp**
 - **Khi user nhấn một phím**
 - **Bắt sự kiện KeyPress:** xuất ra phím được nhấn
 - **Bắt sự kiện KeyDown:** xuất ra các tham số trong KeyEventArgs
 - **Khi user thả phím**
 - **Xóa các thông tin mô tả phím được nhấn trong các label**
- **Cách thực 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**

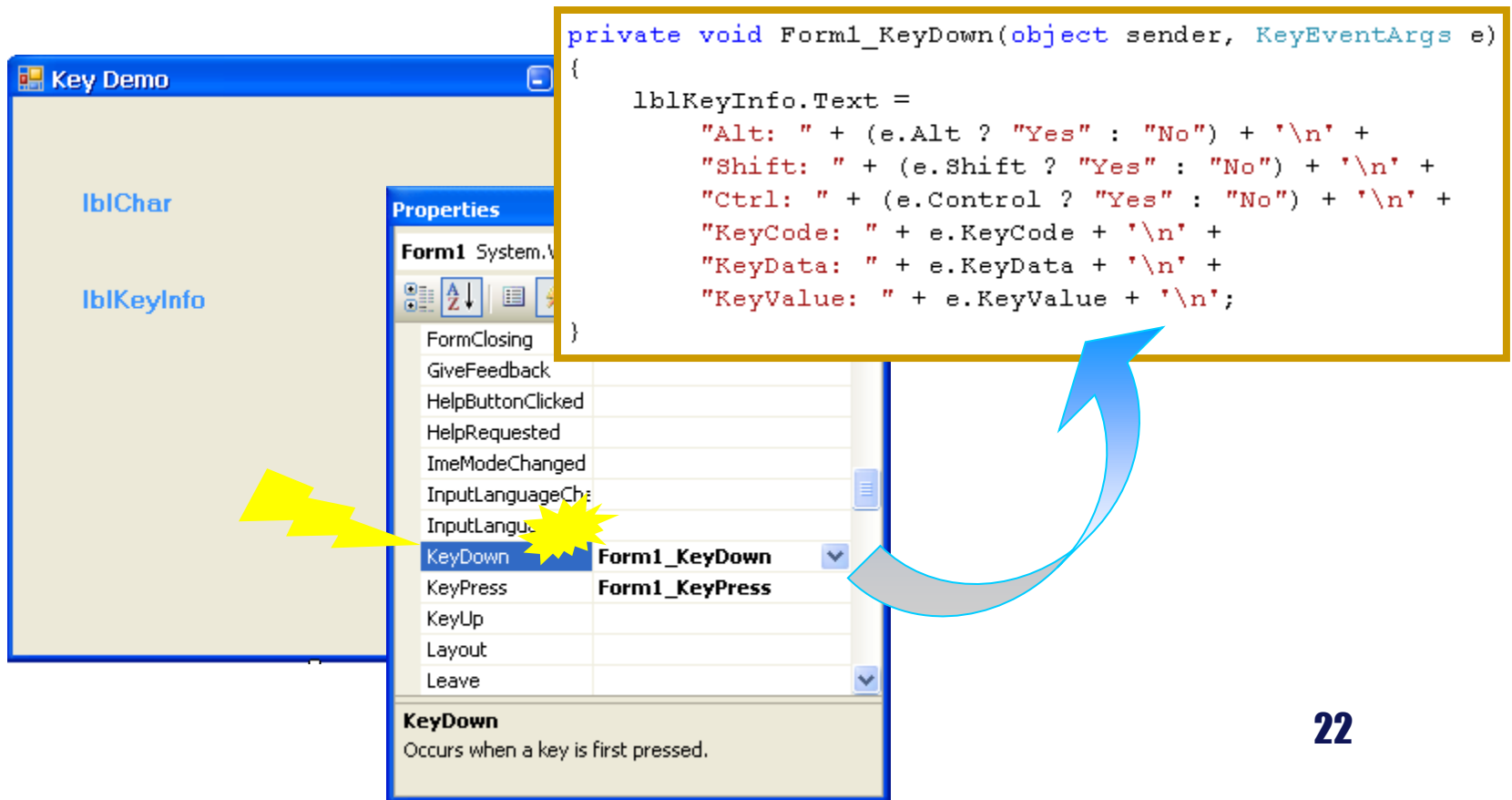
The screenshot illustrates the process of creating a KeyPress event handler in a Windows Forms application. It shows a form titled "Key Demo" with two labels, "lblChar" and "lblKeyInfo". The Properties window for "Form1" is open, and the "KeyPress" event is selected, with the handler "Form1_KeyPress" assigned. A yellow lightning bolt icon indicates the event assignment. A blue arrow points from the event handler name in the Properties window to a code block showing the implementation of the "Form1_KeyPress" method.

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

KeyPress
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 Visual Studio window titled "Key Demo" with two labels: "lblChar" and "lblKeyInfo". The Properties window is open, showing the "Form1" tab. The "KeyDown" event is selected in the event list, and the "Form1_KeyDown" handler is assigned. A yellow lightning bolt points to the event list, and a blue arrow points from the event to the code block.

```
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';
}
```

Keyboard Event

■ Demo

The image displays three overlapping windows, each titled 'Key Demo', illustrating keyboard event data. The windows are arranged in a staggered fashion, with the 'Key pressed: D' window at the bottom-left, the 'Key pressed: @' window in the middle, and the 'Key pressed: 8' window at the bottom-right.

Key Demo (Leftmost window):

Key pressed: D

Alt: No
Shift: Yes
Ctrl: No
KeyCode: D
KeyData: D, Shift
KeyValue: 68

Key Demo (Middle window):

Key pressed: @

Alt: No
Shift: Yes
Ctrl: No
KeyCode: D2
KeyData: D2, Shift
KeyValue: 50

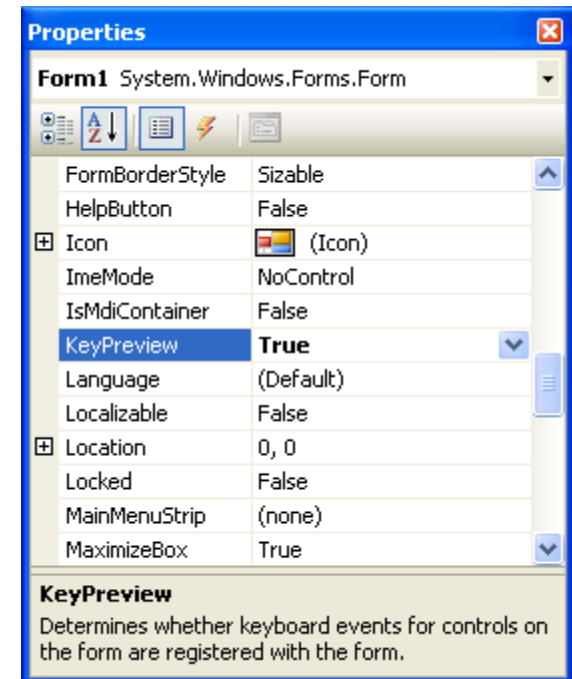
Key Demo (Rightmost window):

Key pressed: 8

Alt: No
Shift: No
Ctrl: No
KeyCode: D8
KeyData: D8
KeyValue: 56

Keyboard Event

- CT Calculator (BT3) 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

Tóm tắt

- **Thiết kế layout trên form**
 - Anchor
 - Dock các control
- **Các control trên form**
 - Control nhập liệu
 - Control chọn giá trị
 - Container control
 - Component
 - Advanced control
- **Mouse event**
- **Keyboard event**