



# BÀI GIẢNG

# LẬP TRÌNH GIAO DIỆN

## CHƯƠNG 5:

## SỰ KIỆN BÀN PHÍM VÀ CHUỘT

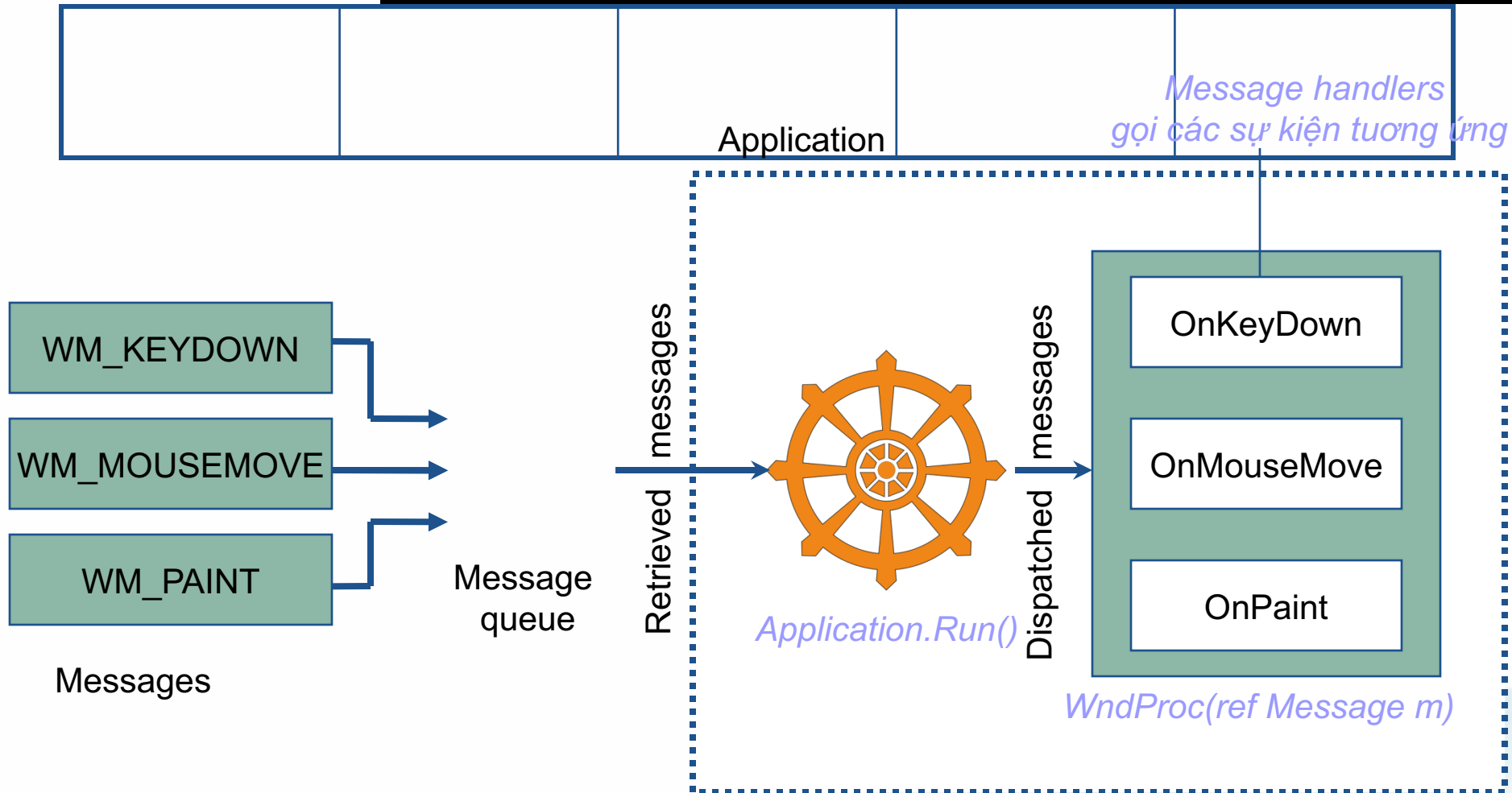


1. Sự kiện bàn phím.
2. Sự kiện chuột.





# Sự kiện trong C#





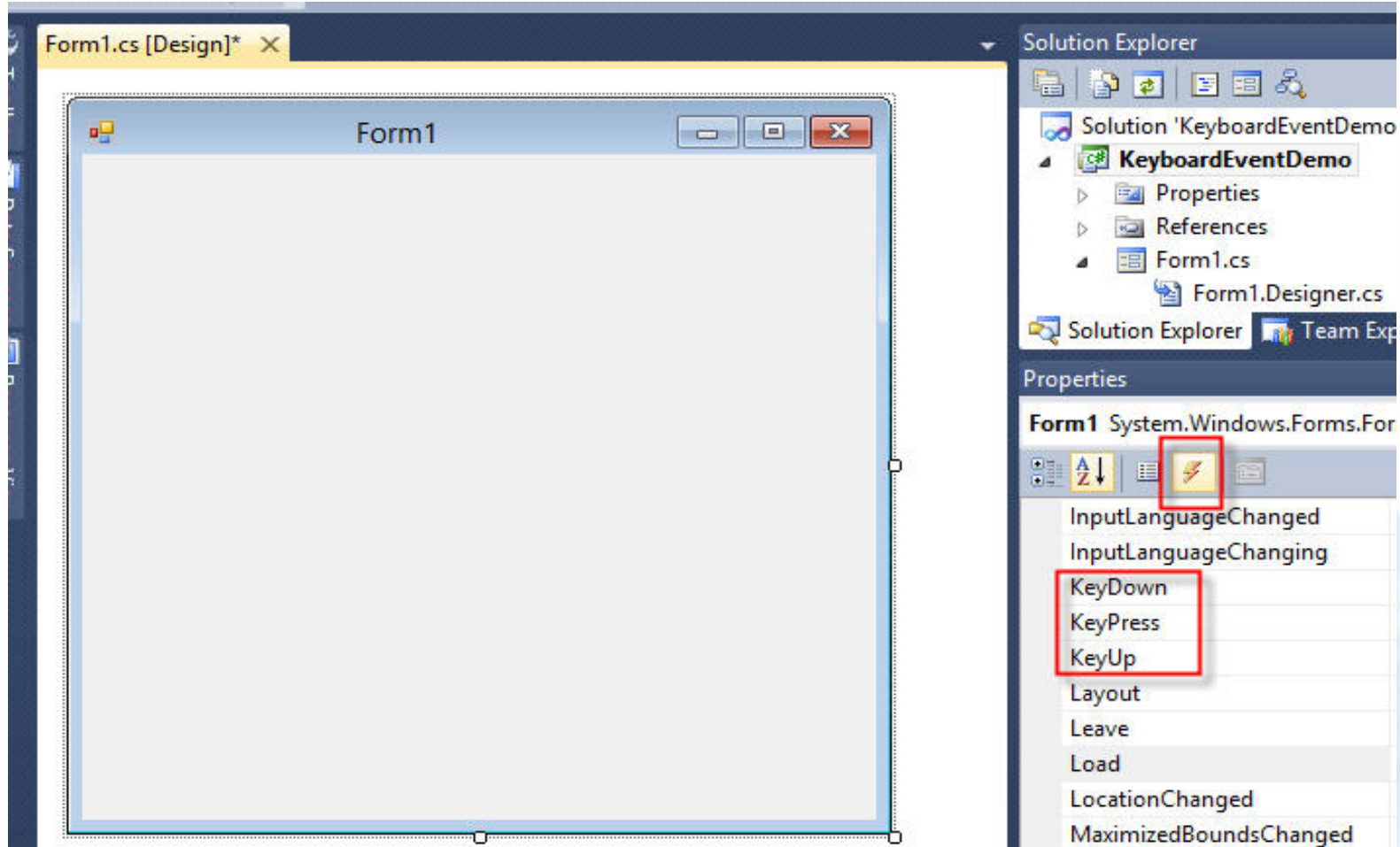
- Ứng dụng phản ứng các sự kiện (nhấn phím, click chuột, ...) bằng cách xử lý các message do Windows gửi đến
- Một ứng dụng Windows điển hình thực hiện một lượng lớn các xử lý để phản hồi các message nó nhận. Và giữa các message nó chờ message kế tiếp đến



# Sự kiện bàn phím (Keyboard)

- Khi người dùng nhấn hoặc thả một phím trên cửa sổ chương trình, chương trình sẽ nhận được các sự kiện, ứng dụng sẽ gọi các hàm xử lý tương ứng với các sự kiện vừa phát sinh.
- Trong một ứng dụng Windows Form, các sự kiện về phím bao gồm:
  - KeyPress: xảy ra khi phím được nhấn, sau sự kiện KeyDown và trước sự kiện KeyUp
  - KeyDown: xảy ra khi một phím được nhấn
  - KeyUp: xảy ra khi một phím được thả

# Sự kiện bàn phím (Keyboard)





# Sự kiện bàn phím

## ■ KeyPress:

- Được phát sinh khi người dùng nhấn một phím bất kỳ.
- Hàm xử lý nhận đối số là đối tượng của lớp **KeyPressEventArgs** chứa **mã ASCII của phím** được nhấn.
- **KeyPressEventArgs** không chứa thông tin về trạng thái các phím chức năng như: Shift, Alt , Ctrl...

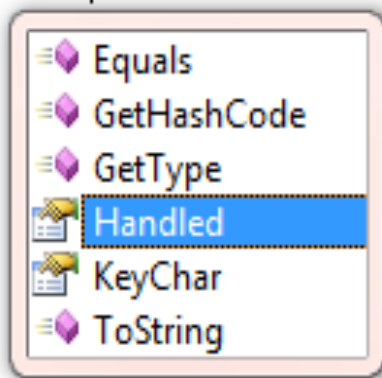




# Sự kiện KeyPress (tt)

- Hai thuộc tính của lớp `KeyPressEventArgs`:
  - `KeyChar`: mã ASCII của ký tự được nhấn
  - `Handled`: `True/False`, cho biết sự kiện

```
private void KeyboardAnMouse_KeyPress(object sender, KeyPressEventArgs e)
{
    e.|
}
```



`bool KeyPressEventArgs.Handled`  
Gets or sets a value indicating whether the `System.Windows.Forms.Control.KeyPress`





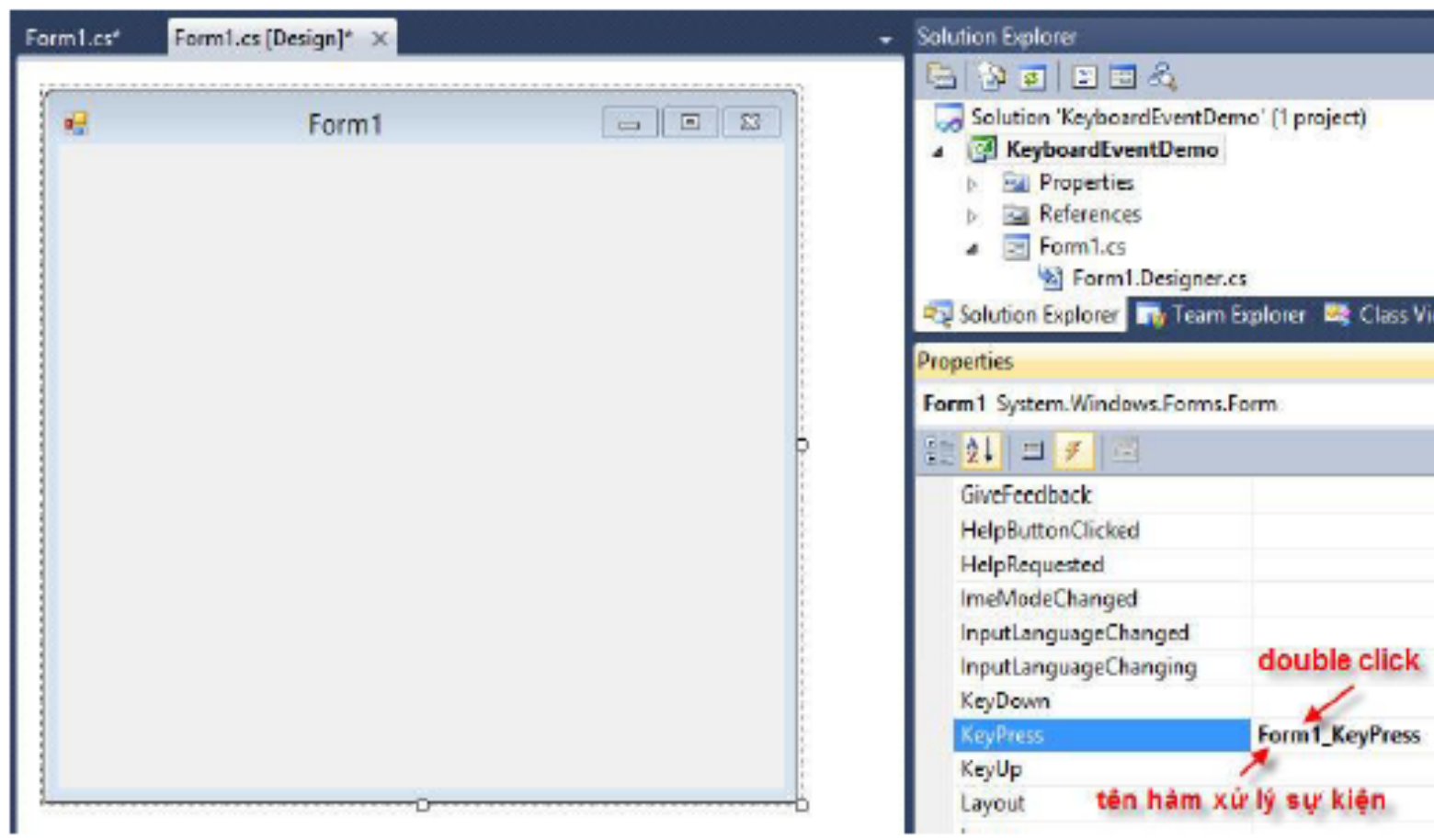
# Sự kiện bàn phím

- Xử lý sự kiện KeyPress trên form:
  - Khai báo hàm xử lý sự kiện KeyPress
  - Thiết lập thuộc tính KeyPreview = True
  - Trong tab Event, double click sự kiện KeyPress
- Xử lý sự kiện KeyPress trên control: tương tự trên form





# Sự kiện bàn phím





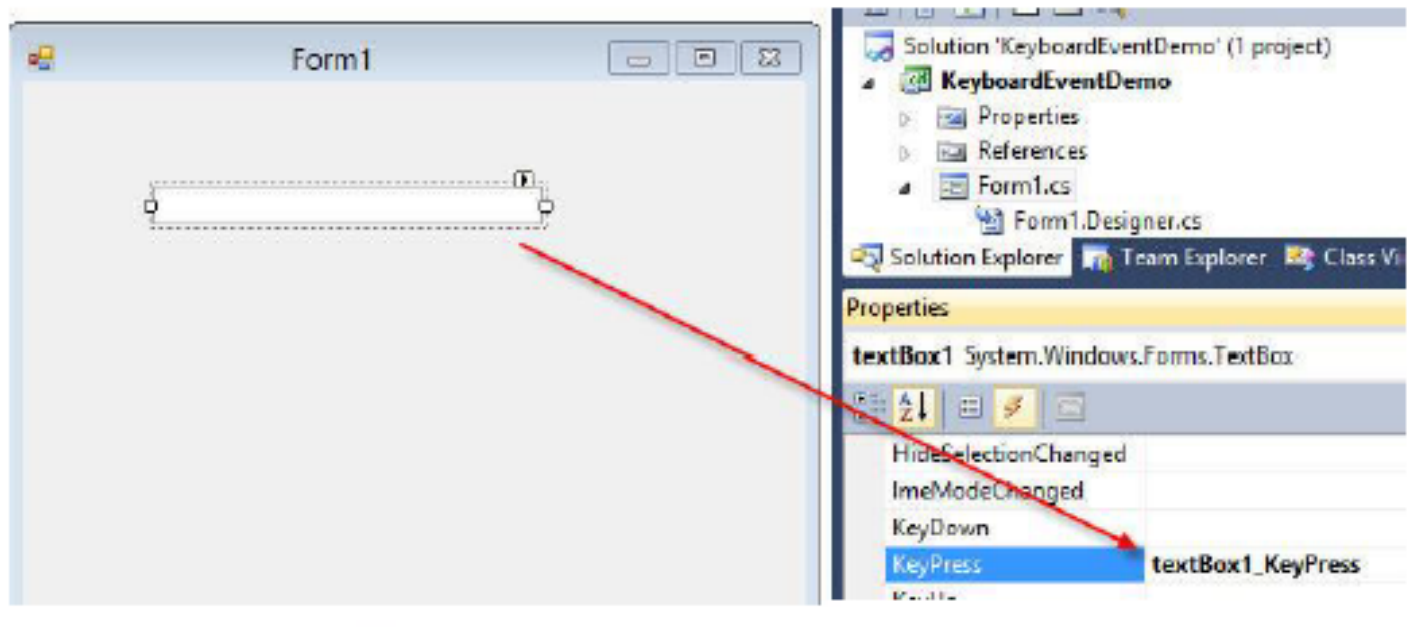
# Sự kiện bàn phím

```
private void Form1_KeyPress(object sender, KeyPressEventArgs e)
{
    switch(e.KeyChar)
    {
        case '+':
            e.Handled = true;
            count++;
            break;
        case '-':
            e.Handled = true;
            count++;
            break;
        default:
            //...
            break;
    }
}
```



# Sự kiện bàn phím

- Ví dụ sử lý sự kiện keypress trên TextBox, chỉ cho nhập số





# Sự kiện bàn phím

- Ví dụ sử lý sự kiện keypress trên TextBox, chỉ cho nhập số

```
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!Char.IsDigit(e.KeyChar) &&
        !Char.IsControl(e.KeyChar))
        e.Handled = true;
}
```



# Sự kiện bàn phím

- KeyUp/keyDown:
  - KeyDown: xảy ra khi một phím được nhấn
  - KeyUp: xảy ra khi một phím được nhả
  - Hàm xử lý nhận đối số là đối tượng của lớp **KeyEventArgs** chứa thông tin về các ký tự non-character (Shift, Ctrl, Alt, PgUp, PgDn ...)



# Sự kiện bàn phím

- Thuộc tính của lớp KeyEventArgs:
  - KeyCode: Trả về một giá trị trong enum Keys tương ứng với một phím được nhấn.
    - Ví dụ: nhấn phím A thì KeyCode trả về Key.A
  - KeyData: Giá trị kiểu Keys kết hợp bởi nhiều phím (các phím modifier và phím thông thường). Thường dùng để kiểm tra tổ hợp phím được nhấn.
    - `if (e.KeyData == (Keys.Control | Keys.A))`
  - KeyValue: Giá trị của KeyCode theo kiểu integer





# Sự kiện bàn phím

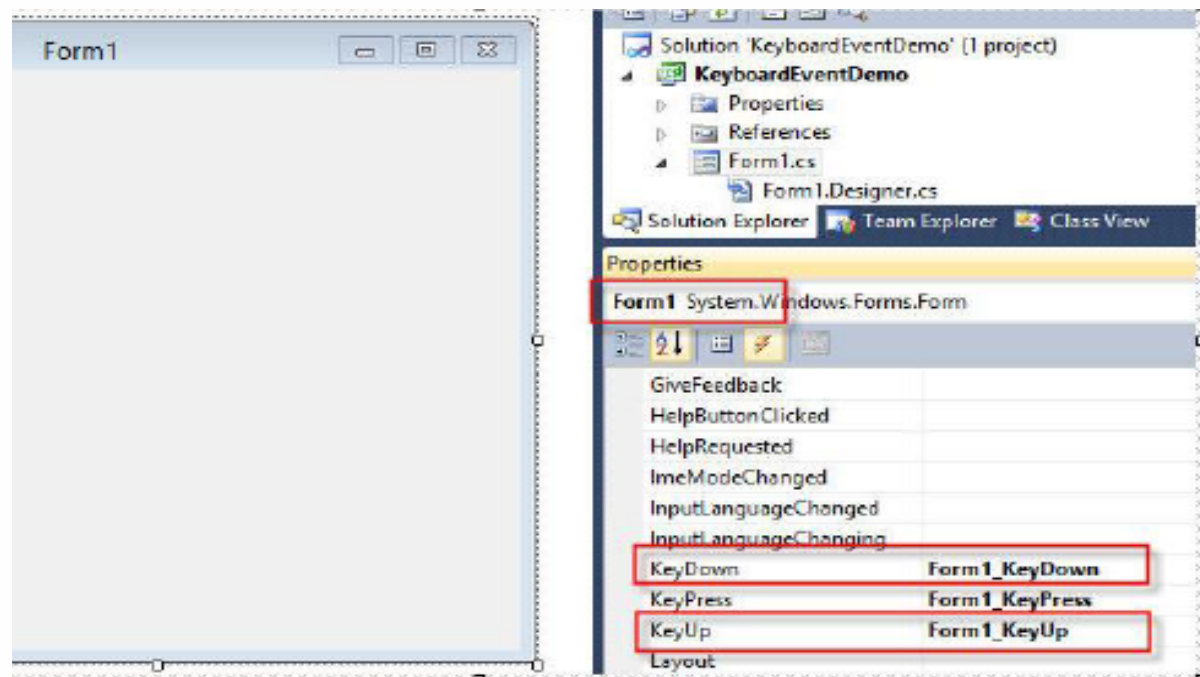
- Thuộc tính của lớp `KeyEventArgs`:
  - `Modifiers`: Xác định được các phím modifier nào được nhấn (`SHIFT`, `CTRL` và `ALT`).
  - `Property` này chỉ trả về các giá trị của các phím `SHIFT`, `CTRL`, `ALT` và `None`.
  - Có thể kiểm tra kết hợp các phím bằng toán tử `OR` `'|'`.

```
protected override void OnKeyDown(KeyEventArgs e)
{
    if (e.Modifiers == (Keys.Control | Keys.Shift | Keys.Alt))
        // do something
}
```



# Sự kiện bàn phím

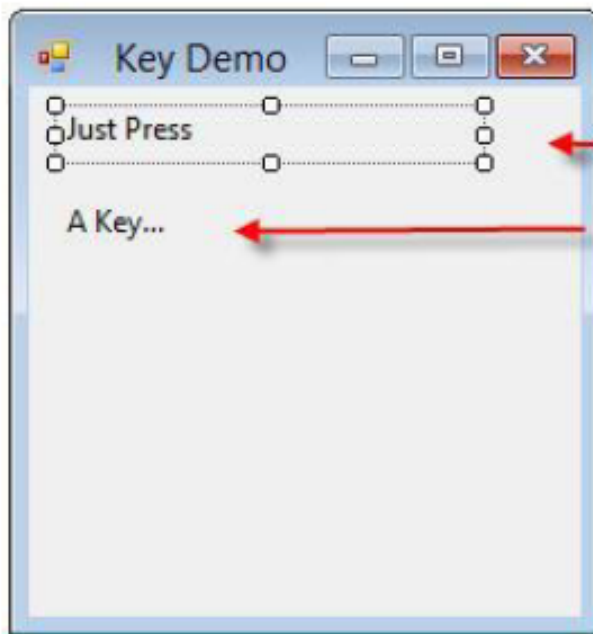
## ■ Xử lý sự kiệnKeyUp/keyDown :



```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.Control && e.KeyCode == Keys.A)
        // Xử lý khi người dùng nhấn Ctrl và A
}
}
```

# Sự kiện bàn phím – ví dụ

- Thiết kế giao diện như sau



Label (Name = charlabel,  
Text = Just Press)

Label (name = keyInfoLabel,  
Text = A Key...)



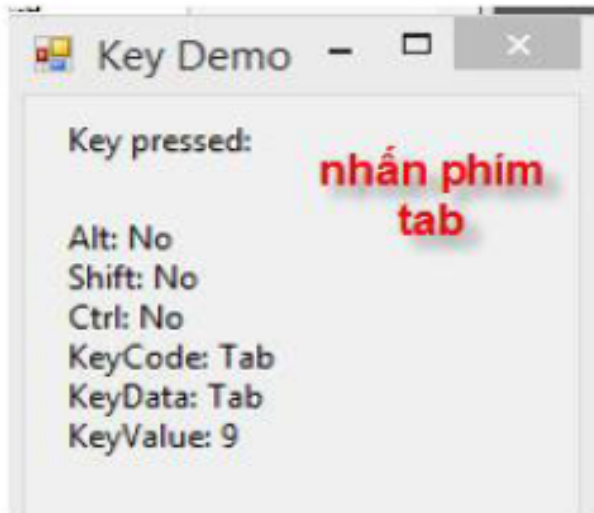
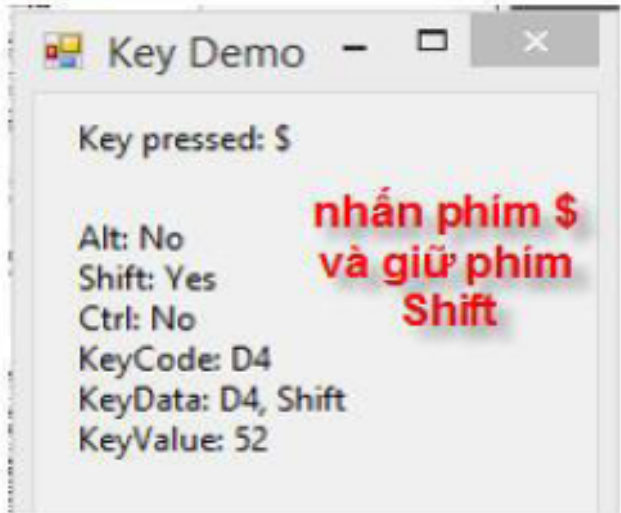
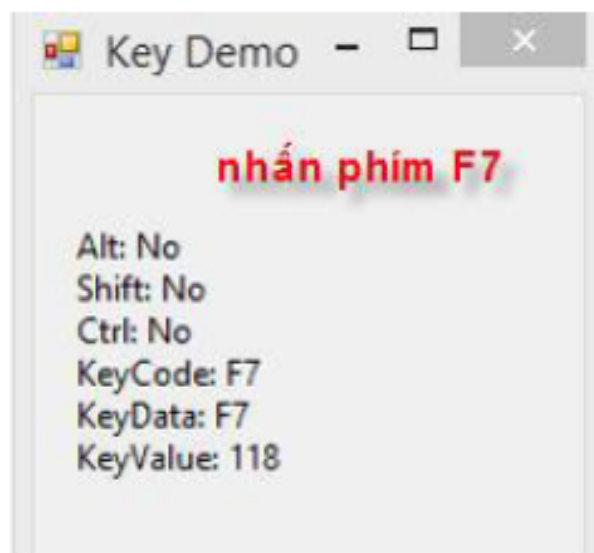
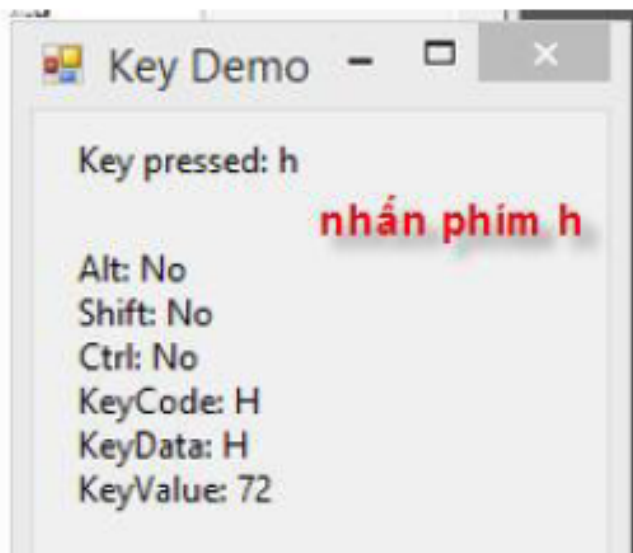
# Sự kiện bàn phím – ví dụ

```
private void KeyDemoForm_KeyPress(object sender, KeyPressEventArgs e )  
{// hiển thị ký tự với thuộc tính KeyChar  
    charLabel.Text = "Key pressed: " + e.KeyChar;  
}
```

```
private void KeyDemoForm_KeyDown( object sender, KeyEventArgs e )  
{ // hiển thị modifier keys, key code, key data và key value  
    keyInfoLabel.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;  
}
```

```
private void KeyDemoForm_KeyUp( object sender, KeyEventArgs e )  
{// xóa các chuỗi trên label khi nhả phím  
    charLabel.Text = "";  
    keyInfoLabel.Text = "";  
}
```

# Sự kiện bàn phím – ví dụ







# Sự kiện chuột

Bao gồm:

- MouseClick
- MouseDoubleClick
- MouseDown: nhấn phím chuột
- MouseMove: con trỏ chuột di chuyển
- MouseUp: nhả phím chuột
- MouseWheel: cuộn nút giữa chuột
- MouseEnter: con trỏ chuột đi vào phạm vi điều khiển
- MouseHover: con trỏ chuột dao động trên vùng điều khiển
- MouseLeave: con trỏ chuột đi rời khỏi vùng điều khiển



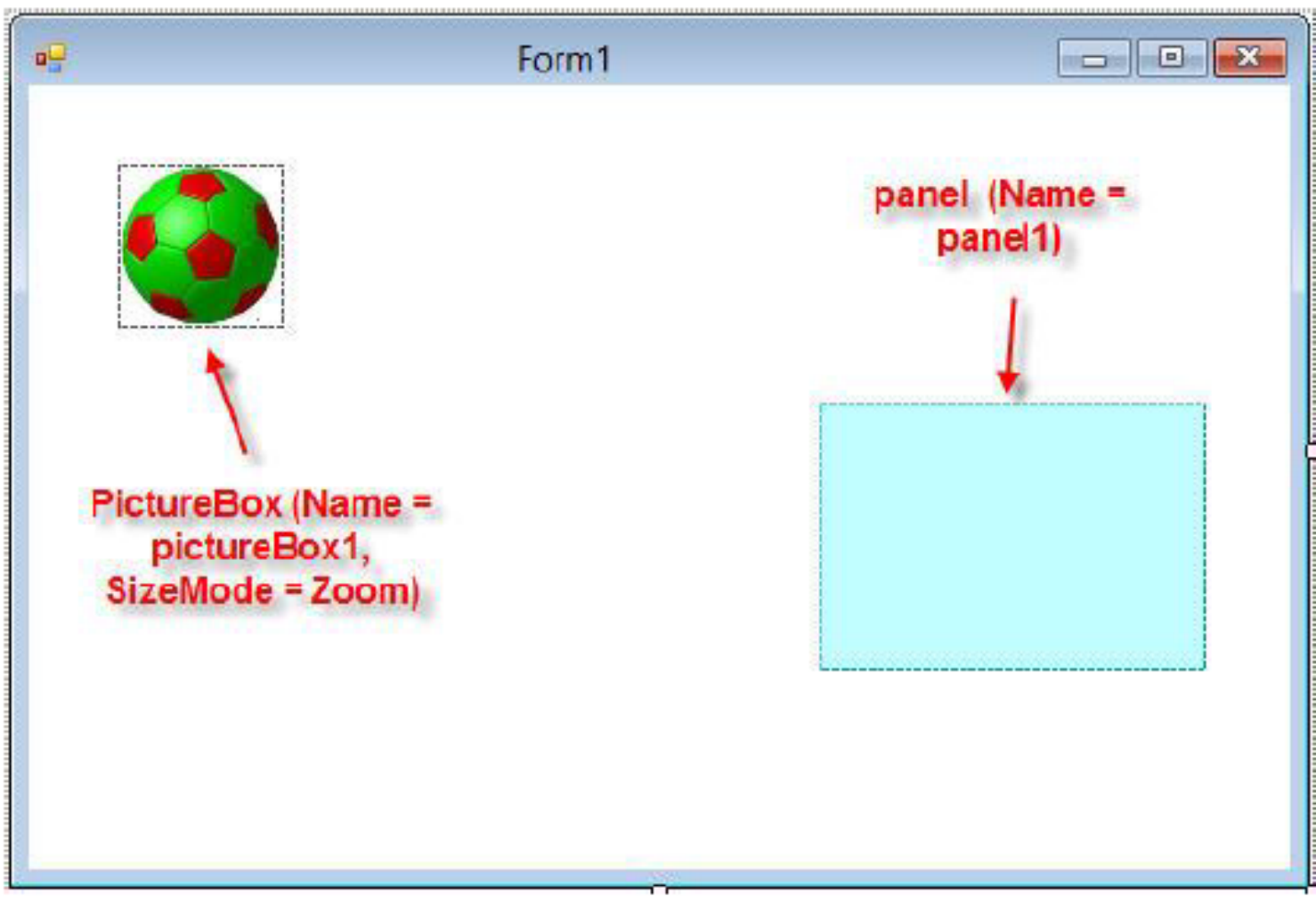
# Sự kiện chuột

- Hàm xử lý nhận đối số là đối tượng của lớp **MouseEventArgs**
- Thuộc tính của lớp MouseEventArgs:
  - Button: nút nào được nhấn
  - Clicks: số lần nhấn phím chuột
  - X: tọa độ x vị trí của chuột
  - Y: tọa độ y vị trí của chuột
  - Location: vị trí con trỏ chuột theo Point.





# Sự kiện chuột – ví dụ





# Sự kiện chuột – ví dụ

```
Point p01dpoint;
private void pictureBox1_MouseDown(object sender, MouseEventArgs e)
{
    p01dpoint = e.Location;
}
private void pictureBox1_MouseMove(object sender, MouseEventArgs e)
{
    if(e.Button==MouseButtons.Left)
    {
        int dx = e.X - p01dpoint.X;
        int dy = e.Y - p01dpoint.Y;
        pictureBox1.Left += dx;
        pictureBox1.Top += dy;
    }
}
private void pictureBox1_MouseUp(object sender, MouseEventArgs e)
{
    if (panel1.Bounds.Contains(pictureBox1.Bounds))
    {
        MessageBox.Show("finish!");
    }
}
```