

본 강의에서 수업자료로 이용되는 저작물은
저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거,
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로
수업자료의 재 복제, 대중 공개·공유 및 수업 목적 외의 사용을 금지합니다.

2024. 8. 30.

부천대학교·한국복제전송저작권협회

C#

5주차 2차시

10장 고급 폼

마우스 이벤트 다루기

5주차 학습 내용

* 1차시

- * 10장 고급 폼
 - * 메뉴 다루기
 - * 4장 델리게이트, 이벤트

* 2차시

- * 10장 고급 폼
 - * 마우스 이벤트 다루기

* 3차시

- * 10장 고급 폼
 - * 키보드 이벤트 다루기
 - * 5주차 과제

[클래스 멤버의 종류]

-자료 멤버

필드, 상수, 이벤트

-함수멤버

메서드, 생성자, 소멸자, 프러퍼티 (속성), 인덱서(색인자), 연산자 중복

-내포형 멤버

클래스형, 구조형, 열거형, 인터페이스형, 대리(델리게이트)형

[객체 지향 프로그램 구성]

-클래스 정의

-객체 선언

-객체 생성

-객체의 멤버 접근 및 활용

C# 프로그래밍



제 10장 고급 폼
마우스 이벤트 다루기



목차

- 메뉴 다루기
- 이벤트 핸들러(이벤트 처리기)
- 델리게이트
- 마우스 이벤트 다루기
- 키보드 이벤트 다루기



마우스 이벤트 다루기

■ 마우스

- 윈도우 사용자에게 가장 편리하고 친숙한 입력장치
- 원폼 애플리케이션의 사용자 상호작용은 대부분 마우스를 통해 이루어 짐
- 사용자가 마우스를 이동하거나 클릭하면 이벤트가 발생

■ 마우스 이벤트

- 이동 이벤트
 - 사용자가 마우스의 위치를 이동시킬 경우 발생
- 선택 이벤트
 - 사용자가 마우스의 버튼을 클릭할 경우 발생



마우스 이동 이벤트

- `mouseenter`
 - 마우스 포인터가 컨트롤이나 폼 영역에 들어올 때 발생
- `mouseover`
 - 마우스 포인터가 컨트롤이나 폼에서 이동하는 것을 멈출 때 발생
 - 매번 발생하지 않으며 처음 멈출 때만 발생
- `mouseleave`
 - 마우스 포인터가 컨트롤이나 폼 영역을 벗어날 때 발생
- `mousemove`
 - 마우스 포인터가 새로운 영역으로 이동할 때 발생
- `mousewheel`
 - 입력포커스를 가지고 있는 컨트롤이나 폼 위에서 마우스 휠 버튼을 회전시킬 때 발생



마우스 이동 이벤트 처리기

- **EventHandler** 델리게이트형의 처리기를 사용하는 이벤트
 - **MouseEnter, MouseHover, MouseLeave**
- **MouseEventHandler** 델리게이트형의 처리기를 사용하는 이벤트
 - **MouseMove, MouseWheel**
 - MouseEventArgs 클래스가 제공하는 프로퍼티를 이용하여 마우스의 위치와 상태에 대한 추가적인 정보 사용 가능

```
public delegate void EventHandler(object sender, EventArgs e);  
public delegate void MouseEventHandler(object sender, MouseEventArgs e);
```




MouseEventArgs 클래스의 프로퍼티

```
public delegate void MouseEventHandler(object sender, MouseEventArgs e);
```

■ Button

- 마우스의 상태를 나타내는 MouseButton 열거형 값
- **MouseButton** 열거형
 - Left : 마우스 왼쪽 버튼을 클릭한 상태
 - Middle : 마우스 중앙 버튼을 클릭한 상태
 - None : 마우스를 누르지 않은 상태
 - Right : 마우스 오른쪽 버튼을 클릭한 상태
 - XButton1 : 첫 번째 X버튼을 클릭한 상태
 - XButton2 : 두 번째 X버튼을 클릭한 상태

IntelliMouse에서
제공하는 버튼

■ Clicks

- 마우스 버튼을 클릭한 횟수

■ Delta

- 마우스 휠의 회전수(휠을 1회 돌리는 것)를 나타내는 값

■ X

- 클라이언트 좌표 내에서, 마우스 위치의 X좌표

■ Y

- 클라이언트 좌표 내에서, 마우스 위치의 Y좌표



마우스 이동 예제

예제 10.3 [ex10_3_xxxx] : MouseWheel 이벤트

1) 폼 설계



2) 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	MouseWheelApp
TextBox:textBox1	Text	0
Button:button1	Text	Reset

4) 멤버

private int num;

3) 이벤트 처리기

컨트롤 : (Name)	이벤트	메소드명
Form : Form1	MouseWheel	Form1_MouseWheel()
Button : button1	Click	Button1_Click()

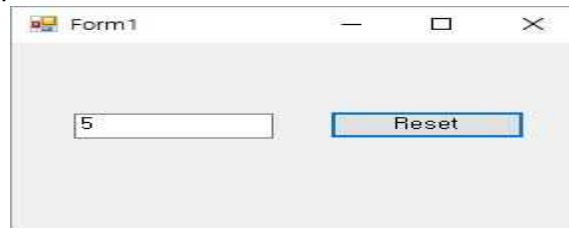


마우스 이동 예제

```
private void Form1_MouseWheel(object sender, MouseEventArgs e)
{
    textBox1.Text = (e.Delta > 0 ? num++ : num--).ToString();
}
private void button1_Click(object sender, EventArgs e)
{
    num = 0;
    textBox1.Text = num.ToString();
}
```

실행 방법 : 폼 위로 마우스의 휠을 이동하여 MouseWheel 이벤트를 발생시킨다.

실행 결과 :



MouseWheel 이벤트 처리 과정

- 1) Form1.cs에 이벤트를 처리기를 작성한다.

```
private void Form1_MouseWheel(object sender, MouseEventArgs e)
{
    textBox1.Text = (e.Delta > 0 ? num++ : num--).ToString();
}
```

- 2) Form1.Designer.cs에 이벤트 등록을 시켜야 한다.

```
this.MouseWheel += new System.Windows.Forms.MouseEventHandler(Form1_MouseWheel);
```



마우스 이동 예제

예제 10.4 [ex10_4_xxxx] : MouseEnter 이벤트

1) 폼 설계



2) 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	MousePositionApp

3) 이벤트 처리기

컨트롤 : (Name)	이벤트	메소드명
Form : Form1	MouseEnter	Form1_MouseEnter()

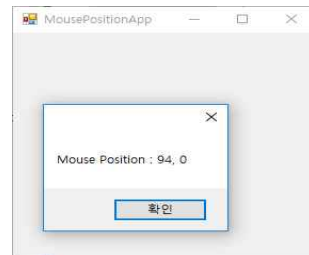


마우스 이동 예제

```
private void Form1_MouseEnter(object sender, EventArgs e) {  
    Point mousePoint = PointToClient(MousePosition);  
    string msg = "Mouse Position : " + mousePoint.X + ", "  
                + mousePoint.Y;  
    MessageBox.Show(msg);  
}
```

실행 방법 : 폼 위로 마우스를 이동하여 MouseEnter 이벤트를 발생시킨다.

실행 결과 :



■ MousePosition 프로퍼티

- 마우스의 좌표를 **전체화면에** 대한 상대좌표로 Point 구조체 형으로 반환

■ PointToClient 메소드

- 전체화면에 대한 상대좌표를 클라이언트 좌표로 변환



마우스 선택 이벤트

- MouseDown
 - 폼이나 컨트롤에서 마우스 버튼을 누를 때 발생
- MouseUp
 - 폼이나 컨트롤에서 마우스 버튼을 누른 후 해제할 때 발생
- Click
 - 폼이나 컨트롤을 클릭할 때 발생
- DoubleClick
 - 폼이나 컨트롤을 더블 클릭할 때 발생



마우스 선택 이벤트 처리기

- EventHandler 델리게이트형의 처리기를 사용하는 이벤트
 - Click, DoubleClick
- MouseEventHandler 델리게이트형의 처리기를 사용하는 이벤트
 - MouseDown, MouseUp



마우스 이벤트의 발생 순서

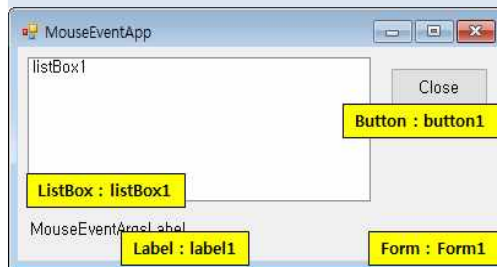
- 마우스의 이벤트가 비동기적으로 불특정 시간에 발생하더라도 상대적인 순서는 보장됨
 - MouseEnter와 MouseLeave사이에 발생하는 이벤트
 - MouseHover : 일정 시간 머무를 때 발생
 - MouseMove
 - Click 이벤트
 - MouseDown과 MouseUp 이벤트 다음에 발생
 - DoubleClick 이벤트
 - Click 이벤트 다음에 발생



마우스 이벤트의 발생 순서

예제 10.5 [ex10_5_xxxx] : 마우스 선택 이벤트

1) 폼 설계



2) 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	MouseEventApp
ListBox : listBox1	Items	
Button : button1	Text	Close
Label : label1	Text	MouseEventArgsLabel

3) 멤버

```
private void UpdateEventLabels(string msg, int x, int y, MouseEventArgs e) {  
    string message = string.Format("{0} X:{1}, Y:{2}", msg, x, y);  
    string eventMsg = DateTime.Now.ToShortTimeString();  
    eventMsg += " " + message;  
    listBox1.Items.Insert(0, eventMsg);  
    listBox1.TopIndex = 0;  
    string mouseInfo;  
    if (e != null) {  
        mouseInfo = string.Format("Clicks: {0}, Delta: {1}, " + "Buttons: {2}",  
                                   e.Clicks, e.Delta, e.Button.ToString());  
    } else { mouseInfo = string.Format("Clicks: {0}", msg); }  
    label1.Text = mouseInfo;  
}
```



마우스 이벤트의 발생 순서

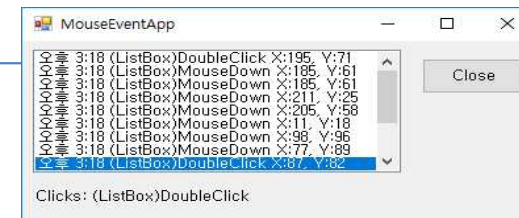
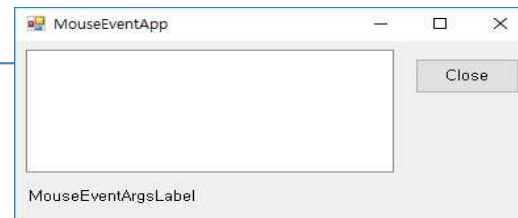
4) 이벤트 처리기

컨트롤 : (Name)	이벤트	메소드명
ListBox : listBox1	MouseDown	listBox1_MouseDown()
	DoubleClick	listBox1_DoubleClick()
Button : button1	Click	Button1_Click()

```
private void listBox1_MouseDown(object sender, MouseEventArgs e) {  
    UpdateEventLabels("(ListBox)MouseDown", ???, ???, ???);  
}  
private void listBox1_DoubleClick(object sender, EventArgs e) {  
    Point mousePoint = PointToClient(???);  
    UpdateEventLabels("(ListBox)DoubleClick", mousePoint.X, mousePoint.Y, ???);  
}  
private void button1_Click(object sender, EventArgs e) {  
    Application.Exit();  
}
```

실행 방법 : ListBox 상에서 MouseDown, DoubleClick 이벤트를 발생시킨다.

실행 결과 :





마우스 이벤트의 발생 순서

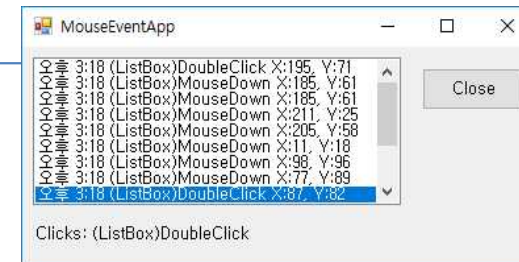
4) 이벤트 처리기

컨트롤 : (Name)	이벤트	메소드명
ListBox : listBox1	MouseDown	listBox1_MouseDown()
	DoubleClick	listBox1_DoubleClick()
Button : button1	Click	Button1_Click()

```
private void listBox1_MouseDown(object sender, MouseEventArgs e) {  
    UpdateEventLabels("(ListBox)MouseDown", e.X, e.Y, e);  
}  
private void listBox1_DoubleClick(object sender, EventArgs e) {  
    Point mousePoint = PointToClient(MousePosition);  
    UpdateEventLabels("(ListBox)DoubleClick", mousePoint.X, mousePoint.Y, null);  
}  
private void button1_Click(object sender, EventArgs e) {  
    Application.Exit();  
}
```

실행 방법 : ListBox 상에서 MouseDown, DoubleClick 이벤트를 발생시킨다.

실행 결과 :

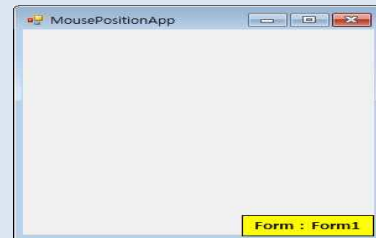




마우스 이벤트의 발생 순서

예제 10.6 [ex10_6_xxxx] : 마우스 이벤트 활용

1) 폼 설계



2) 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	HitTestingApp
Form : Form1	ResizeRedraw	True

3) 멤버

```
protected const int xNum = 5;    // 가로 박스 수
protected const int yNum = 4;    // 세로 박스 수
protected bool[,] abChecked = new bool[yNum, xNum];
protected int cxBlock, cyBlock;
private void InitializeComponent()
{
    // InitializeComponent 메소드 내에 추가

    this.OnResize(System.EventArgs.Empty);

    // 임의로 Resize이벤트를 발생시키는 메서드로서 폼이 실행되었을 때 사각형 박스의 배열을 그리기 위해 사용
}
```



마우스 이벤트의 발생 순서

4) 이벤트 처리기

```
private void Form1_Resize(object sender, System.EventArgs e) //폼의 크기를 변경하면 Resize 이벤트가 발생
{
    cxBlock = ClientSize.Width / xNum;
    cyBlock = ClientSize.Height / yNum;
    Invalidate();
}

private void Form1_MouseUp(object sender, System.Windows.Forms.MouseEventArgs e)
{
    int x = e.X / cxBlock;
    int y = e.Y / cyBlock;
    if (x < xNum && y < yNum)
    {
        abChecked[y, x] ^= true;
        Invalidate(new Rectangle(x * cxBlock, y * cyBlock, cxBlock, cyBlock));
    }
}

private void Form1_Paint(object sender, System.Windows.Forms.PaintEventArgs e)//사각형 배열을 새롭게 그린다.
{
    Graphics g = e.Graphics;
    Pen pen = new Pen(ForeColor);
    for (int y = 0; y < yNum; y++)
        for (int x = 0; x < xNum; x++)
        {
            g.DrawRectangle(pen, x * cxBlock, y * cyBlock, cxBlock, cyBlock);
            if (abChecked[y, x])
            {
                g.DrawLine(pen, x * cxBlock, y * cyBlock, (x + 1) * cxBlock, (y + 1) * cyBlock);
                g.DrawLine(pen, x * cxBlock, (y + 1) * cyBlock, (x + 1) * cxBlock, y * cyBlock);
            }
        }
}
```

실행 방법 : 사각형을 클릭하여 MouseUp 이벤트를 발생시킨다.

실행 결과 :



마우스 이벤트

quiz 10_마우스이벤트1 [quiz10_me1_xxxx] : 마우스 이벤트 퀴즈

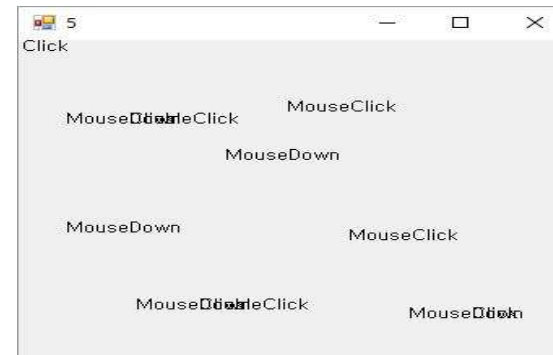
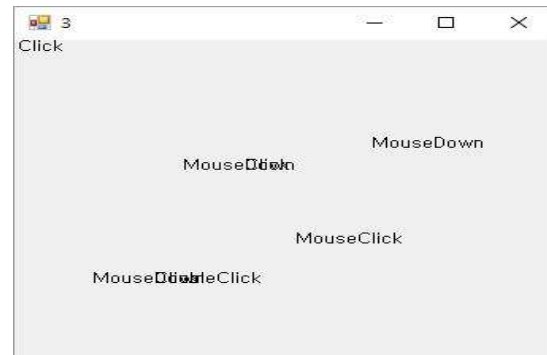
마우스 이벤트의 발생 순서를 알아본다.

1) 각 이벤트의 발생하는 **이벤트 이름 문자를 마우스 위치에 출력**한다.

- MouseClick
- MouseDown
- MouseDoubleClick
- Click

2) Click 이벤트일 경우는 카운트를 증가 시켜 타이틀 바에 표시함으로써 이벤트가 발생하는 것을 표시한다.

3) **System.Threading.Thread.Sleep(2000);**
을 이용하여 출력 을 살펴 본다.





마우스 이벤트

quiz 10_마우스이벤트1 [quiz10_me1_xxxx] : 마우스 이벤트 퀴즈

```
namespace MouseClick
{
    public partial class Form1 : Form
    {
        private int count = 0;

        private void Form1_MouseDown(object sender, MouseEventArgs e)
        {
            Graphics G = ???;
            G.DrawString("MouseDown", Font, Brushes.Black, ???, ???);
            G.Dispose();
        }

        private void Form1_MouseClick(object sender, MouseEventArgs e)
        {
            Graphics G = ???;
            G.DrawString("MouseClick", Font, Brushes.Black, ???, ???);
            G.Dispose();
        }

        private void Form1_MouseDoubleClick(object sender, MouseEventArgs e)
        {
            Graphics G = ???;
            G.DrawString("MouseDoubleClick", Font, Brushes.Black, ???, ???);
            G.Dispose();
        }

        private void Form1_Click(object sender, EventArgs e)
        {
            count++;
            Text = ???;
            Graphics G = ???;
            G.DrawString("MouseDoubleClick", Font, Brushes.Black, ???, ???);
            G.Dispose();
        }
    }
}

//이벤트 순서 : ???
```