

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 3차시

10장 고급 폼

키보드 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
키보드 이벤트 다루기



목차

- 메뉴 다루기
- 이벤트 핸들러
- 델리케이트
- 마우스 이벤트 다루기
- 키보드 이벤트 다루기



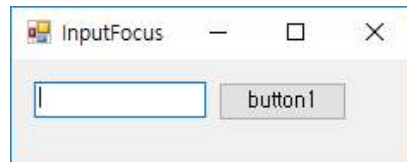
키보드 이벤트 다루기

- 원폼 애플리케이션은 사용자로부터 직접 키보드 입력을 받지 않음
 - 텍스트 박스와 같은 컨트롤을 이용하여 키보드 입력이 이루어짐
- C#은 컨트롤을 이용한 키보드 입력 이외에도 사용자 입력을 직접 처리할 수 있는 방법을 제공함

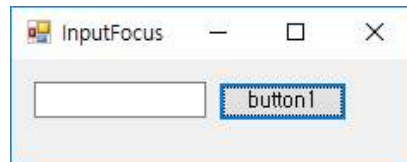


입력 포커스 [1/4]

- 키보드를 통해 입력이 가능한 컨트롤을 표시
 - 키보드를 이용한 사용자의 입력은 여러 개의 컨트롤에서 동시에 사용할 수 없음
 - 입력 포커스를 가지는 컨트롤만이 키보드를 통해 사용자의 입력을 받을 수 있음
 - 입력 포커스를 가지는 컨트롤은 자신의 형태를 변경함
 - 텍스트 상자가 입력 포커스를 가지는 경우



- 버튼 컨트롤이 입력 포커스를 가지는 경우





입력 포커스 [2/4]

■ Focus() 메소드

- 특정 컨트롤로 **입력 포커스를 이동시키기 위한 메소드**
- Control 클래스로부터 파생된 대부분의 컨트롤들이 가지는 메소드
- 특정 컨트롤에 대한 포커스가 변경될 경우, 참을 반환
- 특정 컨트롤에 대한 포커스가 변경되지 못할 경우, 거짓을 반환



입력 포커스 [3/4]

예제 10.7 [ex10_7_xxxx] : 입력 포커스

1) 폼 설계



2) 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	FocusApp
Button : button1	Text	버튼1
	BackColor	ControlDark
Button : button2	Text	버튼2

3) 이벤트 처리기

컨트롤 : (Name)	이벤트	메소드명
Button : button1	Click	button1_Click()
Button : button2	Click	button2_Click()

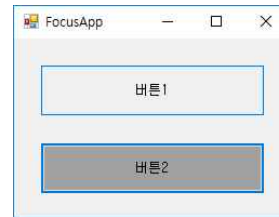


입력 포커스 [4/4]

```
private void button1_Click(object sender, EventArgs e) {  
    this.button1.BackColor = SystemColors.Control;  
    this.button2.Focus();  
    if (this.button2.Focused)  
        this.button2.BackColor = SystemColors.ControlDark;  
}  
private void button2_Click(object sender, EventArgs e) {  
    this.button2.BackColor = SystemColors.Control;  
    this.button1.Focus();  
    if (this.button1.Focused)  
        this.button1.BackColor = SystemColors.ControlDark;  
}
```

실행 방법 : 버튼1과 버튼2를 차례대로 선택하여 Click 이벤트를 발생시킨다.

실행 결과 :



- Focused 프로퍼티
 - 해당 컨트롤이 입력 포커스를 가졌는지 확인



키보드 이벤트

- KeyDown
 - 사용자가 키를 누를 때 발생
 - 키 상태와 보조키를 위한 **Keys 열거형** 정보를 사용할 수 있음
- KeyPress
 - 키가 완전히 눌려진 상태에서 발생
 - 키 문자에 대한 정보를 사용할 수 있음
- KeyUp
 - 키를 떼었을 때 발생
 - 키 상태와 보조키를 위한 **Keys 열거형** 정보를 사용할 수 있음
- 이벤트 발생순서
 - **KeyDown ➡ KeyPress ➡ KeyUp**



Keys 열거형

- 키보드로 입력된 모든 값이 정의된 열거형
 - System.Windows.Forms 네임스페이스에 포함
 - 183개의 값으로 구성
 - 키보드에 대한 열거형 뿐만 아니라 마우스에 대해서도 정의

기호상수	설 명	기호상수	설 명
A	문자 A	D3	숫자 3
F5	기능키 F5	NumPad3	숫자패드 3
LShiftKey	왼쪽 쉬프트 키	PageUp	페이지업 키
RControlKey	오른쪽 컨트롤 키	Delete	델 키
Left	왼쪽 화살표 키	Up	위쪽 화살표 키
Divide	나누기 키(/)	Lbutton	마우스 왼쪽 버튼



키보드 이벤트 처리하기 [1/4]

- **KeyDown, KeyUP 이벤트 처리기**
 - **EventArgs 클래스의 객체를 매개 변수로 가짐**
 - EventArgs 클래스는 키보드 입력을 직접 처리할 수 있는 프로퍼티를 제공

프로퍼티	설 명
Alt	<Alt>키를 눌렀는지 여부를 나타내는 값을 가져옴
Control	<Ctrl>키를 눌렀는지 여부를 나타내는 값을 가져옴
Handled	이벤트가 처리되었는지 여부를 나타내는 값을 가져오거나 설정
KeyCode	KeyDown 또는 KeyUP 이벤트에 대한 키보드 코드를 가져옴
KeyData	Keydown 또는 KeyUp 이벤트에 대한 키 데이터를 가져옴
KeyValue	Keydown 또는 KeyUp 이벤트에 대한 키보드 값을 가져옴
Modifiers	KeyDown 또는 KeyUp 이벤트에 대한 보조 플래그를 가져옴 이는 누른 보조키(<Ctrl>, <Shift> 및 <Alt>)의 조합을 나타냄
Shift	<Shift> 키가 눌렀는지 여부를 나타내는 값을 가져옴

- 키의 상태와 보조키에 대한 정보를 쉽게 얻을 수 있음



키보드 이벤트 처리하기 [2/4]

■ KeyPress 이벤트 처리기

■ KeyPressEventArgs 클래스의 객체를 매개변수로 가짐

- KeyPressEventArgs 클래스는 키 코드와 보조키에 대한 정보 대신에 눌러진 문자 값을 처리할 수 있는 프로퍼티를 제공

프로퍼티	설 명
Handled	이벤트가 처리되었는지 여부를 나타내는 값을 가져오거나 설정
KeyChar	눌려진 문자값

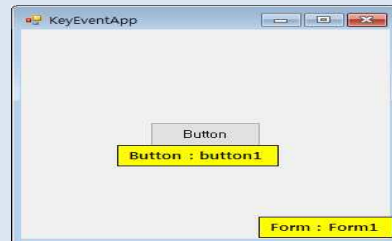
- KeyChar 프로퍼티는 사용자가 누른 키의 실제 문자 값을 반환
 - a키가 눌릴 경우 : 'a'를 반환
 - <Shift>+a가 눌릴 경우 : 'A'를 반환



키보드 이벤트 처리하기 [3/4]

예제 10.9 [ex10_9_xxxx] : 키보드 이벤트

1) 폼 설계



2) 프로퍼티

컨트롤 : (Name)	프로퍼티	값
Form : Form1	Text	KeyEventApp
Button : button1	Text	Button

3) 멤버

```
public int xPt, yPt;  
public static readonly int MOVE = 10;
```

3) 이벤트 처리기

컨트롤 : (Name)	이벤트	메소드명
Button : button1	KeyUp	button1_KeyUp()

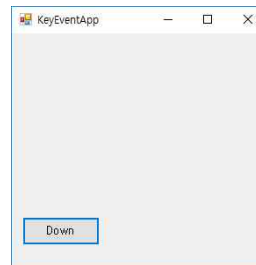
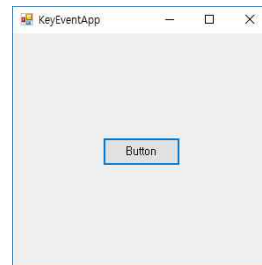


키보드 이벤트 처리하기 [4/4]

```
void button1_KeyUp(object sender, KeyEventArgs e) {  
    this.xPt = this.button1.Location.X;  
    this.yPt = this.button1.Location.Y;  
    switch (e.KeyCode) {  
        case KeyCode.Left:   
            xPt -= MOVE;    break;  
        case KeyCode.Right:   
            xPt += MOVE;    break;  
        case KeyCode.Up:   
            yPt -= MOVE;    break;  
        case KeyCode.Down:   
            yPt += MOVE;    break;  
    }  
    this.button1.Text = e.KeyCode.ToString();  
    this.button1.Location = new Point(xPt, yPt);  
}
```

실행 방법 : 키보드를 눌러서 KeyUp 이벤트를 발생시킨다.

실행 결과 :



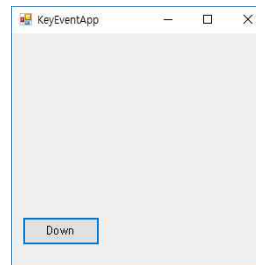
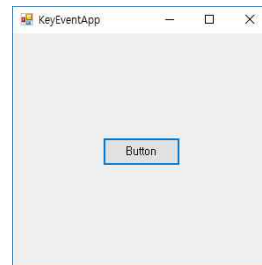


키보드 이벤트 처리하기 [4/4]

```
void button1_KeyUp(object sender, KeyEventArgs e) {  
    this.xPt = this.button1.Location.X;  
    this.yPt = this.button1.Location.Y;  
    switch (e.KeyCode) {  
        case Keys.Left :  
            xPt -= MOVE;    break;  
        case Keys.Right :  
            xPt += MOVE;    break;  
        case Keys.Up :  
            yPt -= MOVE;    break;  
        case Keys.Down :  
            yPt += MOVE;    break;  
    }  
    this.button1.Text = e.KeyCode.ToString();  
    this.button1.Location = new Point(xPt, yPt);  
}
```

실행 방법 : 키보드를 눌러서 KeyUp 이벤트를 발생시킨다.

실행 결과 :





Quiz 키보드이벤트 1

quiz 10_키보드이벤트1 [quiz10_ke1_xxxx] : keyPress 이벤트 퀴즈

KeyPress이벤트를 이용하여 키보드로 부터 문자들을 하나씩 입력 받아 문자열을 조립한다.

Form1

asss



Quiz 키보드이벤트 1

quiz 10_키보드이벤트1 [quiz10_ke1_xxxx] : keyPress 이벤트 퀴즈

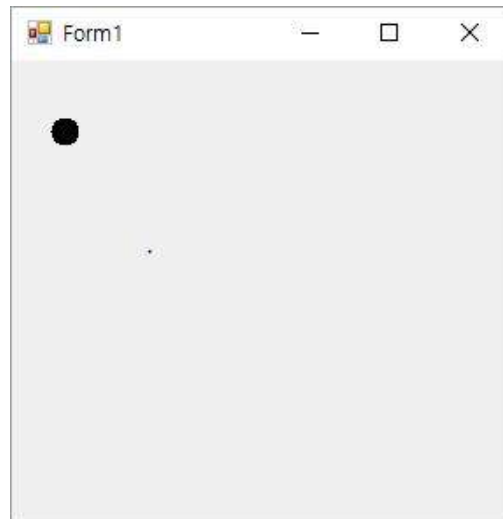
```
namespace KeyPress
{
    public partial class Form1 : Form
    {
        private string str;
        public Form1()
        {
            InitializeComponent();
            str = "";
        }
        private void Form1_Paint(object sender, PaintEventArgs e)
        {
            e.Graphics.DrawString(str, Font, Brushes.Black, 10, 10);
        }
        private void Form1_KeyPress(object sender, KeyPressEventArgs e)
        {
            /* 문자열 입력
            str += e.KeyChar;
            /*/
            /* 공백으로 문자열 삭제
            if (e.KeyChar == ' ')
            {
                str = "";
            }
            else
            {
                str += e.KeyChar;
            }
            /*/
            Invalidate();
        }
    }
}
```



Quiz 키보드이벤트 2

quiz 10_키보드이벤트2 [quiz10_ke2_xxxx] : keyDown 이벤트 퀴즈

폼의 특정 좌표에 타원을 하나 그리되 이동 키를 입력 받아 이 타원을 이동시키시오.





Quiz 키보드이벤트 2

quiz 10_키보드이벤트2 [quiz10_ke2_xxxx] : keyDown 이벤트 퀴즈

```
private int x, y;
public Form1()
{
    InitializeComponent();
    x = 10;
    y = 10;
}
private void Form1_Paint(object sender, PaintEventArgs e)
{
    e.Graphics.FillEllipse(Brushes.Black, ??? - 8, ??? - 8, 16, 16);
}
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    switch (e.???)
    {
        case ???Left:
            ???; //좌표 변경
            ???();
            break;
        case ???Right:
            ???;
            ??? ();
            break;
        case ???Up:
            ???;
            ??? ();
            break;
        case ???Down:
            ???;
            ??? ();
            break;
    }
}
```



Quiz 그림판 프로그램

quiz 10_그림판 [quiz10_draw_xxxx] : 다음과 같은 그림판 프로그램을 작성하시오.

