

윈도우 프로그래밍

13. 윈도우 폼(2)

심미나



목 차

- I. 윈도 폼: 윈도 폼 기본 익히기
- II. 윈도 폼: 윈도 폼에서 메서드 활용하기

II. 원도 폼

: 원도 폼에서 메서드 활용하기

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 정적 연결(디자인에서 연결)

- 화면 디자인

- 도구 상자에서 버튼(Button 클래스), 텍스트 박스(TextBox 클래스), 레이블(Label 클래스)을 드래그하여 화면 구성



- Form1.Designer.cs 파일에 자동 생성된 요소

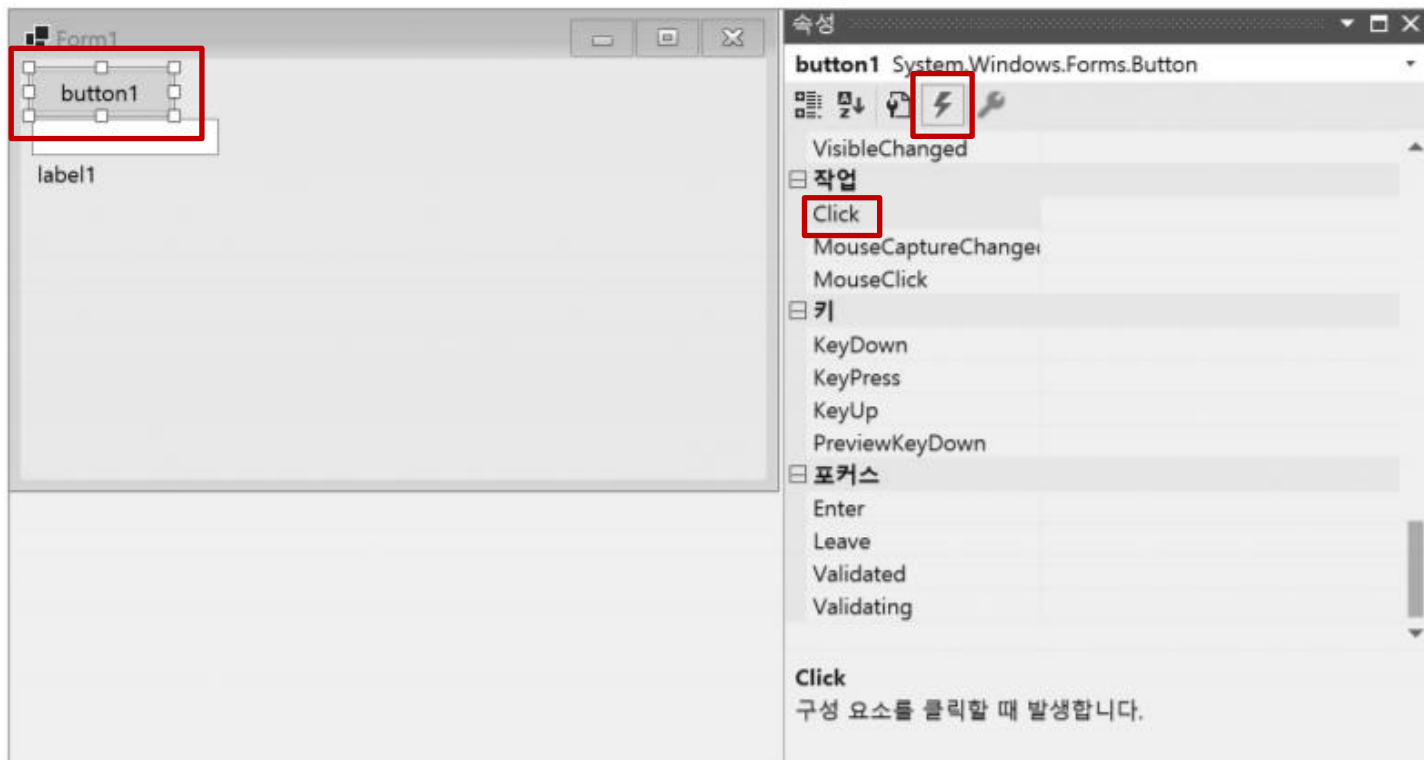
```
01 private System.Windows.Forms.Button button1;  
02 private System.Windows.Forms.TextBox textBox1;  
03 private System.Windows.Forms.Label label1;
```

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 정적 연결(디자인에서 연결)

- 버튼에 적용할 수 있는 이벤트
 - 디자인 화면에서 생성된 버튼 클릭, 속성 화면의 번개 모양 아이콘(이벤트) 클릭



윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 정적 연결(디자인에서 연결)

- 버튼에 적용할 수 있는 이벤트
 - Click 글자 자체 또는 옆 빈 공간을 잡고 더블 클릭하여 코드 확인

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }
}
```

```
private void button1_Click(object sender, EventArgs e)
{
}
}
```

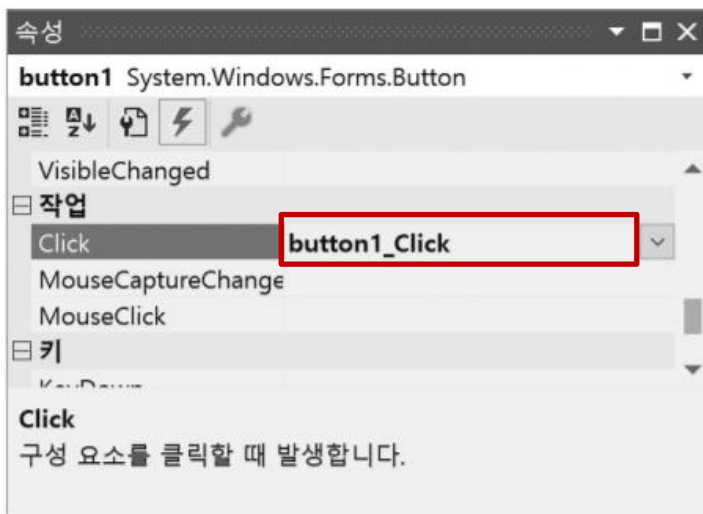
— 자동으로 생성된 이벤트 메서드

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 정적 연결(디자인에서 연결)

- 디자인 화면의 속성 화면
 - Click 부분에 자동 생성된 button1_Click ()메서드가 연결되었는지 확인
 - 연결된 이벤트



윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 정적 연결(디자인에서 연결)

- 버튼 클릭 메서드의 코드 작성

- Button1_click() 메서드

```
01 private void button1_Click(object sender, EventArgs e)
02 {
03     textBox1.Text += "+";
04     label1.Text += "+";
05 }
```

- 버튼을 여러 번 클릭했을 때의 화면



윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 동적 연결(코드에서 연결)

- 이벤트 동적 연결(코드에서 연결)

- Form1.Designer.cs 파일에 자동 생성된 코드

```
01 this.button1.Location = new System.Drawing.Point(12, 12);
02 this.button1.Name = "button1";
03 this.button1.Size = new System.Drawing.Size(75, 23);
04 this.button1.TabIndex = 0;
05 this.button1.Text = "button1";
06 this.button1.UseVisualStyleBackColor = true;
07 this.button1.Click += new System.EventHandler(this.button1_Click);
```

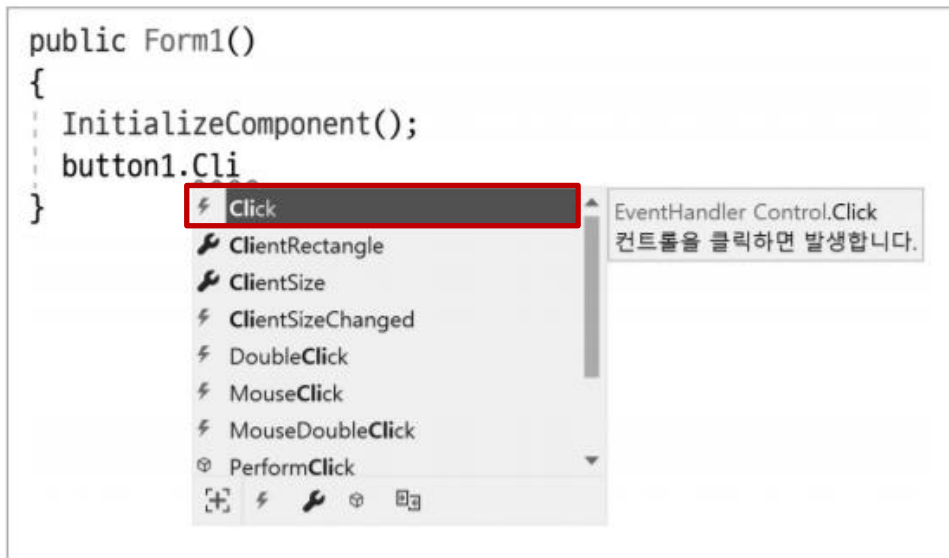
여기가 바로 이벤트를 연결하는 부분입니다.

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 동적 연결(코드에서 연결)

- Form1.cs 파일에서 button1에 Click 이벤트 연결(번개 모양의 이벤트)



윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 동적 연결(코드에서 연결)

- 이벤트 메서드 자동 작성, 이벤트 뒤에 + = 기호 입력
 - 자동 완성 기능으로 코드를 자동 삽입(Tab키를 두 번 누름)

```
public Form1()
{
    InitializeComponent();
    button1.Click +=
}
```

Button1_Click; (삽입하려면 <Tab> 키 누름)

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 동적 연결(코드에서 연결)

- 자동 생성된 이벤트 메서드

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        button1.Click += Button1_Click;
    }
}
```

```
private void Button1_Click(object sender, EventArgs e)
```

```
{
    throw new NotImplementedException();
}
```

구현되지 않았으니 구현해달라고 강제로 오류를 띄우는 코드입니다. 지금 버튼을 클릭하면 강제로 예외가 발생합니다. 이와 관련된 내용은 10장에서 다루지만 지금 기억해두면 그때 도움이 될 것입니다.

```
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text += "+";
    label1.Text += "+";
}
}
```

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 메서드의 매개변수

- 이벤트 메서드의 형태(2개의 매개변수 사용)

```
private void Button1_Click(object sender, EventArgs e)
{
}
}
```

- sender 객체

- 이벤트를 발생시킨 자기 자신을 나타냄
- sender 객체 활용
 - Ex) 버튼을 클릭한 자기자신의 Text 속성 변경하기

```
01 private void Button1_Click(object sender, EventArgs e)
02 {
03     Button self = (Button)sender;
04     self.Text = "저를 클릭했습니다!";
05 }
```

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 메서드의 매개변수

- 이벤트 정보 객체

- 이벤트와 관련된 추가 정보를 알려주는 객체
- e 객체 멤버에 대한 정보(속성, 메서드) 확인

```
private void button1_Click(object sender, EventArgs e)
{
    e.
```

Equals	bool object.Equals(object? obj)
GetHashCode	지정한 개체와 현재 개체가 같은지 여부를 확인합니다.
GetType	참고: 'Equals' 코드 조각을 삽입하려면 두 번 탭하세요.
ToString	

윈도 폼: 윈도 폼에서 메서드 활용하기



이벤트 메서드의 매개변수

- 이벤트 정보 객체
 - FormClosed 이벤트

```
01 public partial class Form1 : Form
02 {
03     public Form1()
04     {
05         InitializeComponent();
06         button1.Click += Button1_Click;
07         FormClosed += Form1_FormClosed;
08     }
09
10     private void Form1_FormClosed(object sender, FormClosedEventArgs e)
11     {
12         throw new NotImplementedException();
13     }
14
15     private void Button1_Click(object sender, EventArgs e) { /* 생략 */ }
16     private void button1_Click(object sender, EventArgs e) { /* 생략 */ }
17
18 }
```

자기 자신(Form1 클래스의 인스턴스)의 FormClosed 이벤트입니다.
이는 부모에서 상속받은 이벤트인데요 상속은 다음 장에서 알아보겠습니다.

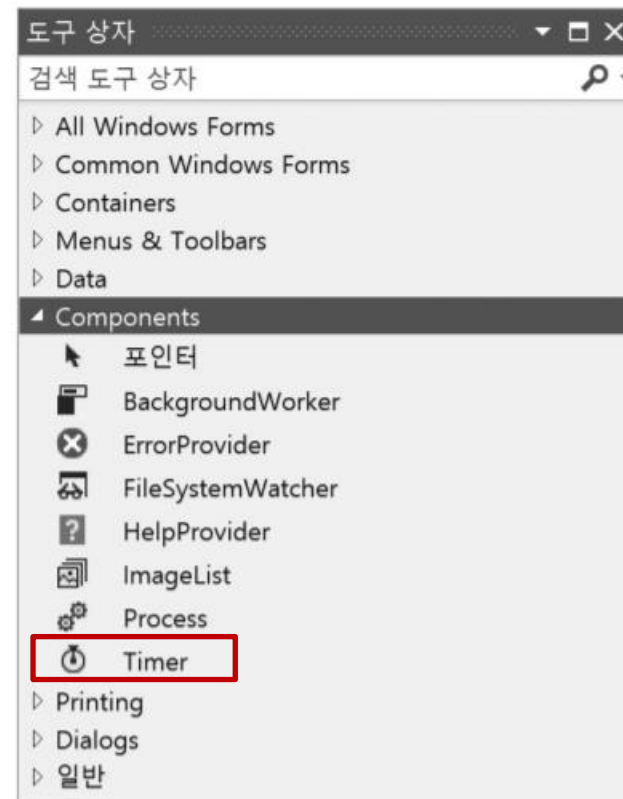
이벤트 정보 객체의 형식이 이전과 다릅니다.

윈도 폼: 윈도 폼에서 메서드 활용하기



백 그라운드 요소의 이벤트

- 폼의 백그라운드에서 작동하는 요소들
 - 도구상자의 구성 요소 확인

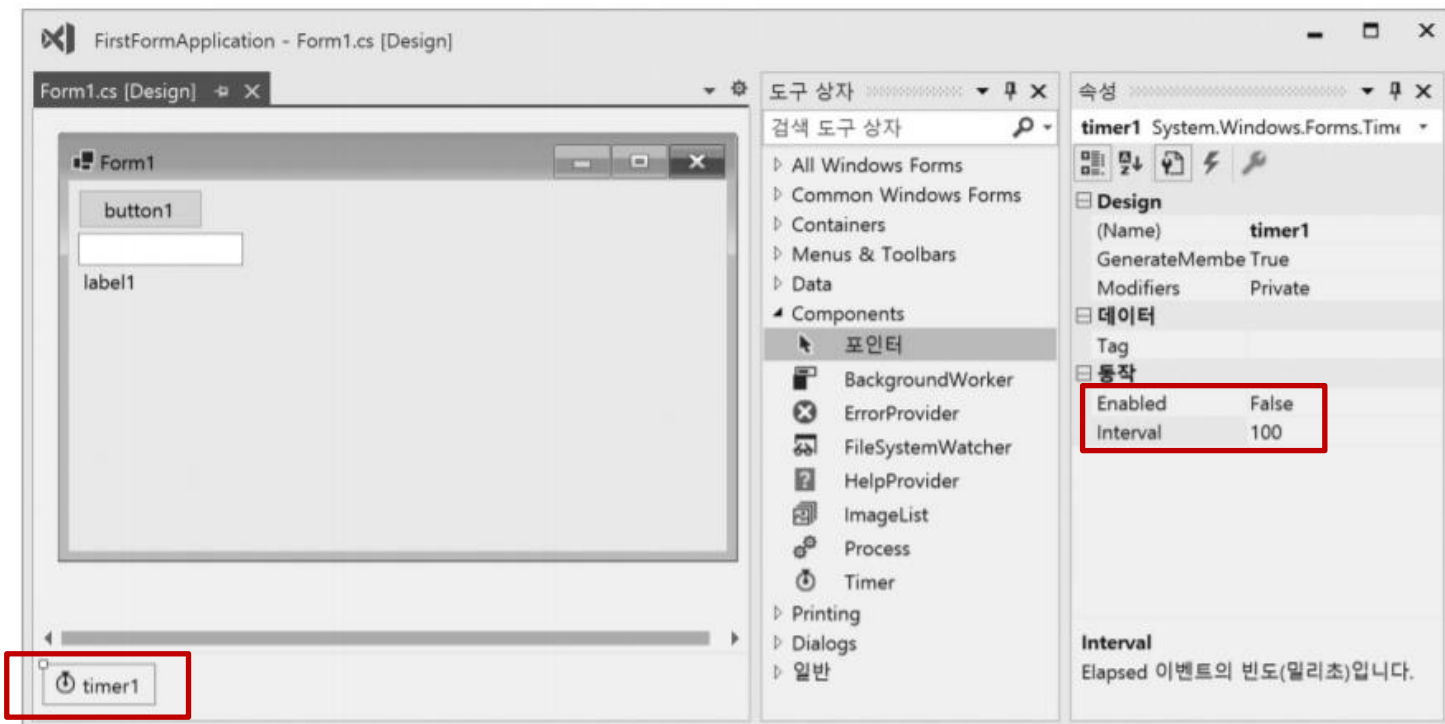


윈도 폼: 윈도 폼에서 메서드 활용하기



백 그라운드 요소의 이벤트

- 폼의 백그라운드에서 작동하는 요소들
 - 타이머(Timer 클래스)
 - 특정 시간 간격마다 특정한 코드를 호출해주는 기능



윈도 폼: 윈도 폼에서 메서드 활용하기

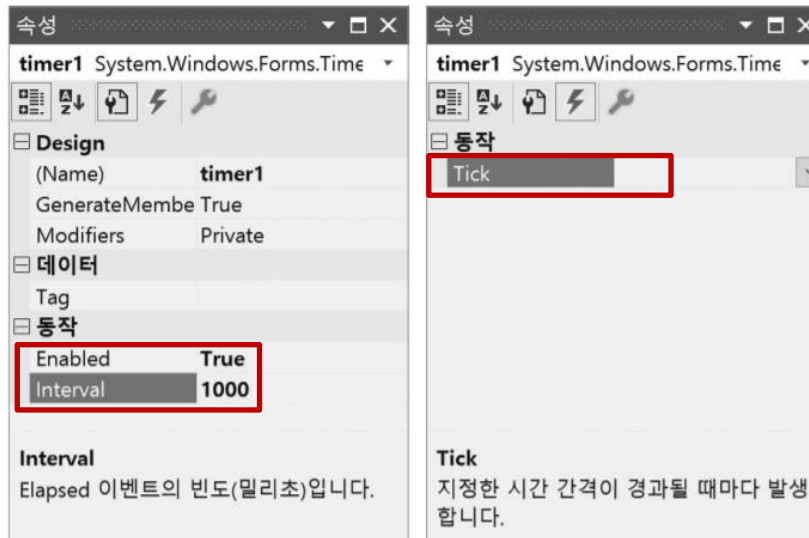


백 그라운드 요소의 이벤트

- 타이머를 이용한 프로그램

- 버튼을 누르면 버튼의 Enabled 속성을 true로 바꿔서 타이머를 작동시키고, 프로그램을 실행한 지 몇 초나 지났는지 확인하는 프로그램

1) 타이머 객체의 속성 지정



2) Tick 이벤트의 이벤트 메서드 자동생성

```
01 private void timer1_Tick(object sender, EventArgs e)
02 {
03
04 }
```

윈도 폼: 윈도 폼에서 메서드 활용하기



백 그라운드 요소의 이벤트

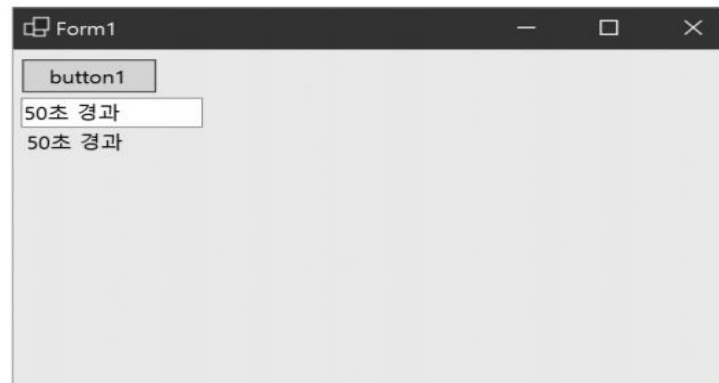
- 타이머를 이용한 프로그램

- 3) Tick 이벤트 메서드의 내부코드 작성

- 시간경과 확인 코드

```
01 private int elapsedTime = 0;  
02 private void timer1_Tick(object sender, EventArgs e)  
03 {  
04     elapsedTime++;  
05     textBox1.Text = elapsedTime + "초 경과";  
06     label1.Text = elapsedTime + "초 경과";  
07 }
```

- 프로그램이 시작된 지 몇 초가 지났는지 출력





감사합니다

mnshim@sungkyul.ac.kr

