

C# 모듈화 연습하기

제주대학교 컴퓨터공학과
변영철 교수

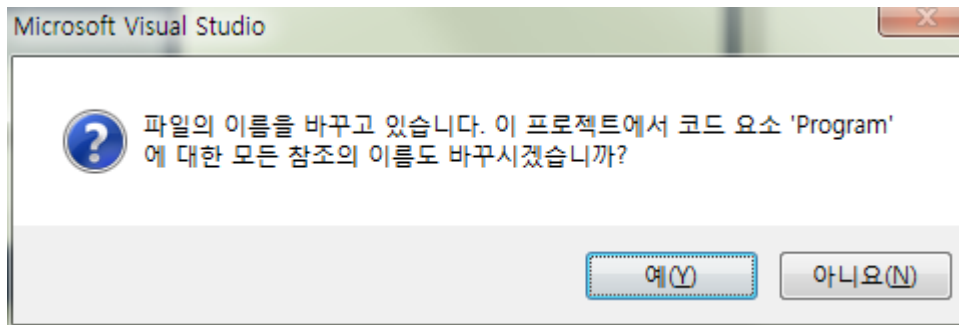
01. 프로젝트 생성

- 프로젝트 생성
 - Visual C# | Windows Forms 응용 프로그램
 - 'My' 프로젝트

02. 파일이름 바꾸기

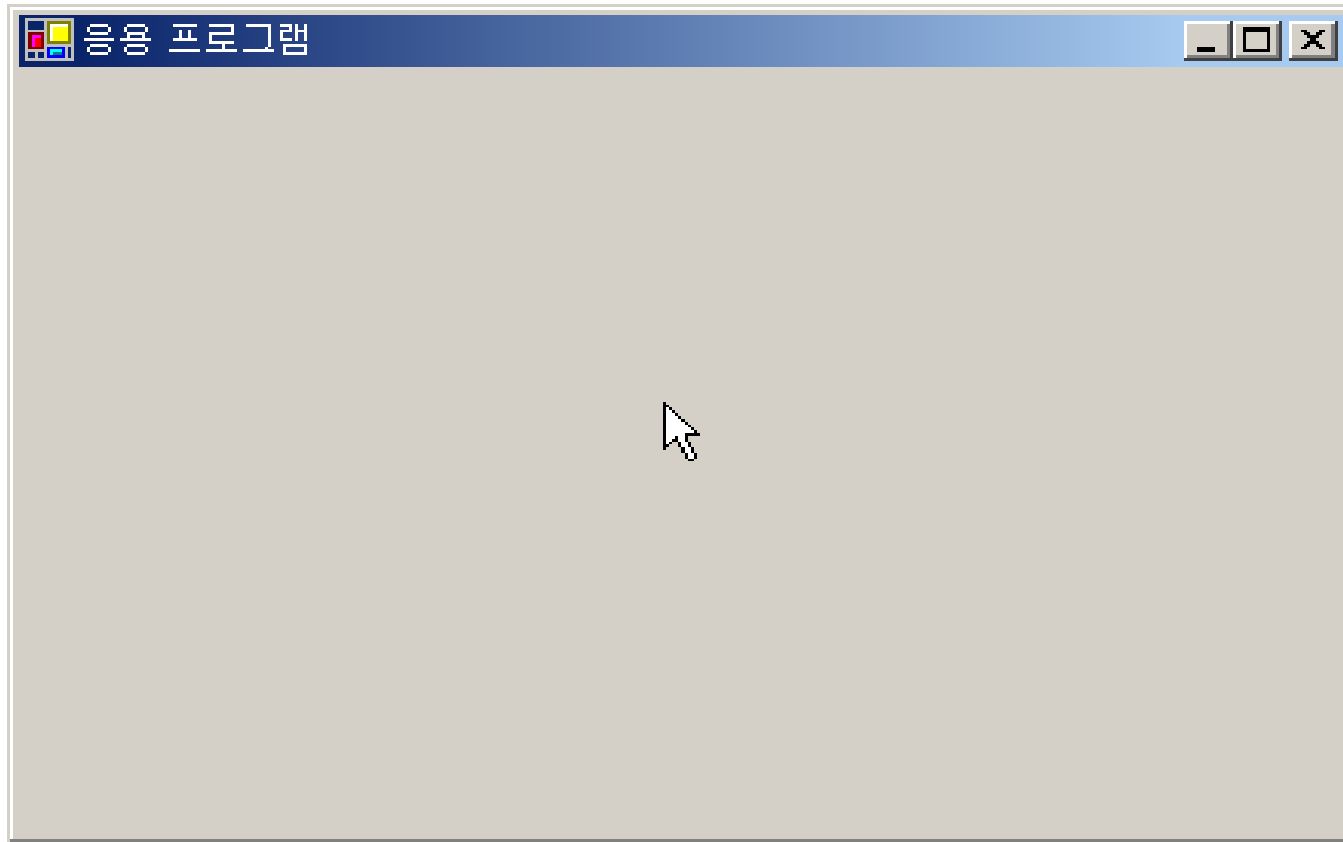
- 파일이름 바꾸기
 - Program.cs → MyApp.cs
 - Form1.cs → MainForm.cs

파일 이름 위에서
오른쪽 마우스 버튼 클릭,
'이름 바꾸기'



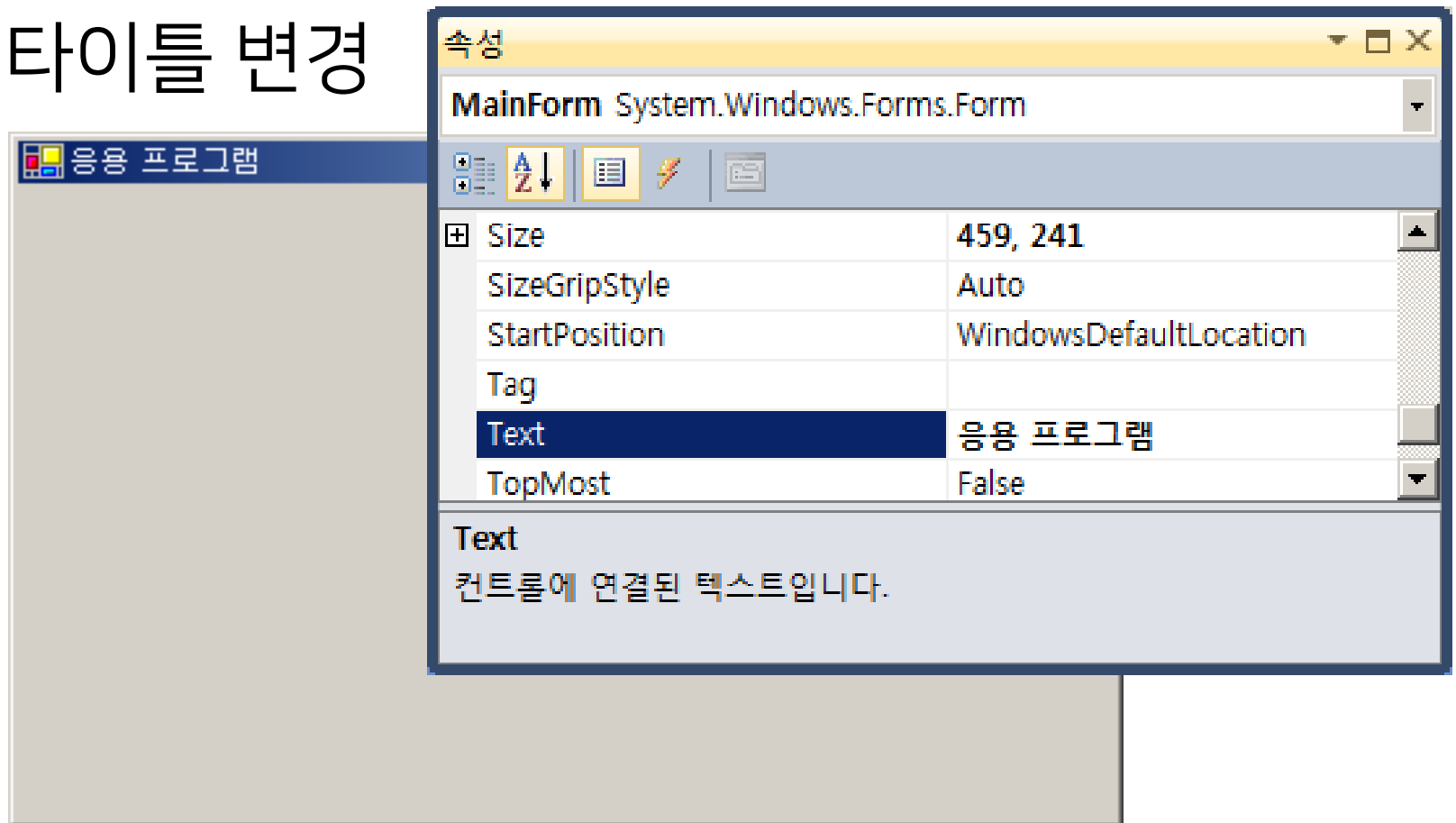
03. 폼 윈도우 비주얼 디자인

- 폼 윈도우 크기 바꾸기



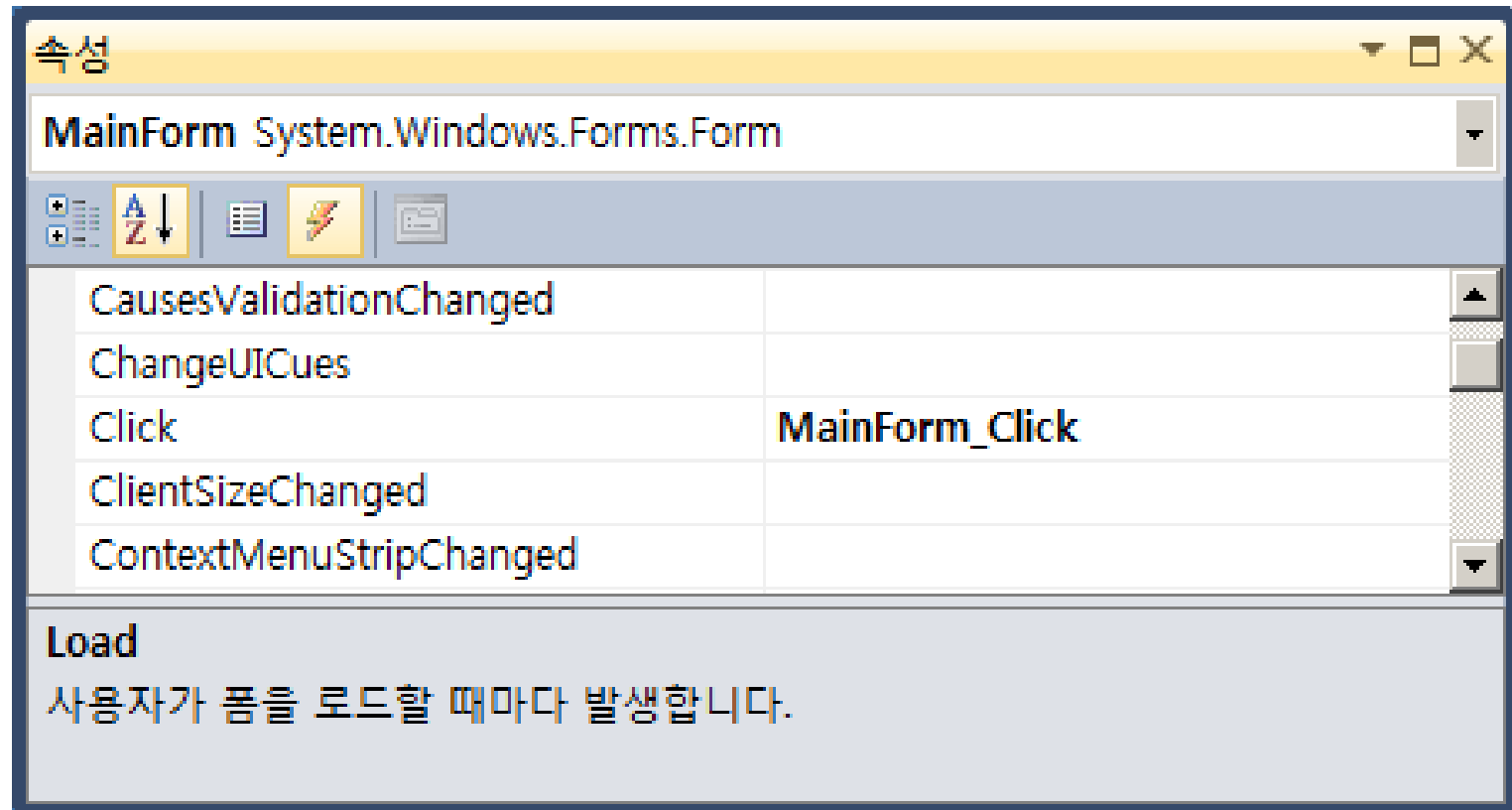
04. 폼 윈도우 비주얼 디자인

- 속성창 (보기 > 속성 메뉴) 표시
- 타이틀 변경



05. Click 이벤트 핸들러 함수 작성

- 속성창에서 번개 선택 후 Click을 더블 클릭
- 그러면 MainForm_Click 함수 자동 생성



06. 핸들러 함수 코드 작성

- 폼 윈도우 클릭할 경우 "Click!" 문자열 출력

화가 객체

```
private void MainForm_Click(object sender, EventArgs e)
{
    Graphics gildong = this.CreateGraphics();
    gildong.DrawString("Click!", Font,
        new SolidBrush(Color.Red), 10, 10);
}
```

07. 코드 추상화

```
public void DrawString() {  
    Graphics gildong = this.CreateGraphics();  
    gildong.DrawString("Click!", Font,  
        new SolidBrush(Color.Red), 10, 10);  
}  
  
private void MainForm_Click(object sender, EventArgs e)  
{  
    DrawString();  
}
```


08. 새로운 클래스로 이동 (1/4)

```
class DrawingUtil {
```

```
}
```

```
namespace My
```

```
{
```

```
    public partial class MainForm : Form
```

```
{
```

08. 새로운 클래스로 이동 (2/4)

• 클래스는 뭐하라고 있는 것?

```
class DrawingUtil {  
    public void DrawString() {  
        Graphics gildong = this.CreateGraphics();  
        gildong.DrawString("Click!", Font,  
            new SolidBrush(Color.Red), 10, 10);  
    }  
}
```

• 객체(util) 만들라고 있는 것

```
private void MainForm_Click(object sender, EventArgs e) {  
    DrawingUtil util = new DrawingUtil();  
    util.DrawString();  
}
```

08. 새로운 클래스로 이동 (3/4)

```
class DrawingUtil {  
    public void DrawString() {  
        Graphics gildong = this.CreateGraphics();  
        gildong.DrawString("Click!", Font,  
            new SolidBrush(Color.Red), 10, 10);  
    }  
}
```

- 하지만 오류가 발생한다.
- 이유는 무엇일까?
- 어떻게 고칠 수 있을까?

08. 새로운 클래스로 이동 (4/4)

```
class DrawingUtil {
```

```
    public void DrawString(Form f) {  
        Graphics gildong = f.CreateGraphics();  
        gildong.DrawString("Click!", f.Font,  
            new SolidBrush(Color.Red), 10, 10);
```

```
    }
```

```
}
```

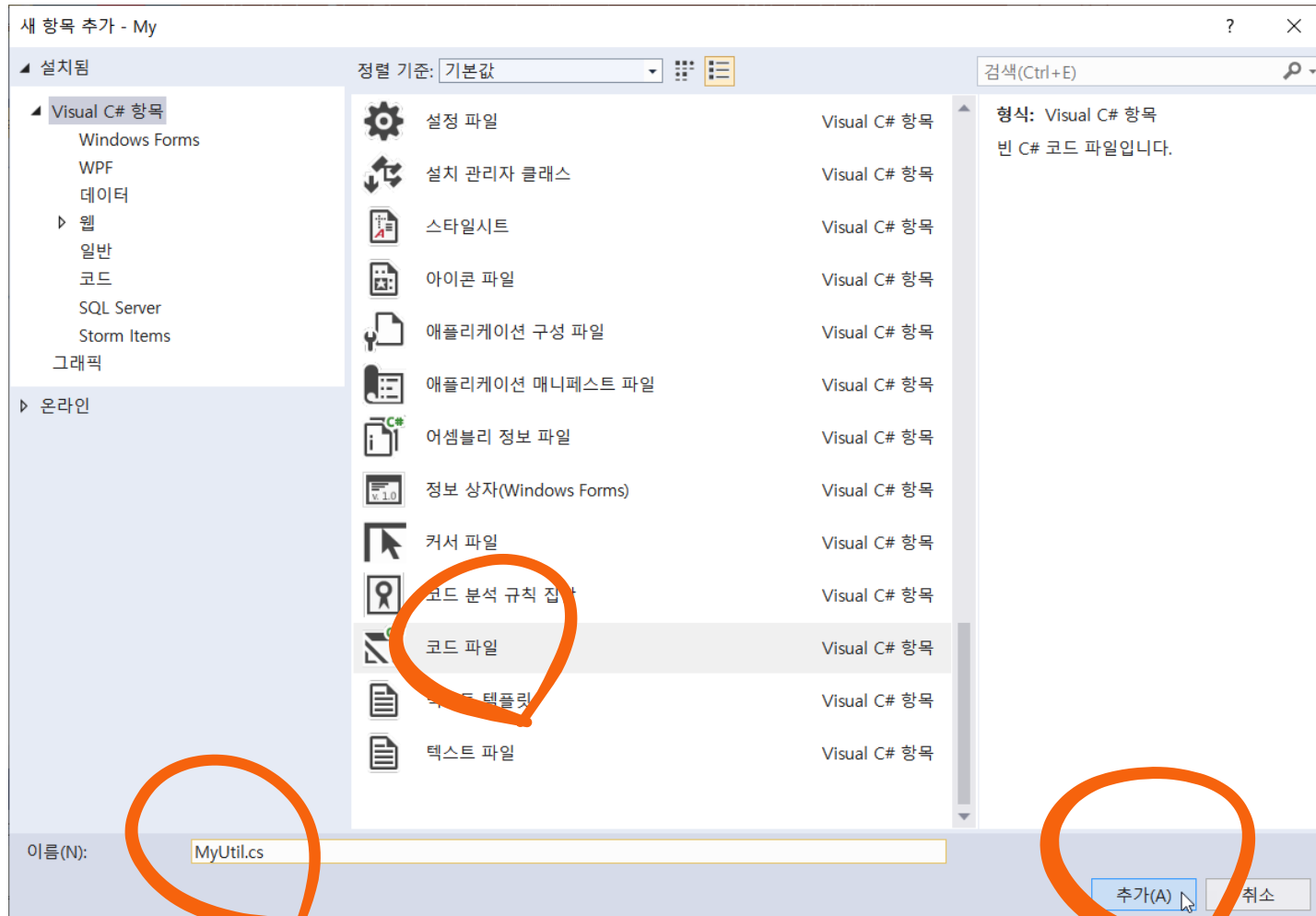
```
private void MainForm_Click(object sender, EventArgs e) {
```

```
    DrawingUtil util = new DrawingUtil();  
    util.DrawString(this);
```

```
}
```

09. 새로운 파일 만들기

- 프로젝트 | 새 항목 추가, MyUtil.cs



10. 새로운 파일로 클래스 옮기기

```
using System;  
using System.Drawing;  
using System.Windows.Forms;
```

- using 문도 함께 복사하면 좋다.
- 진한 색으로 되어 있는 것만...

```
class DrawingUtil {  
    public void DrawString(Form f) {  
        Graphics gildong = f.CreateGraphics();  
        gildong.DrawString("Click!", f.Font, new  
            SolidBrush(Color.Red), 10, 10);  
    }  
}
```

11. 파라미터 정리하기

```
class DrawingUtil {
```

```
    public void DrawString(Form f, String s, int x, int y) {
```

```
        Graphics gildong = f.CreateGraphics();
```

```
        gildong.DrawString("Click!", f.Font, new  
            SolidBrush(Color.Red), 10, 10);
```

```
    }
```

```
}
```

```
private void MainForm_Click(object sender, EventArgs e) {
```

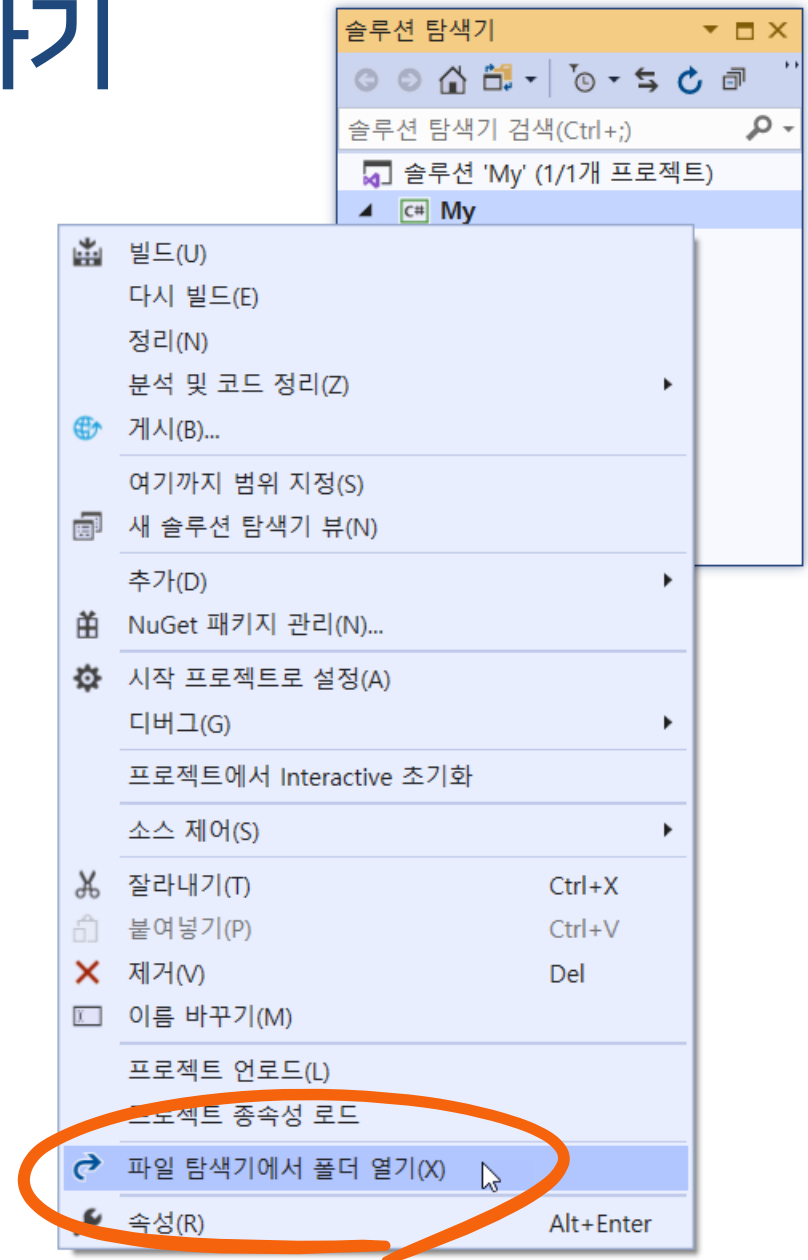
```
    DrawingUtil util = new DrawingUtil();
```

```
    util.DrawString(this, "Click!", 10, 10);
```

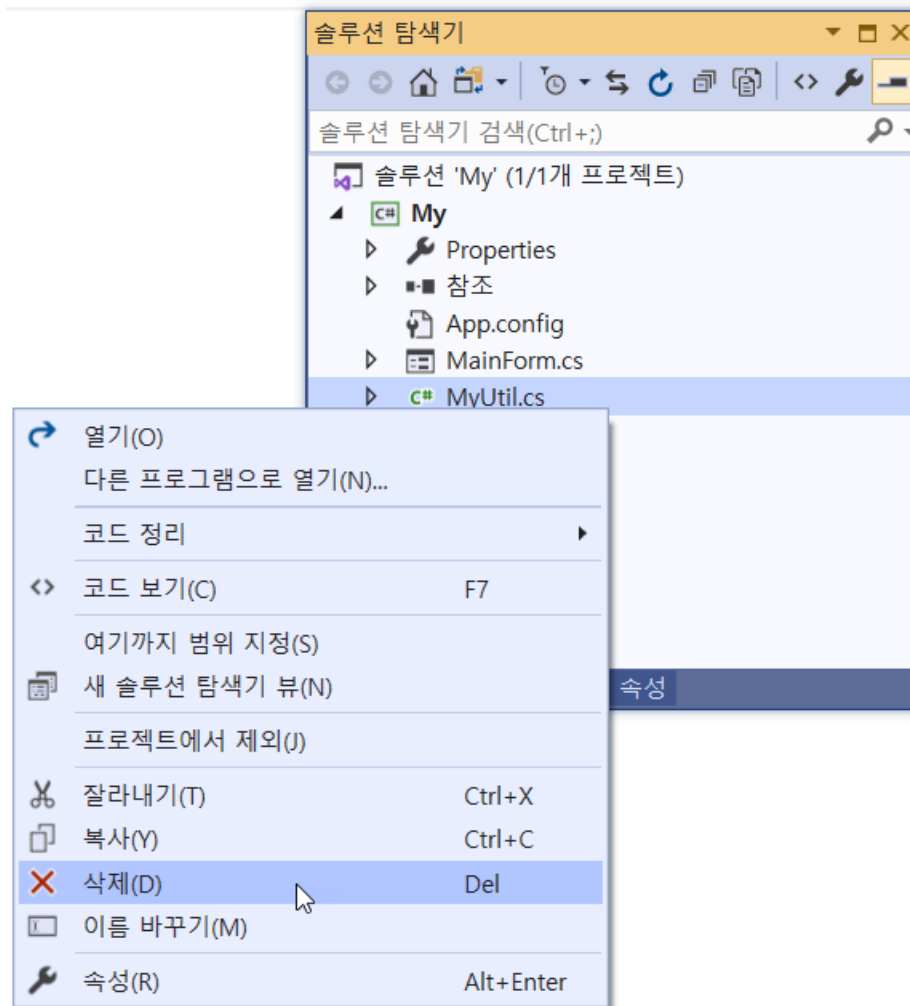
```
}
```

12. 외부 폴더로 이동하기

- 나만의 라이브러리 폴더로 (가령, c:\Wcslib)
- 탐색기로 MyUtil.cs 파일 찾아 이동하기

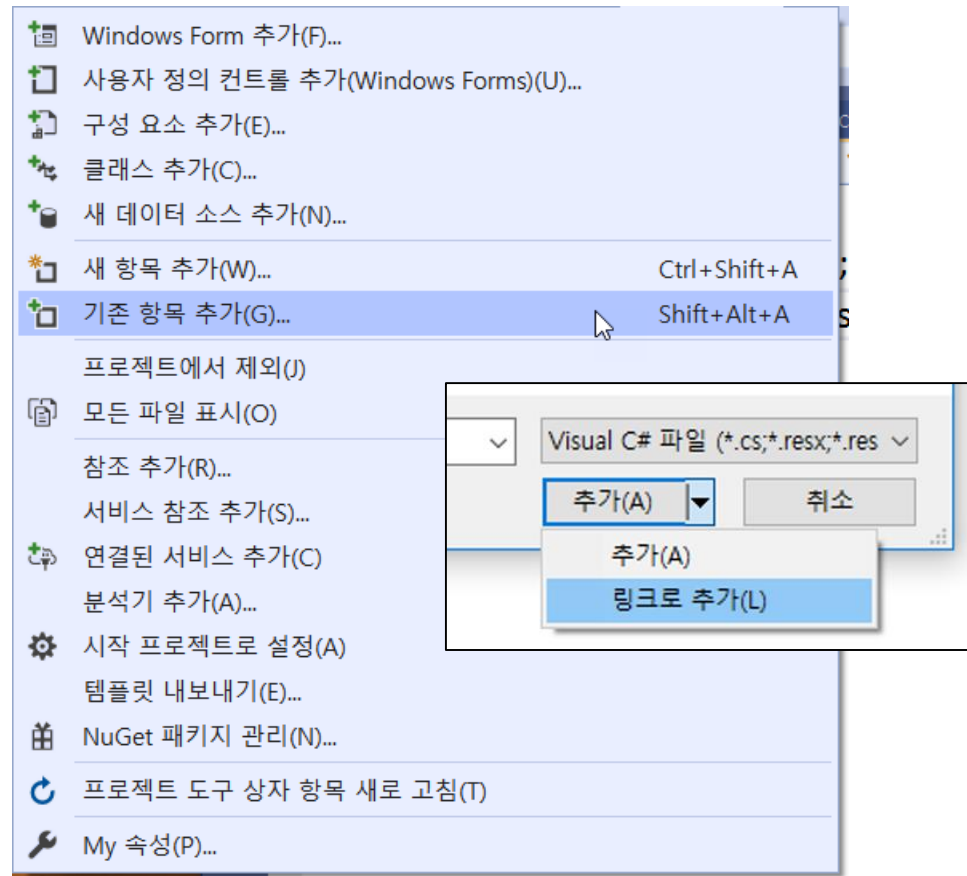


13. 기존의 파일 제거



14. 옮긴 파일을 프로젝트에 새로 추가

- 프로젝트 | 기존 항목 추가
- MyUtil.cs 파일 선택 후 '링크로 추가'



이 장을 공부하면

- 내가 작성한 프로그램에서 일부를 클래스 부품으로 만들 수 있다.
- 클래스 부품을 라이브러리 폴더로 모아둘 수 있다.
- 내가 만든 라이브러리를 쉽게 이용(재사용)할 수 있다.