

제7장

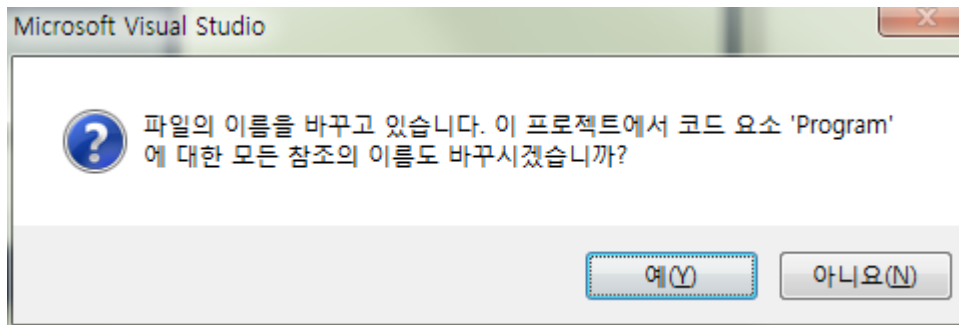
마술사를 이용한 코딩 연습

변영철

01. 프로젝트 작성 및 사전작업

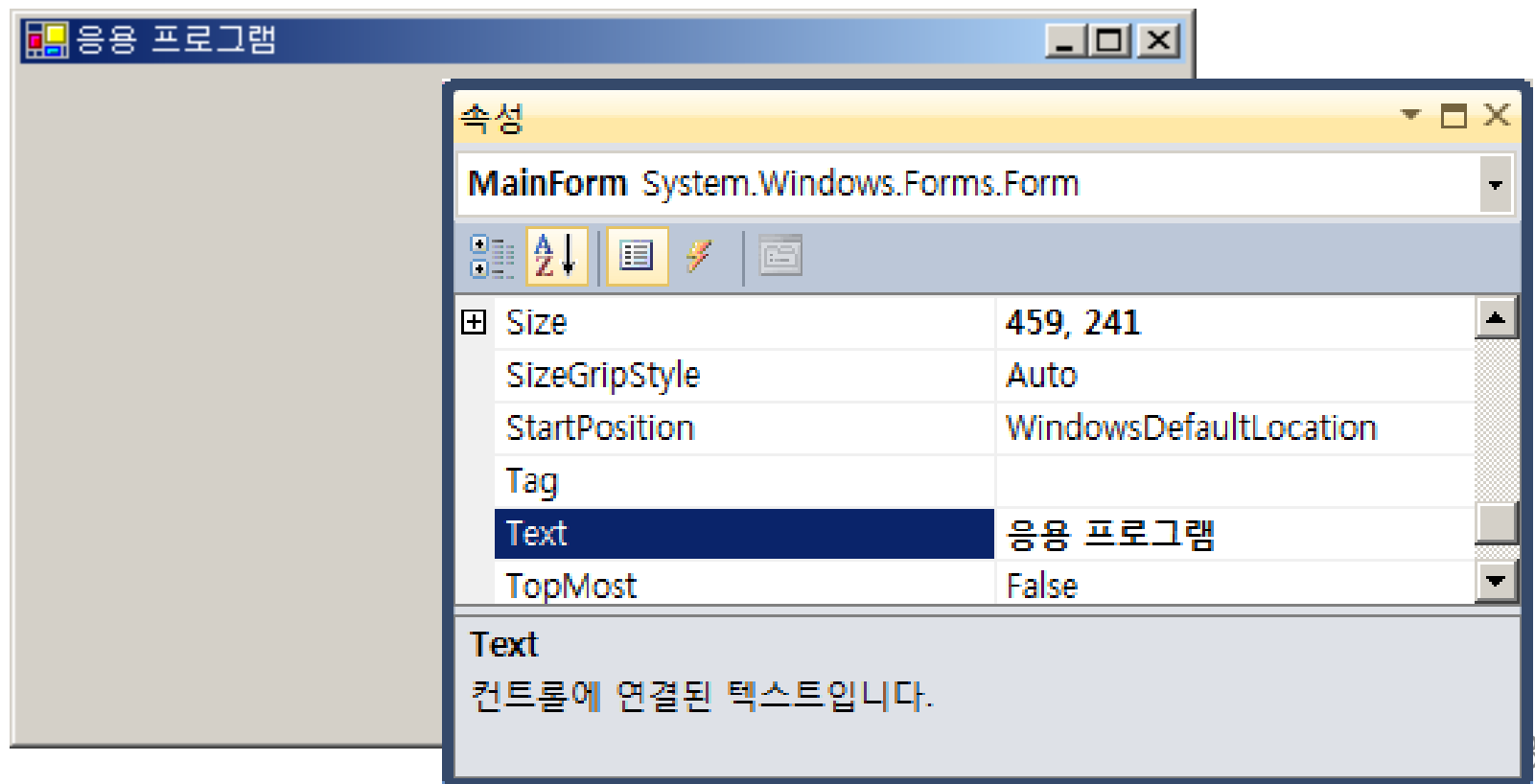
- My3 프로젝트 생성
 - Visual C# | Windows Forms 응용 프로그램
- 파일이름 바꾸기
 - Program.cs → My3.cs
 - Form1.cs → MainForm.cs

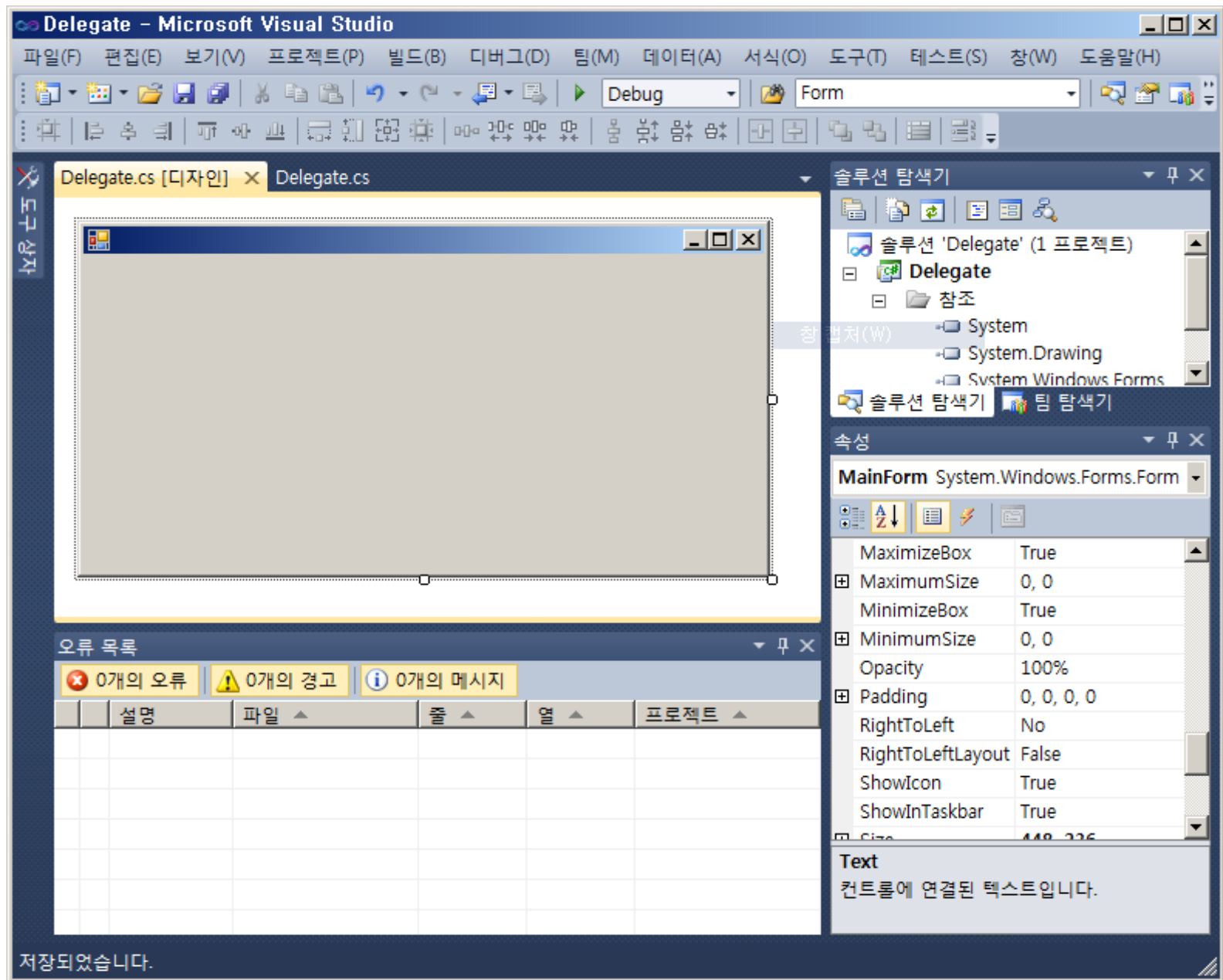
파일 이름 위에서 오른쪽 마우스 버튼 클릭하여 '이름 바꾸기'



02. 폼 윈도우 비주얼 디자인

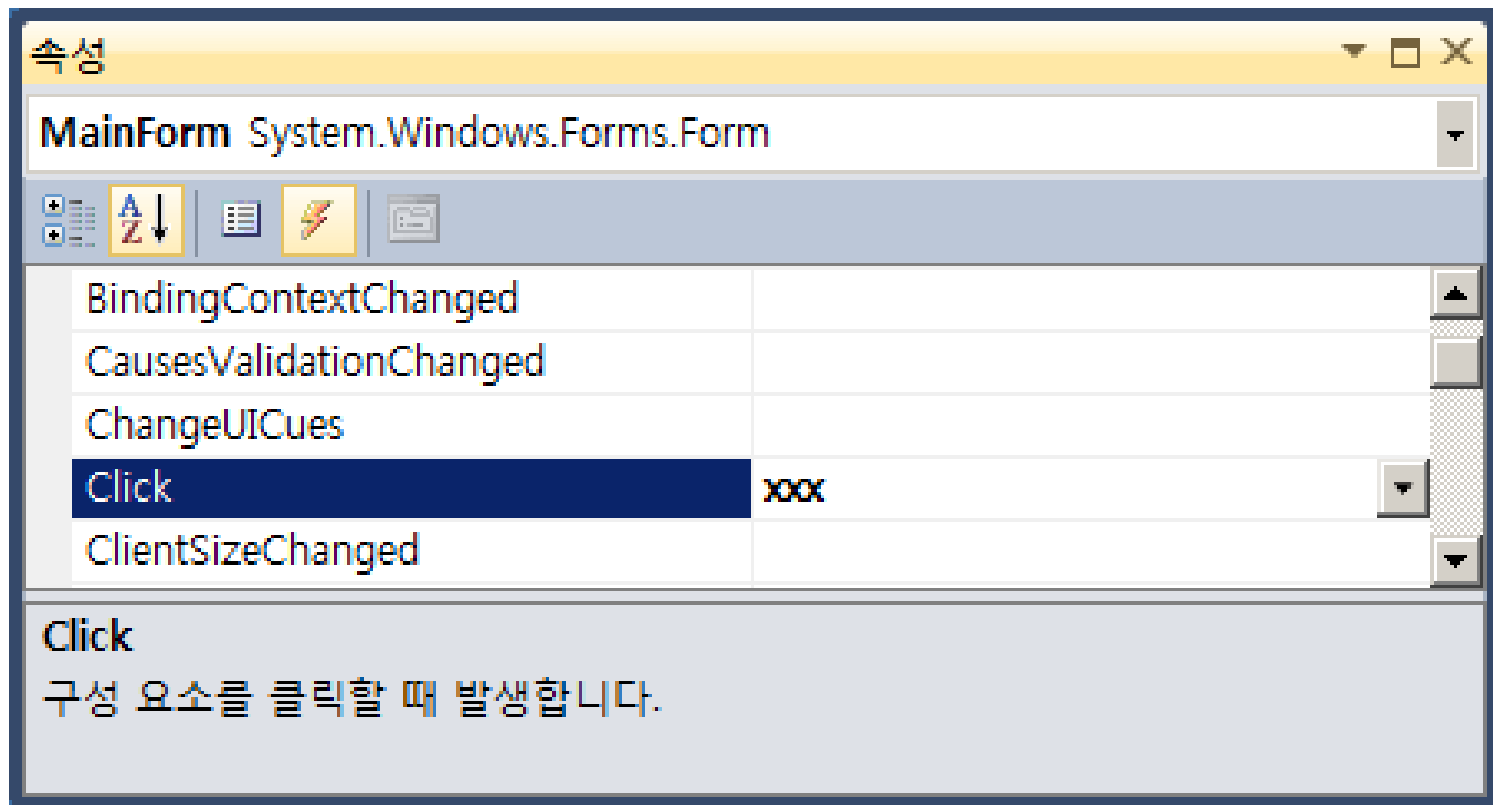
- MainForm.cs 클릭하여 폼 윈도우 크기 변경
- 보기 | 속성 메뉴 항목을 선택하여 속성창 표시
- 타이틀(Text 속성) 변경





03. 핸들러 함수 추가

- Click 이벤트 핸들러 함수 xxx 추가
 - 속성창 | 번개 아이콘 | Click 이벤트 옆에 xxx 기술

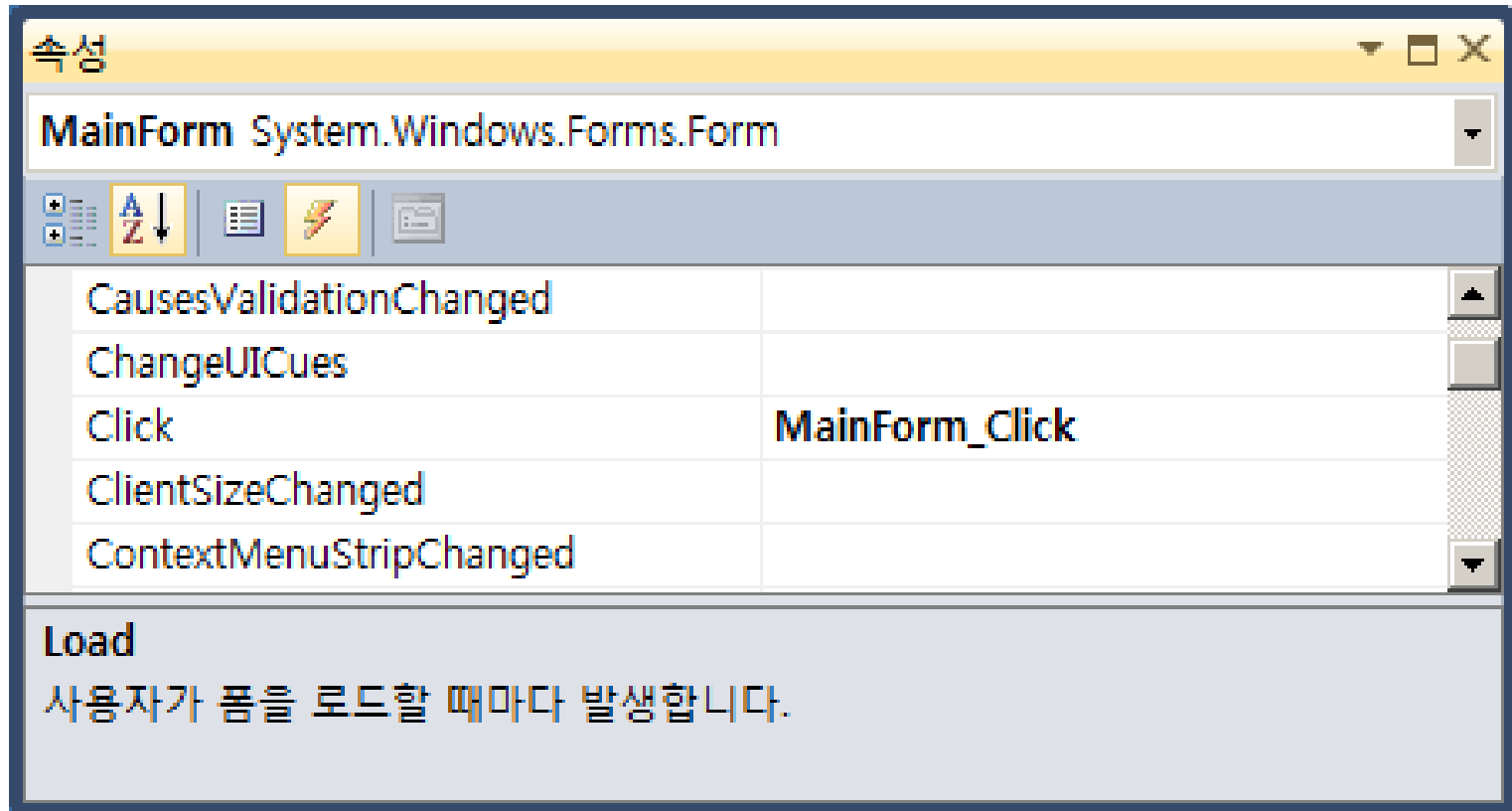


04. 자동으로 생성된 코드 확인

- Click 델리게이트 초기화 코드
 - InitializeComponent 멤버함수 내
- xxx 핸들러 함수 (사용자 코드)


05. 핸들러 함수 이름 변경 연습하기

- 속성 창에서 xxx → MainForm_Click



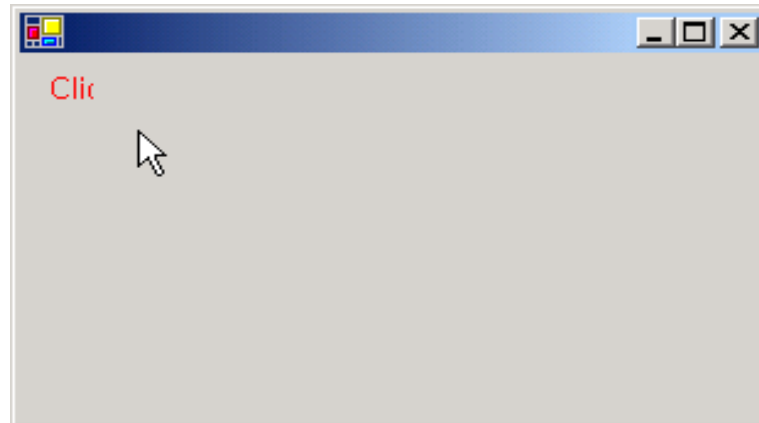
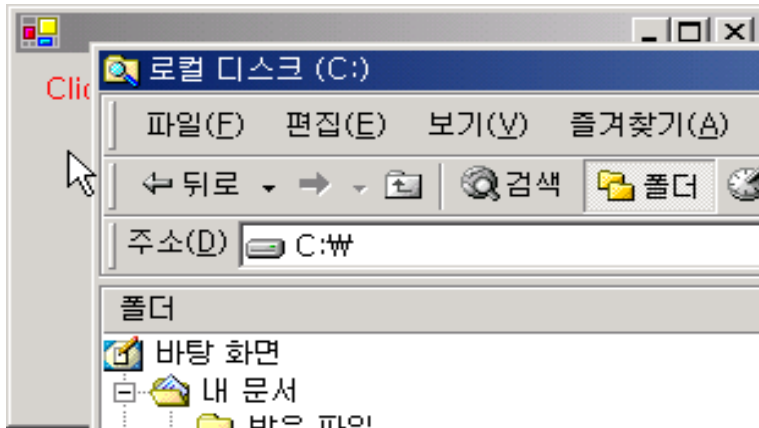
06. 핸들러 함수 코드 작성 연습하기

- 폼 윈도(길동이 얼굴) 클릭할 경우 “Click!” 문자열 출력

```
private void MainForm_Click(object sender, EventArgs e)
{
    
    Graphics cheolsu = this.CreateGraphics();

    cheolsu.DrawString("Click!", Font,
        new SolidBrush(Color.Blue), 10, 10);
}
```


07. 문제점



08. Paint 이벤트와 핸들러 함수 작성

- Paint 이벤트
 - 폼 윈도우가 다시 그려져야 할 때마다 ‘자동으로’ 발생하는 이벤트
- 속성창에서 핸들러 함수 MainForm_Paint 작성 후 코드입력

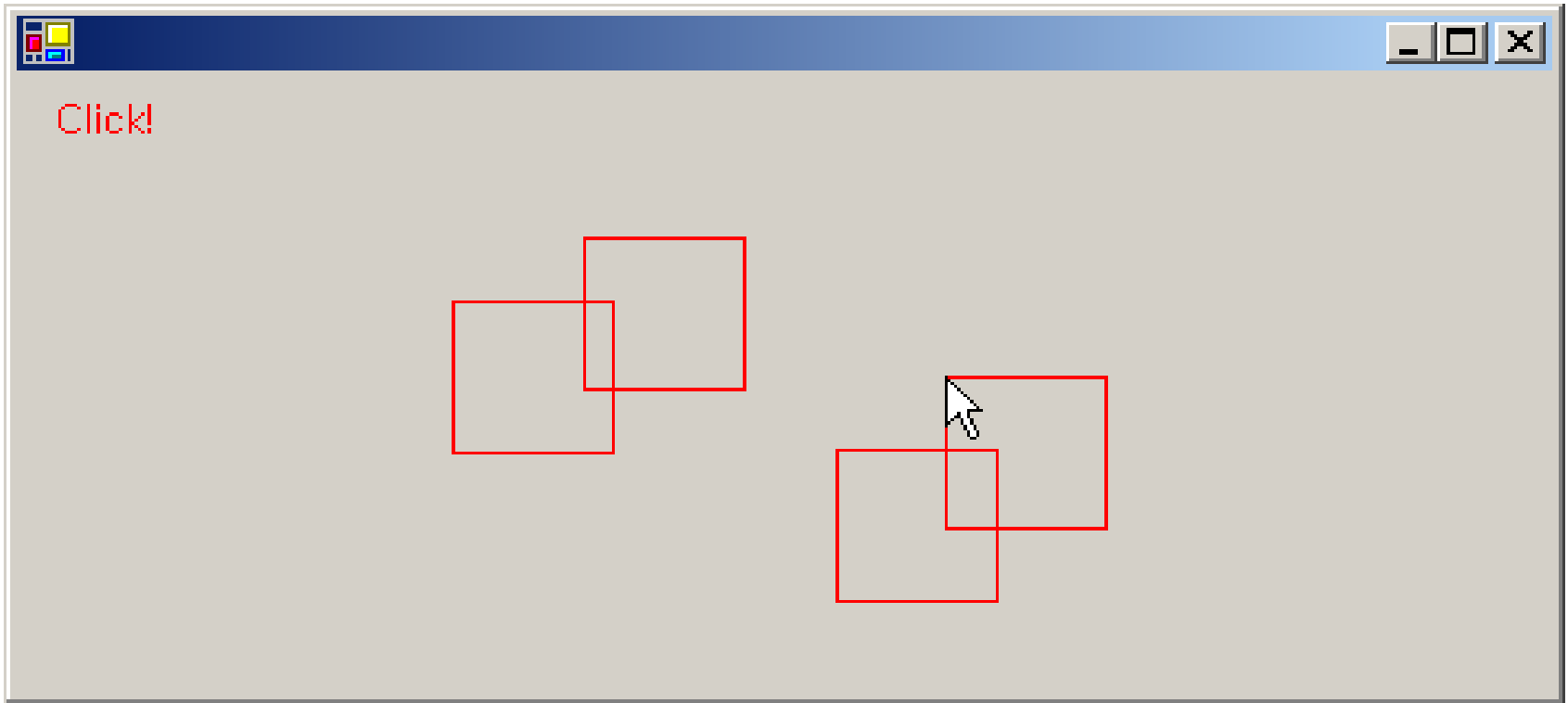
```
Graphics cheolsu = this.CreateGraphics(); //화가
```

```
cheolsu.DrawString("Click!", Font,  
    new SolidBrush(Color.Blue), 10, 10);
```

09. MouseDown 이벤트 핸들러 작성

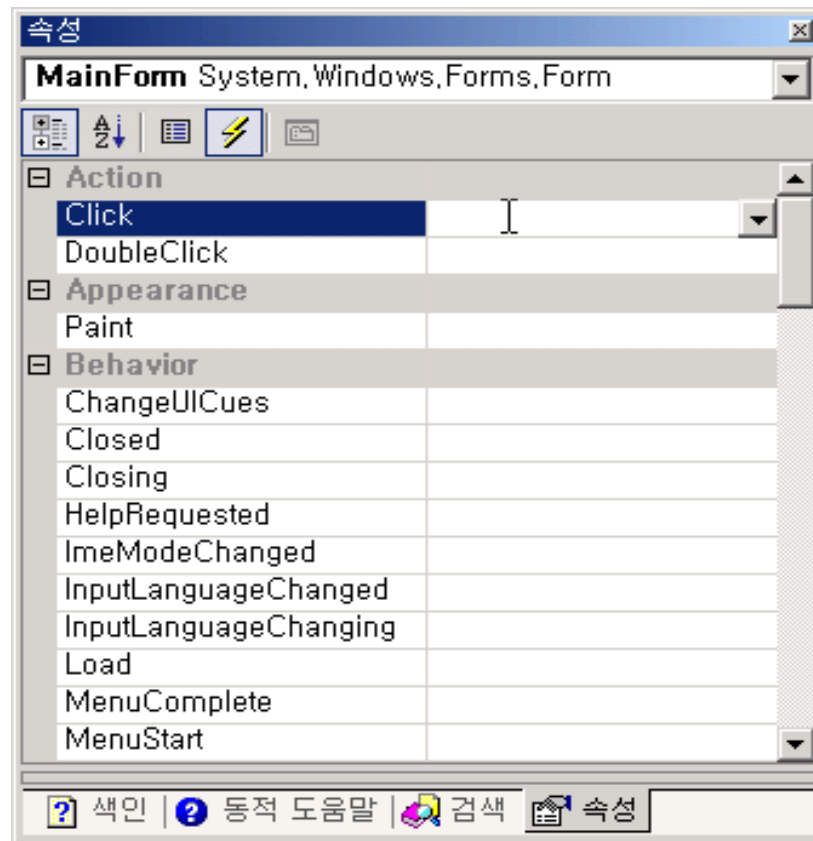
- Click과 마찬가지로 마우스 버튼을 누르면 발생하는 이벤트
 - MouseUp 이벤트는?
- 결국 클릭하면 발생하는 이벤트들은?

```
private void MainForm_MouseDown(object sender,  
MouseEventArgs e)  
{  
    Graphics cheolsu = this.CreateGraphics();  
    cheolsu.DrawRectangle(new Pen(Color.Red), e.X, e.Y, 50, 50);  
}
```



10. 핸들러 함수 제거 연습하기

- 속성 창에서 Click 핸들러 함수 이름 MainForm_Click을 지운 후 Ctrl + S 키를 눌러 저장



10. 핸들러 함수 제거 연습하기

- 하지만,
 - MainForm_Click 핸들러 함수 자체는 제거되지 않음.
- 왜냐하면
 - 혹시나 실수로 힘들게 작성한 코드가 삭제될 수도 있으므로...
- 따라서
 - 핸들러 함수 xxx는 직접 확인하고 삭제하라는 의미

11. 가상함수 오버라이딩 연습하기

- 코드를 직접 입력함

```
protected override void OnClick(EventArgs e)
{
    //base.OnClick(e);
}
```

12. 폼 윈도우에 버튼 추가하기(코딩방법)

```
class MainForm : Form
{
    Button btn;

    public void InitializeComponent()
    {
        btn = new Button();
        btn.Text = "Show Olle";
        btn.Location = new Point(100, 100);
        btn.Size = new Size(100, 30);
        Controls.Add(btn);
    }
}
```

“직접 작성하니 매우
귀찮다!”

13. 드래그-앤-드롭으로 버튼 추가하기

- 비주얼 디자인

“마우스로 디자인 하니
매우 쉽고 편하다!”