

제4장

상속으로 코드 재사용하기

제주대학교 컴퓨터공학과
변영철 교수

1. My2 프로젝트 생성

```
class Point  
{
```

```
    private int iX;  
    private int iY;
```

```
    public void Assign(int x, int y)  
    {  
        iX = x;  
        iY = y;  
    }
```

```
    public int Add()  
    {  
        return iX + iY;  
    }
```

```
}
```

```
class My2  
{
```

```
    static void Main()  
    {
```

```
        Point gildong = new Point();  
        int iResult;
```

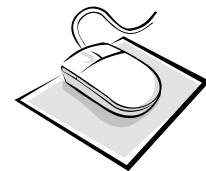
```
        gildong.Assign(2, 3);
```

```
        iResult = gildong.Add();
```

```
        System.Console.WriteLine("두 개의 값을 더한 결과 : " + iResult);
```

```
    }
```

```
}
```



2. 귀찮은 것은 정말 싫은데

- 원 클래스(**Circle**)를 만들어보자!
 - 멤버 변수 : 중심점과 반지름
 - 멤버 함수 : SetRadius, Area
- 코드 중복과 코드 재사용
 - Point 클래스와 Circle 클래스의 코드 중복
 - Point 클래스 코드 재사용 = 상속
- 클래스는...
 - 객체 만들라고 있는 것
 - 재사용하라고 있는 것

3. 관계는 없다

- 코드 재사용 이유
 - 다시 일일이 작성하는 게 귀찮아서
 - 관계를 만들기 위함이 아니다.

4. 보이지 않는 멤버

- 눈에 보이는 멤버만 멤버가 아니다.
 - 보이지 않는 멤버도 있다.
 - 보이지 않는 멤버는 상속받은 것이다.

5. 상속받은 멤버 중 private 멤버

- 중심점 이동 멤버 함수
 - Move 멤버 함수
 - Derived 클래스에서 Base 클래스의 private 멤버 변수를 접근해야 하는데 이는 불가능
- private vs. protected
 - protected는 Derived 클래스에서 접근가능

6. 보이지 않는 멤버는

- 코드는 보이지 않는 멤버는
 - 상속 받은(코드를 재사용하는) 것이다.
- Eclipse 클래스
 - Circle 클래스 재사용
 - 반지름 추가(iRadius2)
 - SetRadius2 멤버 함수

7. 쓸모 없는 코드

- Area 함수 재사용
- 함수 재정의와 new
 - 상속을 받았지만 쓸모 없을 경우 새로(new) 작성
 - 상속을 받았지만 불충분해서, 뭔가 부족해서 새로 작성