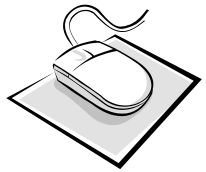
제4장
상속으로 코드 재사용하기

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 (중복해서) 작성하는 게 귀찮아서 재사용
  - 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
  - 파생 클래스(Circle)에서 기반 클래스 (Point)의 private 멤버 변수(iX, iY) 접근 불가
- private vs. protected
  - protected 멤버는 재사용하는(상속받는) 클래스 Circle에서 접근가능

5. 상속받은 멤버 중 private 멤버

- 따라서 protected를 보면 무슨 생각?
  - 아, 이 멤버(iX, iY)는 재사용하는 클래스에서 접근할 수 있도록 하고 있구나.
  - 이 클래스(Point)는 나중에 재사용될 수도 있겠구나.

6. 보이지 않는 멤버는

- 어디 있지? 보이지 않는 멤버는
  - 상속 받은(코드를 재사용하는) 것이다.
- 타원(Ecllipse) 클래스 작성하기
  - Circle 클래스 재사용
  - 반지름 추가(iRadius2)
  - SetRadius2 멤버 함수

7. 쓸모 없는 코드

- 타원에서 원의 Area 함수 재사용
  - 문제 있다!
- 함수 재정의와 new
  - 상속을 받았지만 쓸모 없다. 이 경우 새로(new) 작성
  - 상속을 받았지만 불충분하면, 뭔가 부족하면 새로 작성