제4장
상속으로
코드 재사용하기

변 영 철

1. 가장 간단한 C# 프로그램

- My2 C# 프로젝트
- XXX 클래스, Main 함수

2. 조금 복잡한 프로그램 (Top-Down)

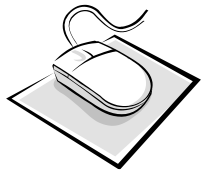
- 두 값을 할당 (Assign)
- 두 값을 표현하는 변수 2개 (iX, iY)
- 두 점을 더하여 반환 (Add)

3. 클래스 분리 및 이름 변경

- Main 함수를 다른 클래스(예, My2App)로 이동
- XXX → Point

4. 전체 코드

```
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 My2App
{
	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);
	}
}
```



5. 귀찮은 것은 정말 싫은데

- 원 클래스(Circle)를 만들어보자.
  - 멤버 변수 : 중심점(iX, iY)과 반지름(iRadius)
  - 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 (새로운 클래스 생성)

6. 관계는 없다

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

7. 보이지 않는 멤버

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

8. 보이지 않는 멤버는

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

9. 쓸모 없는 함수 새로 만들기

- 타원에서 원의 Area 함수 재사용
  - 상속받았지만 쓸모 없는 Area 함수
- Area 새로 정의
- protected를 보면 무슨 생각?
  - 아, 이 멤버(iRadius)는 나중에 재사용할 수 있구나.
  - 재사용하는 클래스에서 이 멤버를 접근하도록 하는구나.

9. 상속받은 함수가 쓸모 없을 때

- new 키워드로 함수 새로 정의하기
  - 걱정하지 마. 상속받은 함수로 Area가 있다는 걸 알고 있어.
  - 그건 그냥 쓸모가 없어서 새로(new) 정의하는 거야.