

제11장

상속과 오버라이딩



1. 상속(inheritance)이란?

■ 상속의 개념

- 기존의 클래스를 재 사용하여 새로운 클래스를 작성하는 것.
- 관계없는 두 개 이상의 클래스를 조상(부모), 자손(자식)으로 직접적 관계를 만들.
- 자손은 조상의 모든 멤버를 상속받음.(단, 생성자, 초기화 블록 제외)
- 자손의 멤버개수가 조상보다 적을 수가 없다.(같거나 많다.) - 다형성 개념 적용

```
public class Bicycle {  
    int id;  
    String brand;  
}
```

```
public class MoutainBike extends Bicycle {  
}
```

많은 상속을 하다 보면,
멤버들이 늘어나는 것
이 당연하다.

다른 클래스 생성의
기반이 된다. 그래서
베이스클래스, 기반
클래스 라고도 부른다.

```
public class MoutainBike {  
    int id;  
    String brand;  
    String frame;  
    int gear;  
    String money;  
}
```

```
public class MoutainBike extends Bicycle {  
    //멤버변수가 5개이다.  
    String frame;  
    int gear;  
    String money;  
}
```

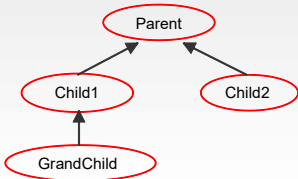
2. 클래스간의 관계(상속)

■ 상속 관계 – 직접적 관계

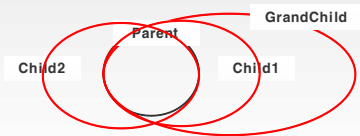
- 공통적으로 들어가는 부분은 조상에서 관리하고, 개별부분은 자손에서 따로 관리함.
- 조상의 변경은 자손의 변경을 일으킨다.(반대는 변경이 없다.)

```
class Parent{}  
class Child1 extends Parent {}  
class Child2 extends Parent {}  
class GrandChild extends Child1 {}
```

* 상속 계층도(상속 관계도)



* 상속 다이어그램



3. 포함 관계(composite)

■ 포함이란? → 다중 상속을 대체하는 방법

- 클래스의 멤버변수로 다른 클래스를 선언하는 것.
- 규모가 작은 클래스를 먼저 만들고, 이것을 조합하여 규모가 큰 클래스로 만들어감.

```
class Point {  
    int x;  
    int y;  
}
```

```
class Circle {  
    int x; // 원점의 x좌표  
    int y; // 원점의 y좌표  
    int r; // 반지름(radius)  
}
```

```
class Circle {  
    Point c = new Point(); // 원점  
    int r; // 반지름(radius)  
}
```

원은 원점과 반지름으로 이루어진다. 하여, 따로 클래스를 만드는 것 보다, 위와 같이 먼저 Point클래스를 만들고, Circle클래스 안에 멤버변수로 사용하는 것이 코드의 재 사용성이 좋으며, 관리하기도 수월하다.

4. 상속이냐? 포함이냐?

■ 이론적 방법 → 절대적인 것은 아니다. 단지 원론적인 얘기라고 생각하자.

- 'is a' 와 'has a' 로 문장을 만들어보자.

ex) Circle is a Point(미 성립)

Circle has a Point(성립)

- 상속 관계 : ~은 ~이다.(is a)

포함 관계 : ~은 ~을 가지고 있다.(has a)

■ 현실적 방법

- 현업에서 개발 시, 만들어질 클래스에 영향을 가장 많이 주는 클래스는 상속하고 보조적인 것은 포함으로 돌려서 작성한다.

```
class Point {  
    int x;  
    int y;  
}
```

```
class Circle extends Point {  
    int r; // 반지름(radius)  
}
```

```
class Circle {  
    Point c = new Point(); // 원점  
    int r; // 반지름(radius)  
}
```

5. 상속과 포함 - 예제

- 원(Circle)은 도형(Shape)이다.(A Circle is a Shape.) : 상속관계
- 원(Circle)은 점(Point)를 가지고 있다.(A Circle has a Point.) : 포함관계

```
class Shape {  
    String color = "blue";  
    void draw() {  
        // 도형을 그린다.  
    }  
}
```

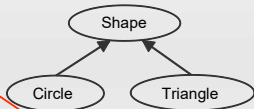
```
class Point {  
    int x;  
    int y;  
  
    Point() {  
        this(0,0);  
    }  
  
    Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
}
```

```
class Circle extends Shape {  
    Point center;  
    int r;  
  
    Circle() {  
        this(new Point(0,0),100);  
    }  
  
    Circle(Point center, int r) {  
        this.center = center;  
        this.r = r;  
    }  
}
```

원점(0,0)과 반지름 100인
인스턴스 생성됨.

```
class Triangle extends Shape {  
    Point[] p;  
  
    Triangle(Point[] p) {  
        this.p = p;  
    }  
  
    Triangle(Point p1, Point p2, Point p3) {  
        p = new Point[] {p1,p2,p3};  
    }  
}
```

```
Circle c1 = new Circle();  
Circle c2 = new Circle(new Point(150,150),50);  
  
Point[] p = {new Point(100,100),  
              new Point(140,50),  
              new Point(200,100)};  
Triangle t1 = new Triangle(p);
```



감사합니다.

