

클래스 상속

01 상속



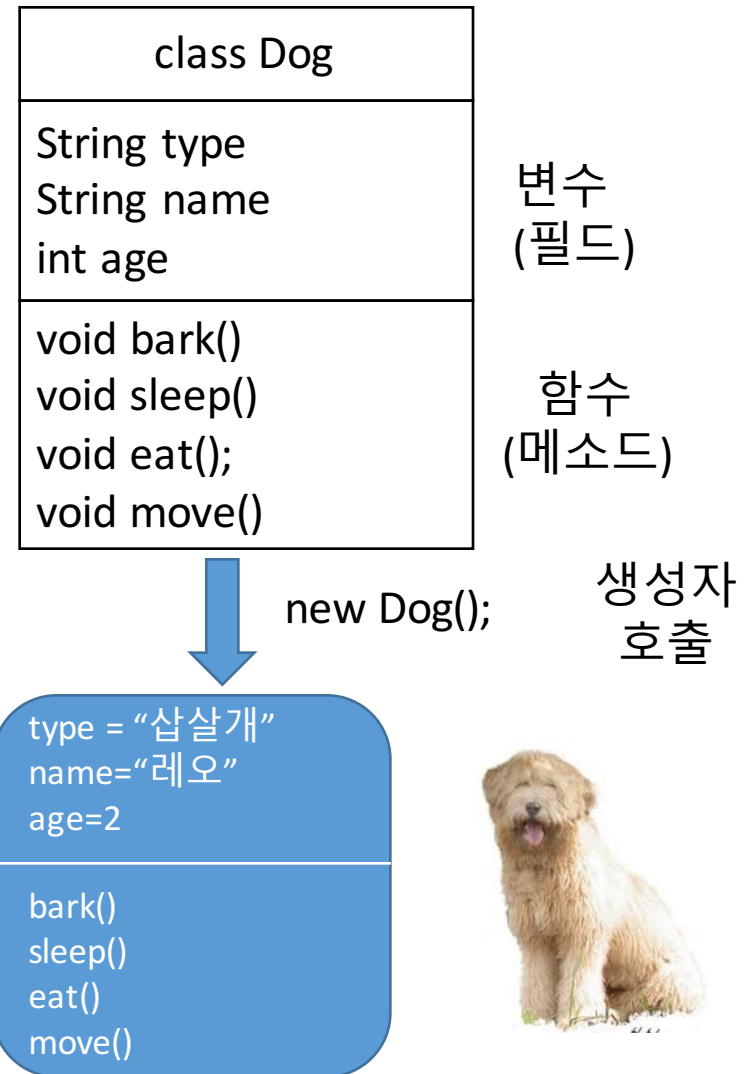
클래스와 객체 (리뷰)

- 클래스

- 관련된 변수 (객체 상태)와 함수(객체 행위)를 하나의 프로그램 단위로 구조화 시킨 것
- 클래스 멤버
 - 멤버 변수(필드)
 - 멤버 함수(메소드)
 - 생성자: 멤버 변수 초기화

- 객체

- 클래스의 한 인스턴스

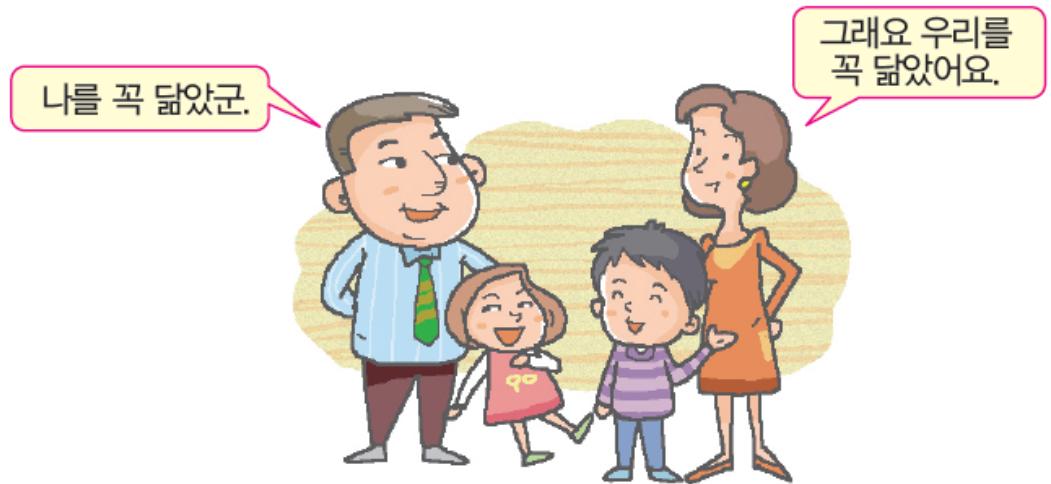


상속 (inheritance)

- 객체 지향 상속
 - 자식이 부모 유전자를 물려 받는 것과 유사한 개념



유산 상속



유전적 상속 : 객체 지향 상속

상속의 필요성

- DogSimulator의 확장



class Dog
-String type -String name -int age
+void bark() +void sleep() +void eat() +void move()



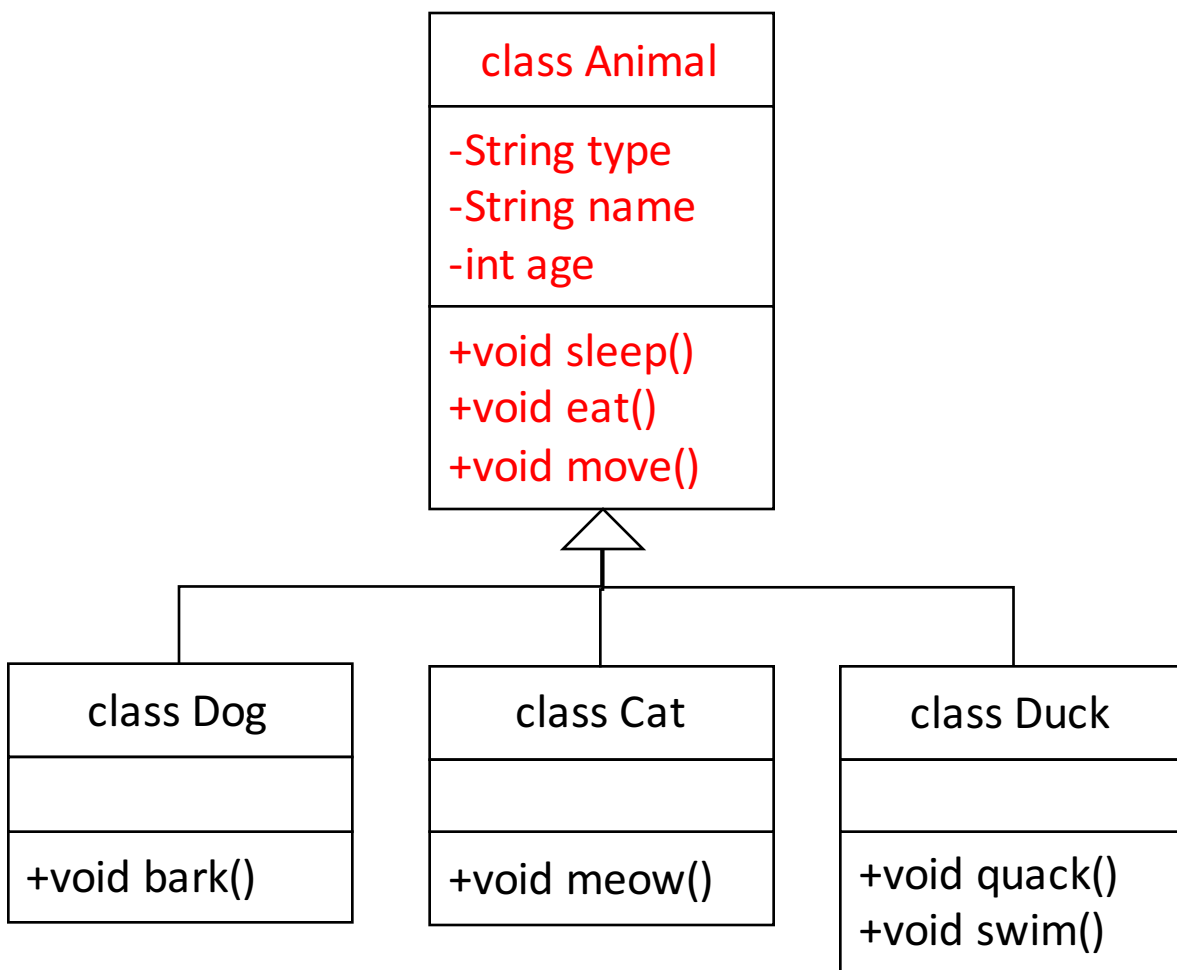
class Cat
-String type -String name -int age
+void meow() +void sleep() +void eat() +void move()



class Duck
-String type -String name -int age
+void quack() +void sleep() +void eat() +void move() +void swim()

클래스
멤버의
중복

상속 계층



서브클래스들의 추상화로
서브클래스들의 공통 멤버를 정의

슈퍼 클래스
(= 부모클래스)

추상화

서브클래스
(= 자식클래스) 구체화

슈퍼 클래스들의 공통 멤버를
상속하며,
서브클래스 고유의 멤버 정의

클래스 상속과 객체

- 상속 선언
 - extends 키워드로 선언
 - 부모 클래스를 물려받아 확장한다는 의미
 - 부모 클래스 -> 슈퍼 클래스(super class)
 - 자식 클래스 -> 서브 클래스(sub class)

```
class Point {  
    int x, y;  
    ...  
}  
  
class ColorPoint extends Point { // Point를 상속받는 ColorPoint 클래스 선언  
    ...  
}
```

서브 클래스

슈퍼 클래스

- ColorPoint는 Point를 물려 받으므로, Point에 선언된 필드와 메소드 선언 필요 없음

예제 : 클래스 상속

(x, y)의 한 점을 표현하는 Point 클래스와 이를 상속받아 점에 색을 추가한 ColorPoint 클래스를 만들고 활용해보자.

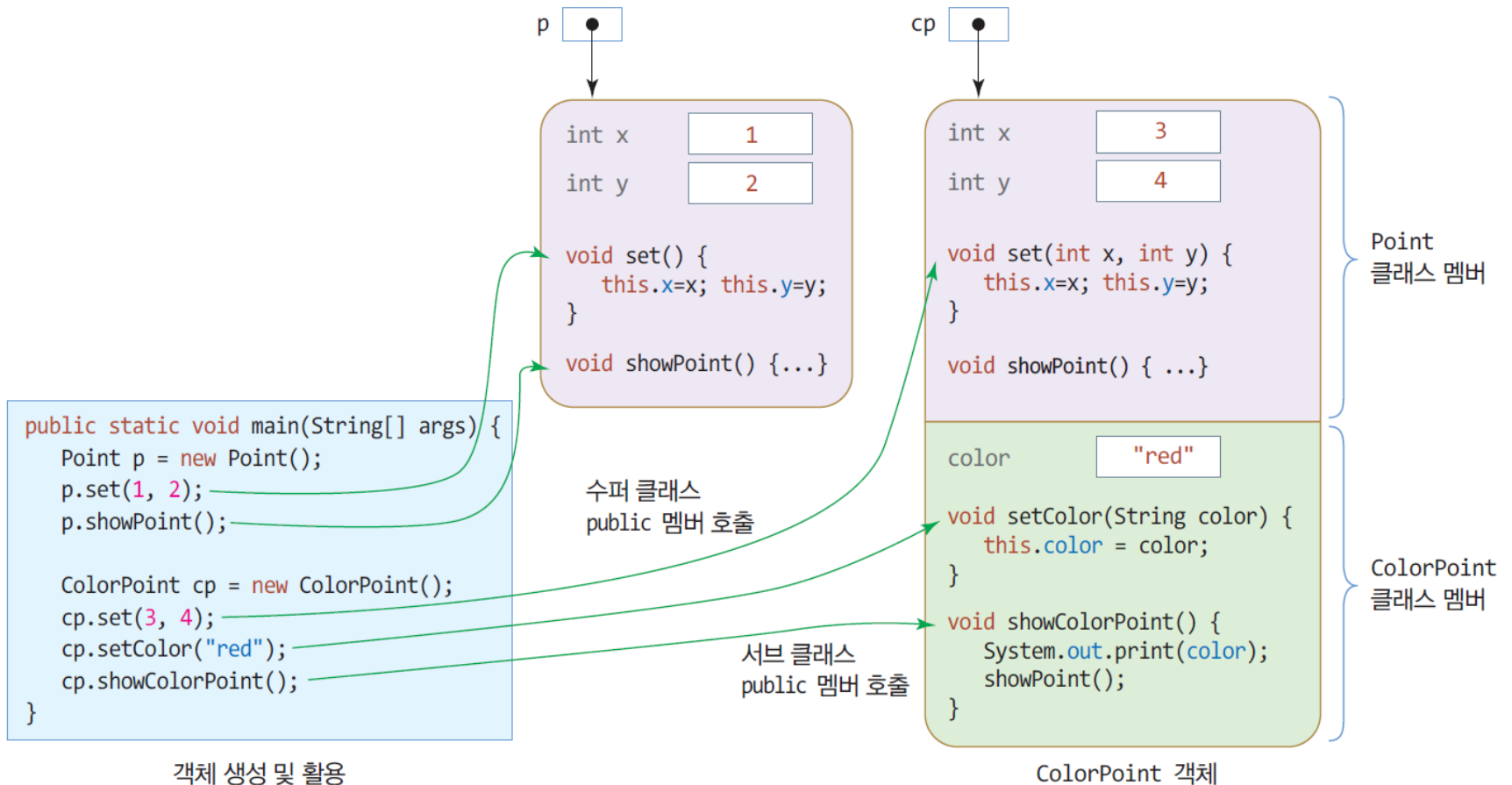
```
class Point {  
    private int x, y; // 한 점을 구성하는 x, y 좌표  
    void set(int x, int y) {  
        this.x = x; this.y = y;  
    }  
    void showPoint() { // 점의 좌표 출력  
        System.out.println("(" + x + ", " + y + ")");  
    }  
}  
  
// Point를 상속받은 ColorPoint 선언  
class ColorPoint extends Point {  
    private String color; // 점의 색  
    void setColor(String color) {  
        this.color = color;  
    }  
    void showColorPoint() { // 컬러 점의 좌표 출력  
        System.out.print(color);  
        showPoint(); // Point의 showPoint() 호출  
    }  
}
```

```
public class ColorPointEx {  
    public static void main(String [] args) {  
        Point p = new Point(); // Point 객체 생성  
        p.set(1, 2);           // Point 클래스의 set() 호출  
        p.showPoint();  
  
        ColorPoint cp = new ColorPoint();  
        cp.set(3, 4);          // Point 클래스의 set() 호출  
        cp.setColor("red");    // ColorPoint의 setColor() 호출  
        cp.showColorPoint();   // 컬러와 좌표 출력  
    }  
}
```

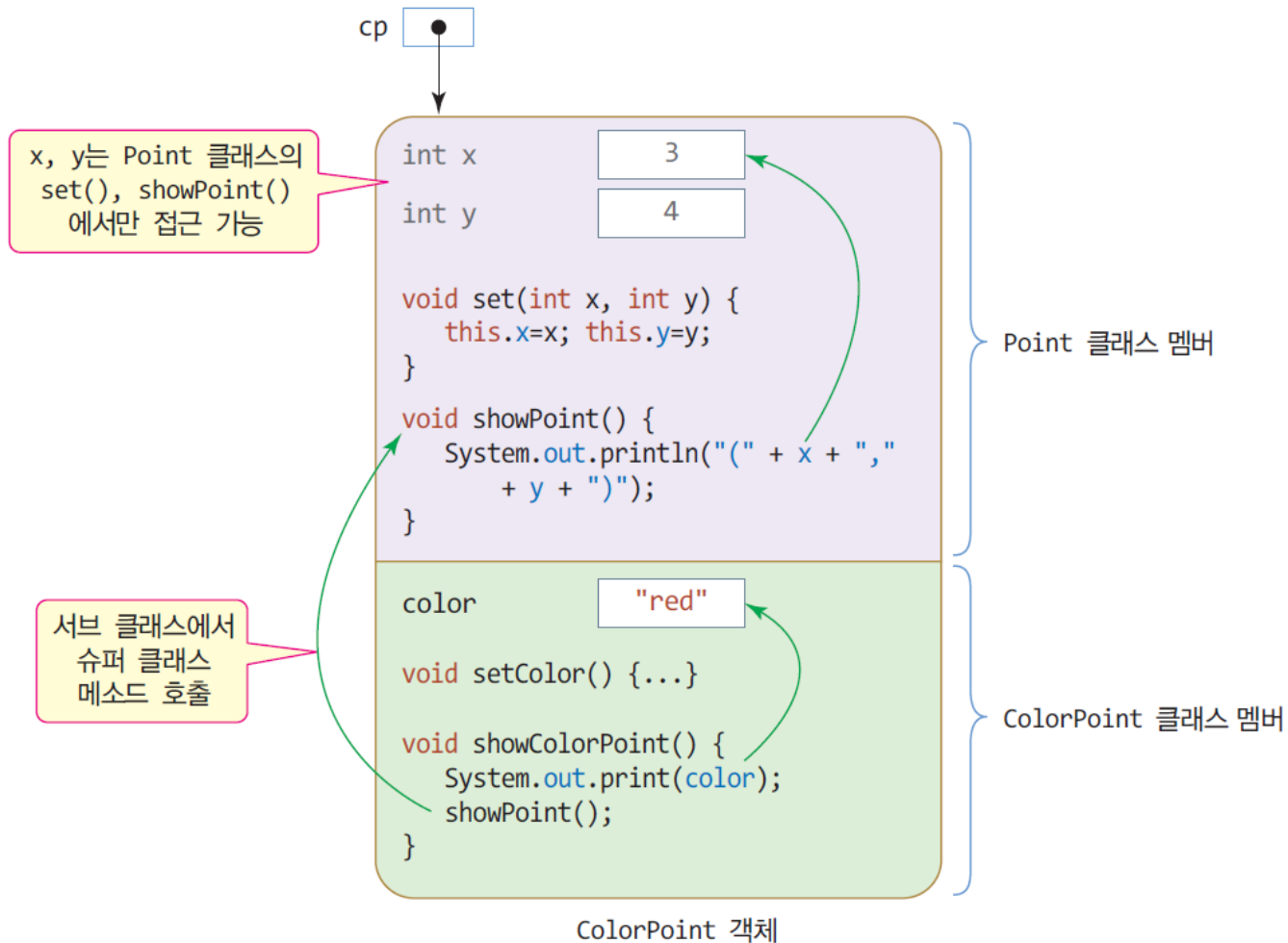
(1,2)
red(3,4)

서브 클래스 객체의 모양

- 슈퍼 클래스 객체와 서브 클래스의 객체는 별개
- 서브 클래스 객체는 슈퍼 클래스 멤버 포함



서브 클래스에서 슈퍼 클래스 멤버 접근



[실습1]

- 원을 표현하는 다음 Circle 클래스가 있다.

```
public class Circle {  
    private int radius;  
    public void setRadius(int radius) { this.radius = radius; }  
    public void showCircle() { System.out.println("반지름= " + radius); }  
}
```

- Circle 클래스를 상속받은 NamedCircle 클래스를 작성하여, 다음 main()을 실행할 때, 다음 실행결과와 같이 출력되도록 하라.

```
public class CircleTest {  
    public static void main(String[] args) {  
        NamedCircle w = new NamedCircle();  
        w.setRadius(5);  
        w.setName("Waffle");  
        w.showNamedCircle();  
    }  
}
```

실행결과

Waffle,반지름= 5