

컴퓨터 프로그래밍1

실습 13주차

- 상속 -

상속 (1/8)

상속

- 이미 존재하는 클래스로부터 멤버들을 물려받는 것

```
class SubClass extends SuperClass
{
    // 추가된 메소드와 필드
}
```

상속을 의미한다. SuperClass를 확장하여 SubClass를 작성한다는 의미이다.

수퍼 클래스 == 부모 클래스 == 베이스 클래스
서브 클래스 == 자식 클래스 == 파생된 클래스

상속 (2/8)

상속

■ 프로그래밍 예제

```
class Car {  
    // 3개의 필드 선언  
    public int speed;        // 속도  
    public int gear;         // 주행거리  
    public String color;     // 색상  
  
    // 3개의 메소드 선언  
    public void setGear(int newGear) { // 기어 설정 메소드  
        gear = newGear;  
    }  
    public void speedUp(int increment) { // 속도 증가 메소드  
        speed += increment;  
    }  
    public void speedDown(int decrement) { // 속도 감소 메소드  
        speed -= decrement;  
    }  
}  
  
class SportsCar extends Car { // Car를 상속받는다.  
    // 1개의 필드를 추가  
    boolean turbo;           // 추가된 필드  
  
    // 1개의 메소드를 추가  
    public void setTurbo(boolean newValue) { // 터보 모드 설정 메소드  
        turbo = newValue;  
    }  
}
```

상속 (3/8)

상속

■ 프로그래밍 예제

```
public class CarTest {  
    public static void main(String[] args) {  
        SportsCar c = new SportsCar();  
        c.color = "Red";  
        c.setGear(3);  
        c.speedUp(100);  
        c.speedDown(30);  
        c.setTurbo(true);  
    }  
}
```

서브 클래스 객체 생성

수퍼 클래스의
필드와 메소드
사용

추가된 메소드 사용

상속 (4/8)

접근 지정자

- 서브클래스는 슈퍼클래스의 전용멤버에 접근할 수 없다
- 슈퍼클래스의 보호멤버는 서브클래스만 접근할 수 있다

```
class Employee {  
    public String name;    // 이름  
    private int RRN;       // 주민등록번호  
    protected int salary; // 월급  
  
    protected int getSalary() {  
        return salary;  
    }  
  
    protected void setSalary(int salary) {  
        this.salary = salary;  
    }  
}
```

수퍼 클래스의 private 멤버는
서브클래스에서 접근할 수 없다.

```
public class Manager extends Employee {  
    private int bonus;  
  
    public void printSalary() {  
        System.out.println(name+":"+ (salary + bonus));  
    }  
  
    private void printRRN() {  
        System.out.println(RRN);  
    }  
}
```

상속 (5/8)

메소드 재정의(overriding)

- 메소드 재정의란 서브클래스가 필요에 따라 상속된 메소드를 다시 정의하여 사용하는 것을 의미한다

```
class Animal {  
    public void sound() {  
        // 아직 특정한 동물이 지정되지 않았으므로 몸체  
    }  
}
```

```
class Dog extends Animal {  
    public void sound() {  
        System.out.println("멍멍!");  
    }  
}
```

```
public class DogTest {  
    public static void main(String[] args) {  
        Dog d = new Dog();  
        d.sound();  
    }  
}
```

메소드 재정의

멍멍!

재정의된 메소드가 호출된다.

상속 (6/8)

super 키워드

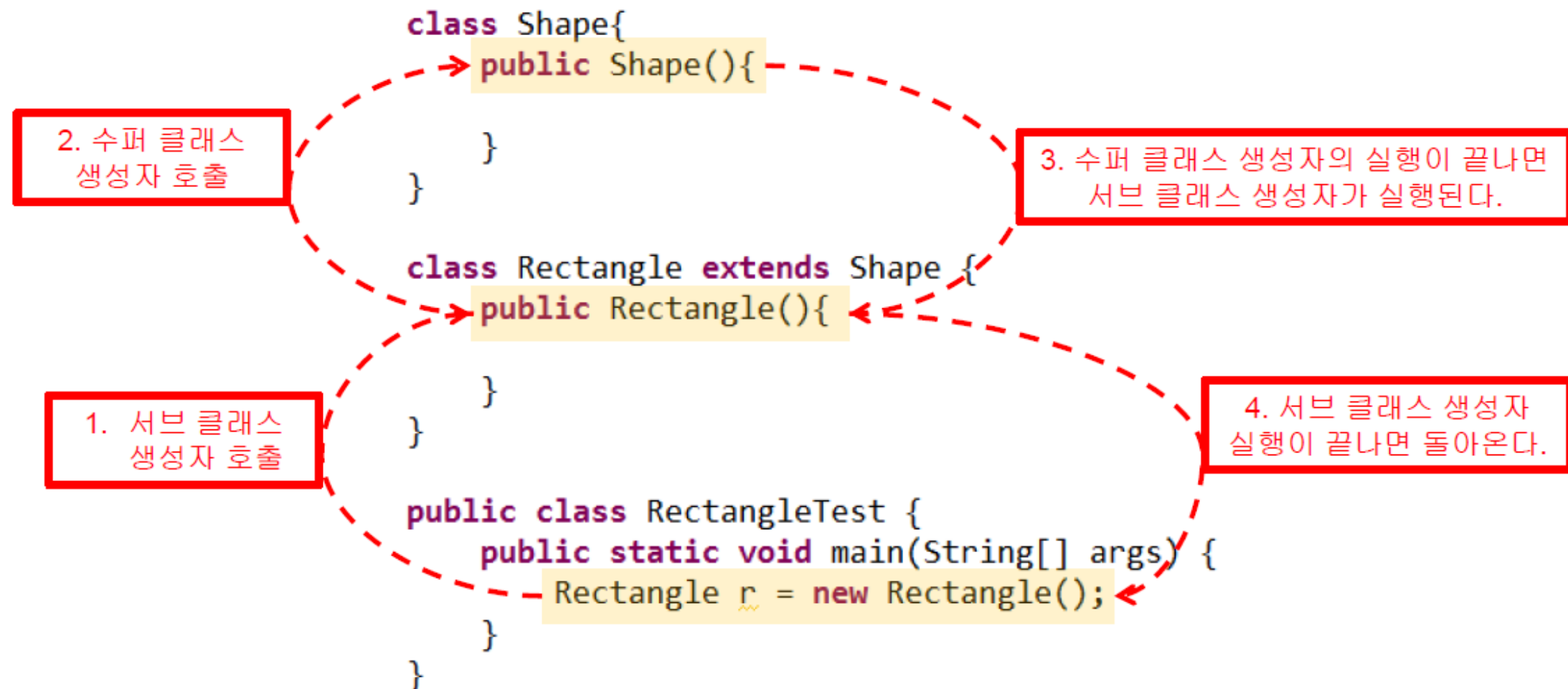
- 수퍼클래스의 멤버를 명시적으로 참조하기 위하여 사용한다

```
class ParentClass{
    int data=100;
    public void print(){
        System.out.println("수퍼 클래스의 print() 메소드");
    }
}
class ChildClass extends ParentClass {
    int data=200;
    public void print(){
        super.print();
        System.out.println("서브 클래스의 print() 메소드");
        System.out.println(this.data);
        System.out.println(super.data);
    }
    public static void main(String[] args){
        ChildClass obj = new ChildClass();
        obj.print();
    }
}
```

수퍼 클래스의 print() 메소드
서브 클래스의 print() 메소드
200
100

상속 (7/8)

상속과 생성자



상속 (8/8)

프로그래밍 예제

```
class SuperClass{  
    public SuperClass(){  
        System.out.println("Super Class 생성!");  
    }  
}
```

```
class SubClass extends SuperClass{  
    public SubClass(){  
        System.out.println("Sub Class 생성!");  
    }  
}
```

```
public static void main(String[] args) {  
    SubClass sc = new SubClass();  
}
```

terminated: test.java:Application.java

Super Class 생성!

Sub Class 생성!

실습

▮ 프로그래밍 실습 – p.264

- 일반적인 책을 나타내는 Book클래스를 상속받아서 잡지를 나타내는 Magazine 클래스를 작성하라
 - Book클래스는 제목, 페이지수, 저자등의 정보를 가진다.
 - Magazine클래스는 추가로 발매일 정보를 가진다
 - 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성한다.
 - 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라

과제

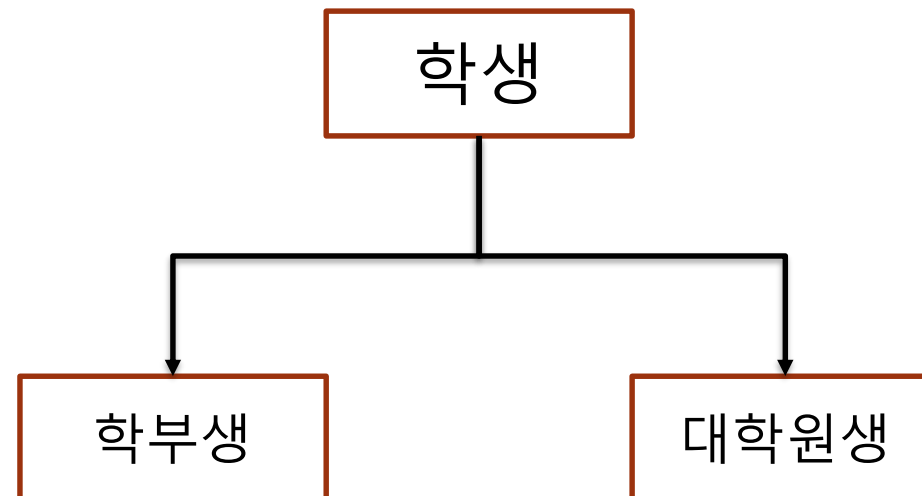
■ 과제 1. – p.264

- Person 클래스를 설계하라.
- Person 클래스는 이름, 주소, 전화번호를 필드로 가진다.
- 하나 이상의 생성자를 정의하고 각 필드에 대하여 접근자와 설정자 메소드를 작성하라.
- 이어서 Person을 상속받아서 Customer를 작성하라.
- Customer는 고객 번호와 마일리지를 필드로 가지고 있다.
- 한 개 이상의 생성자를 작성하고 적절한 접근자 메소드와 설정자 메소드를 작성하라.
- 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.

과제

과제 2. – p.265 7번 그림 참고

- 모든 학생은 이름, 학번, 소속 학과, 학년, 이수학점 수를 가진다.
- 추가적으로 학부생은 소속 동아리명을 가진다.
- 대학원생은 조교 유형과 장학금 비율을 가진다.
 - 조교 유형에는 교육 조교와 연구 조교가 있다.
 - 장학금 비율은 0과 1 사이의 값이다.
- 각 클래스는 적절한 생성자 메소드, 접근자 메소드, 변경자 메소드를 가진다.
- 이러한 클래스의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.



Q&A



수고하셨습니다