

제 07 장 상속(실습)

☐ 개념 확인 학습

1. 부모 클래스의 필드와 메소드를 자식 클래스에서 사용할 수 있도록 하는 자바의 기능은 무엇입니까?
2. 상속관계에서 부모 클래스의 메소드를 자식 클래스에서 다시 정의하는 것을 의미하는 용어는 무엇입니까?
3. 클래스, 메소드, 필드에 `final` 키워드가 붙었을 때 각각의 의미는 무엇입니까?
4. 자바에서 자동 타입변환이 이루어질 때는 어떠한 경우입니까?
5. 객체가 어떤 타입 인지 조사할 때 사용하는 연산자는 무엇입니까?
6. 다형성의 의미를 설명하세요.
7. `abstract` 클래스는 어느 경우에 사용하나요?
8. `abstract` 메소드의 형태와 사용법을 설명하세요.

☐ 적용 확인 학습 & 응용 프로그래밍

part1 기본문제 : 1~9

part2 프로그래밍문제 : 10~18

1. `protected` 클래스에 있는 메소드를 그 클래스 안에서만 사용하려면 어떤 접근제어 지시어를 붙여야 하나요?

① `static` ② `final` ③ `private` ④ `protected`

2. 다음 프로그램의 출력을 쓰세요.

```
class Animal {
    String name;
    public Animal() {
        name = "UNKNOWN";
        System.out.println("동물입니다:" + name);
    }
    public Animal(String name) {
        this.name = name;
        System.out.println("동물입니다:" + name);
    }
}
class Lion extends Animal {
    public Lion() { System.out.println("사자입니다."); }
    public Lion(String name) {
        super(name);
        System.out.println("사자입니다.");
    }
}
public class Test {
    public static void main(String[] args) {
        Lion lion=new Lion("Brave");
        Lion lion2=new Lion();
    }
}
```

3. 다음 소스에서 오류를 찾아보세요. 오류가 있다면 소스를 수정하여 해결하세요.

```
public class Bike {
    private int gear;
    public int speed;
}
public class MountainBike expands Bike {
    public int seatHeight;

    public void MountainBike(int g) {
        super();
        gear=g;
    }
}
```

4. 다음과 같은 클래스 정의에 대하여 답하세요.

```
class Student {
    protected int number;
    public String name;
}
class GraduateStudent extends Student {
    private String tel;
    public String lab;
}
public class Test {
    public static void main(String[] args) {
        GraduateStudent s = new GraduateStudent();
    }
}
```

(1) 위의 코드에서 수퍼 클래스는 _____이고 서브 클래스는 _____이다.

(2) 서브 클래스의 생성자에서 수퍼 클래스의 생성자를 명시적으로 호출하도록 생성자를 추가하세요.

(3) `GraduateStudent s = new GraduateStudent();`와 같이 객체를 생성했을 때, 다음 중 필드를 잘못 접근한 것은 무엇입니까?

```
s.number = 10;
s.name = "홍길동";
s.tel = "010-1234-5678";
s.lab = "Java Lab";
```

5. 다음 프로그램의 실행 결과는 무엇입니까?

```
class A {
    public A() { System.out.println("1"); }
    public A(int x) { System.out.println("2"); }
}
class B extends A {
}
public class Test {
    public static void main (String args []) {
        B b = new B();
        System.out.println("실행 완료");
    }
}
```

6. 다음 프로그램의 실행 결과는 무엇입니까?

```
class A {
    public A() { System.out.println("1"); }
    public A(int x) { System.out.println("2"); }
}
class B extends A {
}
public class Test {
    public static void main (String args []) {
        A a = new B();
        System.out.println("실행 완료");
    }
}
```

7. 다음과 같은 클래스 정의에서 질문에 답하세요.

```
class ClassA {
    public void methodOne(int i) { }
    public void methodTwo(int i) { System.out.println("A2"); }
    public static void methodThree(int i) { }
    public static void methodFour(int i) { System.out.println("A4"); }
}
class ClassB extends ClassA {
    public static void methodOne(int i) { }
    public void methodTwo(int i) { System.out.println("B2"); }
    public void methodThree(int i) { }
    public static void methodFour(int i) { System.out.println("B4"); }
}
public class Test {
    public static void main(String args[]) {
        ClassA aa = new ClassA();
        ClassB bb = new ClassB();
        ClassA ab = new ClassB();

        ClassB.methodFour(0);
        ClassA.methodFour(0);
        aa.methodTwo(0);
        bb.methodTwo(0);
        ab.methodTwo(0);
    }
}
```

(1) 어떤 메소드가 수퍼클래스의 메소드를 재정의하고 있나요? (힌트) 정적 메소드로 인스턴스 메소드를 재정의할 수 없다.

(2) 어떤 메소드가 수퍼클래스의 메소드를 가리고 있나요? (힌트) 서브클래스에서 똑같은 정적 메소드를 정의하는 경우, 가린다고(hide) 한다. 이 경우, 서브클래스 객체에서 호출되면 서브클래스의 정적메소드가, 수퍼클래스 객체에서 호출되면 수퍼클래스의 정적메소드가 호출된다.

(3) 컴파일 오류를 지적하세요. 실제로 컴파일해보아도 좋습니다.

(4) 컴파일 오류 수정 후 프로그램의 출력은 어떻게 되나요?

8. 다음 프로그램의 출력을 쓰세요.

```
class A {
    int val=3;
    public void f(A a) { System.out.print("Af(a)"); }
    public void f(B b) { System.out.print("Af(b)"); }
}
class B extends A {
    int val=5;
    public void f(A a) { System.out.print("Bf(a)"); }
    public void f(B b) { System.out.print("Bf(b)"); }
}
public class Test {
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        A ab = (A) b; //A ab = new B();

        System.out.println("a.val = " + a.val);
        System.out.println("b.val = " + b.val);
        System.out.println("ab.val = " + ab.val);

        System.out.print("\n a.f(a)="); a.f(a);
        System.out.print("\n a.f(b)="); a.f(b);

        System.out.print("\n b.f(a)="); b.f(a);
        System.out.print("\n b.f(b)="); b.f(b);

        System.out.print("\n a.f(ab)="); a.f(ab);
        System.out.print("\n b.f(ab)="); b.f(ab);

        System.out.print("\n ab.f(a)="); ab.f(a);
        System.out.print("\n ab.f(b)="); ab.f(b);
        System.out.print("\n ab.f(ab)="); ab.f(ab);
    }
}
```

9. 다음 프로그램의 출력은 무엇인가요?

```
class A {
    final public int myMethod(int a, int b) { return 0; }
}
class B extends A {
    public int myMethod(int a, int b) { return 1; }
}
public class Test {
    public static void main(String args[]) {
        B b = new B();
        System.out.println("x = " + b.myMethod(0, 1));
    }
}
```

10. 오른쪽 URL 그림과 같은 클래스 상속 관계를 가정합니다.

(1) 다음과 같은 문장은 적법한가요? 그 이유는 무엇인가요?

```
Point2D p = new Point3D();
```

(2) 다음과 같은 문장은 적법한가요?

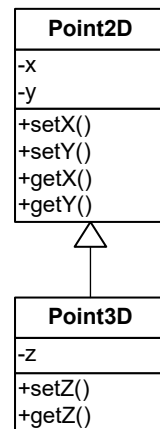
만약 적법하지 않다면 적법하도록 고쳐보세요.

```
p.setX(100);
p.setZ(40);
```

(3) 다음과 같은 문장은 적법한가요? 그 이유는 무엇인가요?

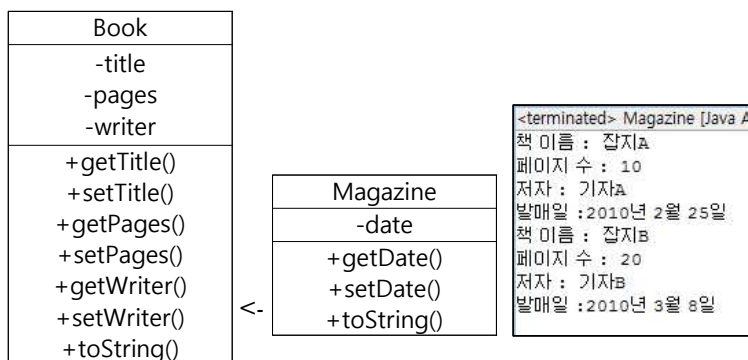
```
Point3D p = new Point2D();
```

(4) Point2D와 Point3D 클래스를 실제로 작성하여 보세요.



11. Person 클래스를 설계합니다. Person 클래스는 이름, 주소, 전화번호를 필드로 가집니다. 하나 이상의 생성자를 정의하고 각 필드에 대하여 접근자/설정자 메소드를 작성하세요. Person 클래스를 상속 받아 Customer 클래스를 작성합니다. Customer는 고객 번호와 마일리지를 필드로 가집니다. 한 개 이상의 생성자를 작성하고 적절한 접근자/설정자 메소드를 작성하세요. 테스트 클래스를 작성하여 위 클래스 타입의 객체를 만들고 각 객체의 모든 정보를 출력하세요.

12. 일반적인 책을 나타내는 Book 클래스를 상속받아서 잡지를 나타내는 Magazine 클래스를 작성합니다. Book 클래스는 제목, 페이지수, 저자 등의 정보를 가집니다. Magazine 클래스는 추가로 발매일 정보를 가집니다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성하세요. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하세요.

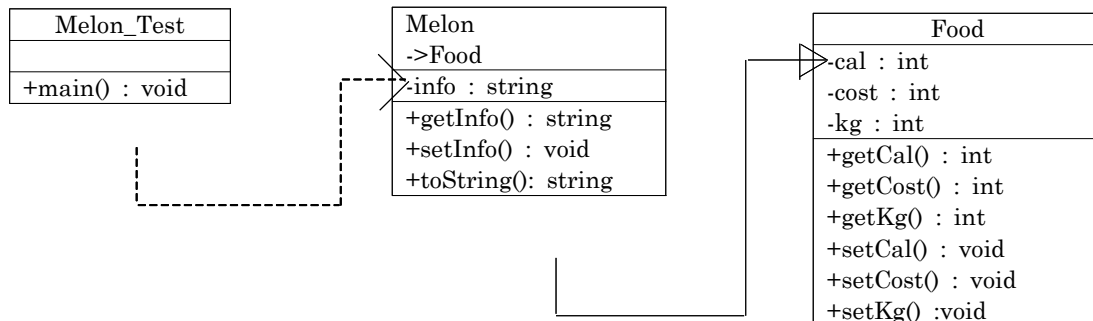


```

public class TestMagazine {
    public static void main(String[] args) {
        Magazine magA = new Magazine("잡지A",10,"기자A","2010년 2월 25일");
        Magazine magB = new Magazine("잡지B",20,"기자B","2010년 3월 8일");

        System.out.println(magA.toString());
        System.out.println(magB.toString());
    }
}
  
```

13. 일반적인 음식을 나타내는 **Food** 클래스를 상속받아서 멜론을 나타내는 **Melon** 클래스를 작성합니다. **Food** 클래스는 칼로리, 가격, 중량 등의 정보를 가집니다. **Melon** 클래스는 추가로 경작농원정보를 가집니다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성합니다. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스도 작성합니다.



14. 일반적인 휴대폰을 나타내는 **Phone** 클래스를 작성합니다. **Phone**에는 제조사maker, 가격price, 통신타입type(2G 또는 3G) 등의 정보가 저장되어 있습니다. **Phone**을 상속받아 **SmartPhone** 클래스를 작성합니다. **SmartPhone** 클래스에는 내부 메모리 크기memory, 블루투스지원여부hasBluetooth 등의 필드가 추가됩니다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성하세요. **PhoneTest** 클래스에서 이들 클래스들의 객체 배열을 만들고 각 객체의 모든 정보를 출력하세요.

15. abstract 메소드 sound()를 가지고 있는 abstract 클래스 Bird를 작성하고 Bird를 상속받아서 Dove 클래스를 작성하세요. Dove 클래스의 sound()에서는 “coo coo”를 출력합니다.


16. 슈퍼 클래스인 Shape에 도형의 면적(double getArea())을 계산하는 메소드를 제공합니다. 삼각형을 나타내는 클래스 Triangle은 Shape을 상속받아 작성합니다. 삼각형에 맞도록 면적(x=밑변의 길이, y=높이)을 계산하는 메소드를 재정의 합니다. 삼각형의 x, y는 생성자에서 저장합니다. 클래스 Circle도 Shape을 상속받아 작성합니다. 원에 맞도록 면적(x=반지름, y=3.14)을 계산하는 메소드를 재정의 합니다. 원의 x, y는 생성자에서 저장합니다. Test 클래스를 작성하여서 삼각형, 원 객체를 생성하고 다형성을 이용하여 각 객체의 정보와 면적을 출력하세요. (힌트 abstract class 사용)

17. 다음 조건에 따라 클래스를 작성하고 테스트 하세요.

- Fruit class : me() 메소드에서 “나는 과일” 리턴
- Fruit 상속 받은 Apple class : me() 메소드에서 “나는 사과” 리턴
- Fruit 상속 받은 Grape class : me() 메소드에서 “나는 포도” 리턴
- Fruit 상속 받은 Pear class : me() 메소드에서 “나는 배” 리턴
- Test class : main() 메소드에서 Fruit 타입의 배열을 생성해 Fruit, Apple, Grape, Pear 객체를 저장하고 각각의 me()메소드 호출하여 결과를 출력

18. 다음 문장을 잘 읽고 조건에 따라 클래스를 작성하고 테스트 하세요.

- 우리 집은 가족(Family)이 4명(dad, mom, boy, girl)이고, 엄마가 추석 선물로 모든 가족에게 핸드폰(Phone)을 사 주었습니다. 엄마가 구매한 핸드폰은 잘 동작(operation) 했지만 통화(call)만 되는 폰입니다.
- 그런데 딸은 이 핸드폰으로 남친의 얼굴을 보며 영상통화를 할 수 없다고 안드로이드폰(AndrPhone)으로 교환(change)했습니다.
- 또한 사춘기 아들은 자신의 핸드폰을 아무렇게나 사용하다가 액정이 깨져 새로운 핸드폰을 사야 했습니다. 아들은 애플폰(ApplePhone)으로 교환(change)했고 이 애플폰은 영상통화 뿐 만 아니라 통화내용을 녹음하는 기능도 가능합니다.
- 위의 내용과 실행 결과를 참고하여 상속, 오버라딩, 객체배열, 다형성의 개념을 사용하여 필요한 클래스를 작성하고 아래의 메인 메소드를 사용하여 실행 결과를 테스트하시오.



```
//
//FamilyExample.java
//
public class FamilyExample {
    public static void main(String[] args) {

        Family family = new Family();

        family.operation(); System.out.println();

        System.out.println( family.change("boy", "applephone") + "\n");
        System.out.println( family.change("girl", "andrphone") + "\n");

        family.operation(); System.out.println();
    }
}
```


19. **AnimalTest** 클래스가 다음과 같이 동작하도록 클래스들을 완성하시오. 단 **Animal** 클래스는 객체 생성이 불가능 합니다. **reaction()**은 **Animal** 클래스에서 인스턴스를 확인하여 출력하고, **sound()**는 오버라이딩하여 사용합니다. **Dog**는 멍멍, **Cat**은 야~옹 합니다.

```
<terminated> AnimalTest [Java Application] C:\Program Files\Java\jdk-1.8.0_101\bin\java.exe
Nabee고양이 꼬리내리고 야~옹
Anonymous강아지 꼬리올리고 멍멍
Anonymous고양이 꼬리내리고 야~옹
Anonymous강아지 꼬리올리고 멍멍
Rock강아지 꼬리올리고 멍멍

Pet들 중 Dog는 3마리
```

```
package test;
```

```
public class AnimalTest {
    public static void main(String[] args) {
        Animal[] pets = {
            new Cat("Nabee"),
            new Dog(),
            new Cat(),
            new Dog(),
            new Dog("Rock")
        };

        for (Animal ani : pets) {
            System.out.print("\n" + ani + "_" + ani.reaction(ani));
            ani.sound();
        }
        System.out.print("\n\nPet들 중 Dog는 " + Dog.getCountDog() + "마리\n");
    }
}
```