

자바프로그래밍 II 3주차 실습 과제

(this_java_07장_상속_ex)

20145128

박인근

문제 1. 다음 프로그램의 출력은?

```
class Parent {
    Parent() {
        System.out.print("Parent");
    }
}
public class Child extends Parent {
    public static void main(String[] args) {
        new Child();
        new Parent();
    }
}
```

해답 : ParentParent

문제 2. protected 클래스에 있는 메소드를 클래스 안에서만 사용하려면 어떤 접근제어 지시어를 붙여야 하는가?

- ① static ② final ③ private ④ protected

해답 : 3

문제 3. 다음과 같은 클래스 정의에 대하여 답하라.

```
class Student {
    private int number;
    public String name;
}
public class GraduateStudent extends Student {
```

```
public String lab;  
}
```

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

해답 : 수퍼 클래스 : Student 서브 클래스 : GraduateStudent

(2) 생성자를 추가하여 보라. 서브 클래스의 생성자에서 수퍼 클래스의 생성자를 명시적으로 호출하게 하라.

해답 :

```
class Student {  
    private int number;  
    public String name;  
    Student(){  
    }  
  
public class GraduateStudent extends Student {  
    public String lab;  
    GraduateStudent() {  
        super();  
    }  
}
```

(3) GraduateStudent s = new GraduateStudent();와 같이 객체를 생성하였다고 하자. 다음 중 필드를 잘못 접근한 것은?

- ① s.number = 10;
- ② s.lab = "Image Processing Lab";
- ③ s.name = "홍길동";

해답 : 1

문제 4. 다음 소스에서 오류를 찾아보자.

```
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;
        }
    }
}

```

해답 : 3가지 오류 존재 . 첫 번째는 하위 클래스에서 상위클래스를 상속받을 때 extends가 아닌 extends 를 사용하여야 함. 두 번째는 하위클래스(MountainBike)의 생성자에서 super(); 를 사용할 수 없는데 이유는 상위클래스(Bike) 의 생성자가 존재하지 않기 때문이다. 세 번째는 마찬가지로 하위클래스 생성자에서 gear=g 부분인데 gear는 상위클래스에서 private로 다른 모든 클래스의 접근을 제한하는 접근제한자를 사용하였기 때문에 하위클래스에서 사용이 불가하다.

문제 5. 다음 프로그램의 출력을 쓰시오.

```

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();
    }
}

```

해답 :

동물입니다:Brave

사자입니다.

동물입니다:UNKNOWN

사자입니다.

문제 6. 다음 프로그램의 출력은?

```
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));  
    }  
}
```

해답 : A클래스에서 myMethod를 final 메서드로 생성하였기 때문에 하위 클래스인 B클래스에서 myMethod 오버라이딩 불가로 인해 오류 발생

문제 7. 다음 프로그램의 실행 결과는?

```
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");  
    }  
}
```

해답 :

1

실행 완료

문제 8. 다음과 같은 클래스 정의에서 질문에 답하라.

```
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 슈퍼클래스의 메소드를 재정의하고 있는가? (힌트) 정적 메소드로 인스턴스 메소드를 재정의할 수 없다.

해답 : MethodTwo

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

해답 : MethodFour

(3) 컴파일 오류를 지적하라. 실제로 컴파일해보아도 좋다.

해답 : 먼저 하위클래스의 methodOne(int i) 부분에서 오류가 발생하는데 이유는 상위클래스에 이미 같은 메소드가 존재하고 정적 메소드로는 인스턴스 메소드를 재정의 할 수 없기 때문에 오류가 발생한다. 다음으로는 하위클래스의 methodThree(int i) 부분에서 오류가 발생하는데 이유는 상위클래스에 static을 사용하여 정적메소드를 정의했기 때문에 하위 클래스에서 재정의 할 수 없다.

(4) 컴파일 오류 수정 후 프로그램의 출력은?

해답 :

B4
A4
A2
B2
B2

문제 12. 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 작성하여 보자. Customer는 고객 번호와 마일리지를 필드로 가지고 있다. 한 개 이상의 생성자를 작성하고 적절한 접근자 메소드와 설정자 메소드를 작성한다. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.

해답 소스 :

[Person 클래스]

```
package week3;

public class Person {
    String name; // 이름
    String address; // 주소
    String tel; //전화번호

    Person() {
        this("unknown", "unknown", "unknown");
    }

    Person(String n, String a, String t) {
        this.name = n;
        this.address = a;
        this.tel = t;
    }

    public void setName(String n){
        this.name = n;
    }

    public void setAddress(String a){
        this.address = a;
    }

    public void setTel(String t){
        this.tel = t;
    }

    public String getName(){
        return "고객 이름 : " + this.name + "\n";
    }

    public String getAddress(){
        return "고객 주소 : " + this.address + "\n";
    }

    public String getTel(){
        return "고객 전화번호 : " + this.tel + "\n";
    }
}
```

[Customer 클래스]

```
package week3;

public class Customer extends Person {
    String Pnum; //고객번호
    String Pm; //고객 마일리지

    Customer() {
        this("unknown", "unknown");
    }

    Customer(String pn, String pm) {
        this.Pnum = pn;
        this.Pm = pm;
    }

    public String getPnum() {
        return "고객번호 : " + this.Pnum + "번\n";
    }

    public void setPnum(String pnum) {
        this.Pnum = pnum;
    }

    public String getPm() {
        return "고객 마일리지 : " + this.Pm + "점\n";
    }

    public void setPm(String pm) {
        this.Pm = pm;
    }
}
```

[Prun 클래스]

```
package week3;

public class Prun {
    public void run() {
        Person p = new Person();
        Customer c = new Customer();

        System.out.print("--생성할 때 정보 미 입력시 Person객체 접근자
메소드로 확인--\n");
        System.out.print(p.getName());
        System.out.print(p.getAddress());
        System.out.print(p.getTel());
        System.out.print("--생성할 때 정보 미 입력시 Customer객체 접근자
메소드로 확인--\n");
        System.out.print(c.getName());
    }
}
```



```

        System.out.print(c.getAddress());
        System.out.print(c.getTel());
        System.out.print(c.getPnum());
        System.out.print(c.getPm());

        System.out.print("--설정자 메소드로 정보 변경(Person 객체)--\n");
        p.setName("박인근");
        p.setAddress("서울시 동동동");
        p.setTel("010-0000-0000");
        System.out.print(p.getName());
        System.out.print(p.getAddress());
        System.out.print(p.getTel());

        System.out.print("--설정자 메소드로 정보 변경(Customer 객체)--\n");

        c.setName("박근인");
        c.setAddress("서울시 군군군");
        c.setTel("010-1111-0000");
        c.setPnum("83");
        c.setPm("1224");
        System.out.print(c.getName());
        System.out.print(c.getAddress());
        System.out.print(c.getTel());
        System.out.print(c.getPnum());
        System.out.print(c.getPm());
    }

    public void run2() {
        Person p2 = new Person("홍길동", "서울시 동동구", "010-1234-5678");
        Customer c2 = new Customer("128", "840");

        System.out.print("--생성할 때 정보 입력시 Person객체 접근자 메소드로 확인--\n");
        System.out.print(p2.getName());
        System.out.print(p2.getAddress());
        System.out.print(p2.getTel());
        System.out.print("--생성할 때 정보 입력시 Customer객체 접근자 메소드로 확인--\n");
        System.out.print(c2.getName());
        System.out.print(c2.getAddress());
        System.out.print(c2.getTel());
        System.out.print(c2.getPnum());
        System.out.print(c2.getPm());

        System.out.print("--설정자 메소드로 정보 변경(Person 객체)--\n");
        p2.setName("참조은");
        p2.setAddress("서울시 구구동");
        p2.setTel("010-1200-3400");
        System.out.print(p2.getName());
        System.out.print(p2.getAddress());
        System.out.print(p2.getTel());
    }

```

```

        System.out.print("--설정자 메소드로 정보 변경 (Customer 객체)--\n");

        c2.setName("참나쁜");
        c2.setAddress("서울시 공공시");
        c2.setTel("010-2211-0330");
        c2.setPnum("32");
        c2.setPm("50");
        System.out.print(c2.getName());
        System.out.print(c2.getAddress());
        System.out.print(c2.getTel());
        System.out.print(c2.getPnum());
        System.out.print(c2.getPm());
    }
}

```

[PTest 클래스]

```

package week3;
import java.util.Scanner;

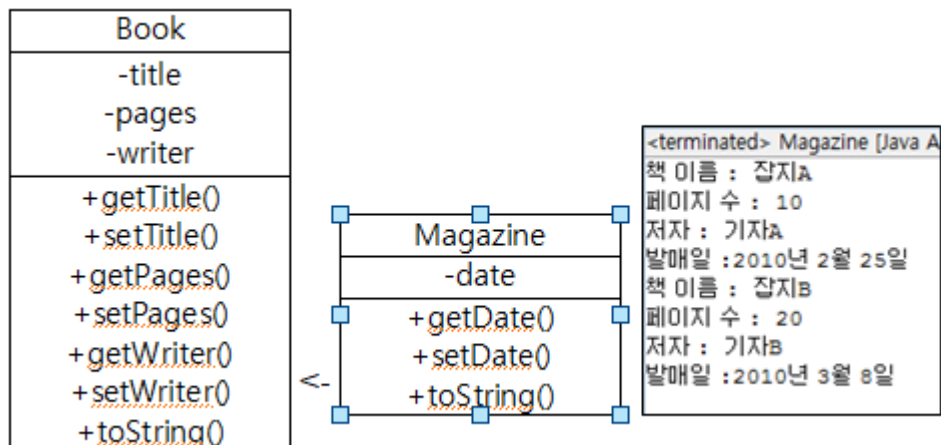
public class PTest {
    public static void main(String args[]){
        Prun pr = new Prun();
        Scanner k = new Scanner(System.in);
        int n;

        while(true) {
            System.out.println("Person & Customer 클래스의 테스트
클래스입니다.");

            System.out.println("보고 싶은 정보를 입력해 주세요.");
            System.out.println("1 : 고객정보 입력 x 객체, 2 : 고객정보
입력 o 객체, 3 : 종료");
            n = k.nextInt();
            if(n==1) {
                pr.run();
            }
            else if(n==2) {
                pr.run2();
            }
            else if(n==3) {
                System.out.println("종료합니다.");
                break;
            }
        }
    }
}

```

문제 13. 일반적인 책을 나타내는 Book 클래스를 상속받아서 잡지를 나타내는 Magazine 클래스를 작성하여 보자. Book 클래스는 제목, 페이지수, 저자 등의 정보를 가진다. Magazine 클래스는 추가로 발매일 정보를 가진다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성한다. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.



```
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());
    }
}
```

해답 소스 :

[Book 클래스]

```
package week3;

public class Book {
    String title; // 책 이름
    int pages; // 페이지 수
    String writer; // 저자
```

```

Book(String t, int p, String w){
    this.title = t;
    this.pages = p;
    this.writer = w;
}

public String getTitle() {
    return this.title;
}
public void setTitle(String title) {
    this.title = title;
}
public int getPages() {
    return this.pages;
}
public void setPages(int pages) {
    this.pages = pages;
}
public String getWriter() {
    return this.writer;
}
public void setWriter(String writer) {
    this.writer = writer;
}

public String toString(){
    return "책 이름 : " + this.title + "\n페이지 수 : " + this.pages
+ "\n저자 : " + this.writer;
}
}

```

[Magazine 클래스]

```

package week3;

public class Magazine extends Book{
    String date; // 발매일 정보

    Magazine(String t, int p, String w, String d){
        super(t,p,w);
        this.date = d;
    }

    public String getDate() {
        return this.date;
    }

    public void setDate(String date) {
        this.date = date;
    }

    public String toString(){
        return super.toString() + "\n발매일 : " + this.date;
    }
}

```

[BandMTest 클래스]

```
package week3;

public class BandMTest {
    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());
    }
}
```