

문제풀이 2

[연습문제]

[7-3] 오버라이딩의 정의와 필요성에 대해서 설명하시오

오버라이딩이란, 부모 클래스로부터 상속 받은 함수를 자식 클래스에서 다시 한번 재정의 하는 것으로, 이를 통해 자식 클래스에 좀 더 특화된 형태로 사용할 수 있게 된다.

[답안] 오버라이딩(Overriding)이란, '조상 클래스로부터 상속받은 메서드를 자손 클래스에 맞게 재정의 하는 것' 을 말한다. 조상 클래스로부터 상속받은 메서드를 자손 클래스에서 그대로 사용할 수 없는 경우가 많기 때문에 오버라이딩이 필요하다.

[7-4] 다음 중 오버라이딩의 조건으로 옳지 않은 것은? (모두 고르시오)

1. 조상의 메서드와 이름이 같아야 한다.
2. 매개변수의 수와 타입이 모두 같아야 한다.
3. 접근 제어자는 조상의 메서드보다 좁은 범위로만 변경할 수 있다.
 - 조상 클래스의 메서드를 자손 클래스에서 오버라이딩할 때
 - 접근 제어자를 조상 클래스의 메서드보다 좁은 범위로 변경할 수 없다.
 - 예외는 조상 클래스의 메서드보다 많이 선언할 수 없다.
 - 인스턴스 메서드를 static 또는 그 반대로 변경할 수 없다.
4. 조상의 메서드보다 더 많은 수의 예외를 선언할 수 있다.

[7-8] 다음 중 접근제어자를 접근범위가 넓은 것에서 좁은 것의 순서로 바르게 나열한 것은?

- **public** → (default) → **protected** → **private**

[7-9] 다음 중 제어자 final 을 붙일 수 있는 대상과 붙였을 때 그 의미를 적은 것이다. 옳지 않은 것은? (모두 고르시오)

1. 지역변수 - 값을 변경할 수 없다.
2. 클래스 - 상속을 통해 클래스에 새로운 멤버를 추가할 수 없다.
3. ~~메서드 - 오버로딩을 할 수 없다.~~
4. ~~멤버변수 - 값을 변경할 수 없다.~~
 - 생성자를 통해서 딱 한번 값을 변경할 수 있다.

[7-12] 다음 중 접근 제어자에 대한 설명으로 옳지 않은 것은? (모두 고르시오)

1. public 은 접근제한이 전혀 없는 접근 제어자이다.
2. (default) 가 붙으면, 같은 패키지 내에서만 접근이 가능하다.
3. 지역변수에도 접근 제어자를 사용할 수 있다.
4. protected 가 붙으면, 같은 패키지 내에서도 접근이 가능하다.
5. ~~protected 가 붙으면, 다른 패키지의 자손 클래스에서 접근이 가능하다.~~

[7-13] Math 클래스의 생성자는 접근 제어자가 private 이다. 그 이유는 무엇인가?

Math 클래스의 인스턴스를 사용할 이유가 없기 때문이다. 즉, 클래스 함수만 사용하기를 의도하였기 때문이다.

[7-15] 클래스가 다음과 같이 정의되어 있을 때, 형변환을 올바르게 하지 않은 것은? (모두 고르시오.)

```
class Unit {}

class AirUnit extends Unit {}
class GroundUnit extends Unit {}

class Tank extends GroundUnit {}
class AirCraft extends AirUnit {}

Unit u = new GroundUnit();
Tank t = new Tank();
AirCraft ac = new AirCrate();
```

1. u = (Unit)ac;
2. u = ac;
3. GroundUnit gu = (GroundUnit)u;
4. AirUnit au = ac;
5. t = (Tank)u;
6. GroundUnit gu = t;

1. 부모타입으로 자식타입의 객체를 참조할 때는 묵시적으로 형변환이 일어난다.
2. 부모타입 객체를 자식타입으로 참조하게 할때 명시적 형변환을 해줘야한다

[7-16] 다음 중 연산결과가 true 가 아닌 것은? (모두 고르시오)

```
class Car {}
class FireEngine extends Car implements Movable {}
class Ambulance extends Car {}

FireEngine fe = new FireEngine();
```

1. fe instanceof FireEngine ⇒ True
2. fe instanceof Movable ⇒ True
3. fe instanceof Object ⇒ True
4. fe instanceof Car ⇒ True
5. **fe instanceof Ambulance ⇒ False**

[7-21] 다음과 같이 attack 메서드가 정의되어 있을 때, 이 메서드의 매개변수로 가능한 것 두 가지를 적으시오.

```
interface Movable {
    void move(int x, int y);
}

void attack(Movable f) {
    /* 내용 생략 */
}
```

[7-24] 다음 중 인터페이스의 장점이 아닌 것은?

1. 표준화를 가능하게 해준다.
2. 서로 관계없는 클래스들에게 관계를 맺어 줄 수 있다.
3. 독립적인 프로그래밍이 가능하다.
4. 다중상속을 가능하게 해준다.
5. ~~패키지간의 연결을 도와준다.~~

[7-25]

```
Outer.Inner inner = new Outer().new Inner();
System.out.println(inner.iv);
```

[7-26]

??

[7-29] 지역 클래스에서 외부 클래스의 인스턴스 멤버와 static 멤버에 모두 접근할 수 있지만, 지역변수는 final이 붙은 상수만 접근할 수 있는 이유는 무엇인가?

- 지역 클래스 또한, 해당 클래스의 인스턴스 멤버라고 생각할 수 있기 때문이다.