

[lab07 보고서]

디지털미디어학과

2019111677

김지연

1. 다음과 같은 관계의 인터페이스와 클래스가 있다고 하자. MyClass에서 반드시 구현해야 하는 메소드는 어떤 것인가? (이론문제 임. 즉, 문제에 대한 답만 제출)

```
interface A {  
    public float mA(int a);  
}  
  
interface B extends A {  
    public int mB(int a);  
  
    public Object mBB(int a);  
}  
  
class C {  
    public void mC(int a) {  
        System.out.println("안녕하세요");  
    }  
}  
  
public class MyClass extends C implements B {  
  
}
```

반드시 구현해야 하는 메소드

- **mA, mB, mBB**

2. 다음의 인터페이스 선언과 사용에서 잘못된 점을 모두 지적해 보세요. (이론문제)

```
public interface Edible {  
    boolean amount;  
    final int TYPE=10;  
    public void eat() {    };  
};  
public class Sandwich extends Edible {  
    public void eat() {    }  
}
```

boolean amount;

-> 필드(변수)는 선언될 수 없다. 따라서 boolean amount는 삭제한다.

public void eat() { };

-> 몸체가 없는 추상 메소드만 정의 가능하다. 따라서 public void eat();로 수정한다.

public class Sandwich extends Edible

-> 클래스에서 인터페이스를 상속할 때는 extends가 아닌 implements예약어를 사용해야 한다. 따라서 public class Sandwich implements Edible로 수정한다.

- 인터페이스에서 상수는 정의 가능하기 때문에 final int TYPE=10; 은 문제없음

- extends는 클래스를 상속할 때 사용하는 예약어

3. 다음과 같은 조건을 만족하는 프로그램을 작성하세요.

(실습문제 임. 즉, 1.분석-2.소스코드-3.결과화면 제출)

(a) 인터페이스 GraphicsObject에서는 int getArea()과 void draw()가 선언된다.

(b) GraphicsObject을 구현하는 "Rectangle"클래스를 작성한다.

(c) Rectangle 클래스는 2개의 private 정수 필드인 length와 width을 가진다.

(d) setDimensions(int l, int w) 메소드는 length와 width의 값을 설정한다.

[분석]

GraphicsObject 인터페이스에 몸체가 없는 추상 메소드 getArea()와 draw()를 선언한다. GraphicsObject 인터페이스를 implements하는 Rectangle 클래스를 작성한다. 2개의 정수 length와 width를 private로 선언한다. getArea()와 draw()의 몸체와 setDimensions() 설정자를 작성한다. Main type에 Rectangle클래스를 객체로 생성하고, setDimensions설정자를 사용하여 가로와 세로의 높이를 지정한다. getArea()와 draw()메소드를 메세징하여 호출한다.

[소스코드]

```
public interface GraphicsObject {
    int getArea();
    void draw();
}

public class Rectangle implements GraphicsObject {
    private int length ;
    private int width ;

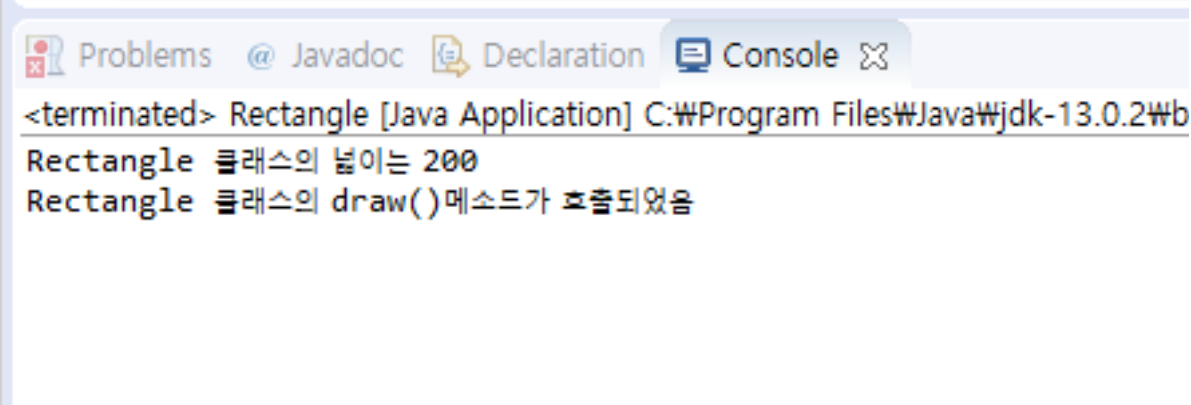
    public void setDimensions(int l, int w){
        length = l;
        width = w;
    }

    public int getArea() {
        return width*length;
    }

    public void draw() {
        System.out.println("Rectangle 클래스의 draw()메소드가 호출되었음");
    }

    public static void main(String[] args) {
        Rectangle r = new Rectangle();
        r.setDimensions(10, 20);
        System.out.println("Rectangle 클래스의 넓이는 " + r.getArea());
        r.draw();
    }
}
```

[결과화면]



The screenshot shows a console window with a tab bar at the top containing 'Problems', 'Javadoc', 'Declaration', and 'Console'. The 'Console' tab is active. The output text is as follows:

```
<terminated> Rectangle [Java Application] C:\Program Files\Java\jdk-13.0.2\bin
Rectangle 클래스의 넓이는 200
Rectangle 클래스의 draw()메소드가 호출되었음
```

4. 비디오 플레이어는 play와 stop이라는 조작을 할 수 있다. 따라서 인터페이스 controllable을 아래와 같이 정의할 수 있다.

```
public interface controllable {  
    void play();  
    void stop();  
}
```

controllable 인터페이스를 사용하는 클래스 Test를 다음과 같은 방법으로 각각 정의해 보세요. (실습문제)

1) controllable 인터페이스를 implements하는 Test 클래스를 정의(아래 코드가 정상적으로 실행되도록 프로그램을 완성하세요.)

```
public class Test implements controllable{  
    //이 부분을 완성하세요.  
    Public static void main(String[] args) {  
        Test test = new Test();  
        test.play();  
        test.stop();  
    }  
}
```

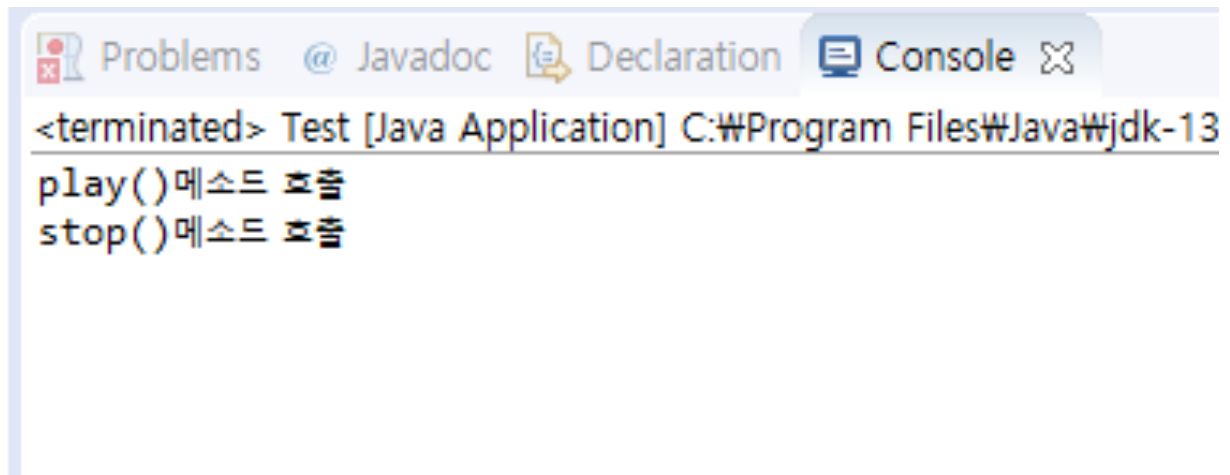
[분석]

Test클래스에 Controllable인터페이스를 implements 시킨다. test클래스에 인터페이스의 구현부를 만들어 준다. {}안에 추상 메소드의 몸체를 정의한다는 의미이다. Main type에 test클래스의 객체를 생성하고 메세징하여 메소드를 호출한다.

[소스코드]

```
public class Test implements Controllable{  
    public void play() {  
        System.out.println("play()메소드 호출");  
    }  
    public void stop() {  
        System.out.println("stop()메소드 호출");  
    }  
    public static void main(String[] args) {  
        Test test = new Test();  
        test.play();  
        test.stop();  
    }  
}
```

[결과화면]



2) Test 클래스의 main()에서 무명 클래스로 controllable 인터페이스를 구현

```
public static void main(String[] args) {  
    controllable c = // 이 부분을 완성하세요.  
    c.play();  
    c.stop();  
}
```

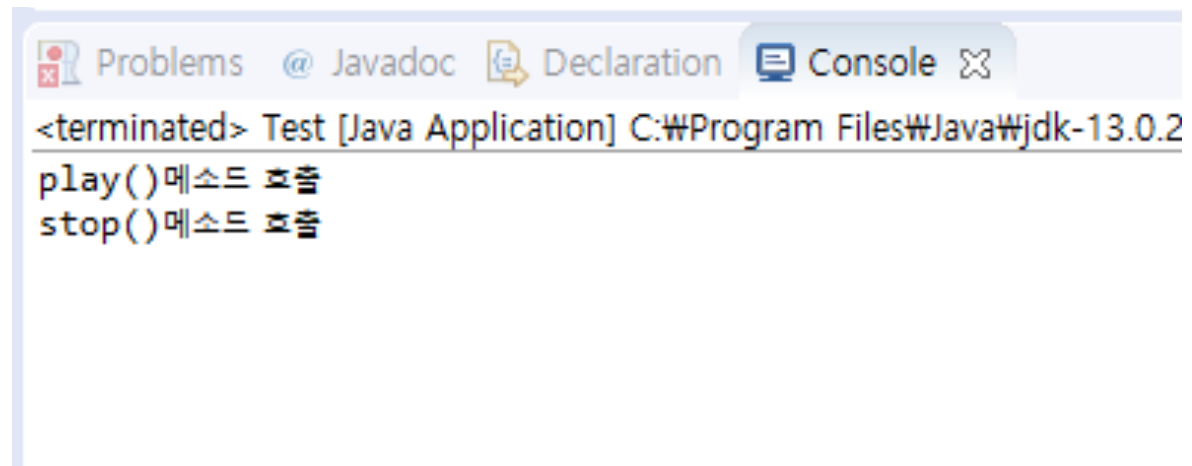
[분석]

Controllable인터페이스를 무명클래스로 구현하기 위해 클래스를 정의하면서 동시에 객체를 생성한다. new키워드 다음 인터페이스 이름을 적고, 클래스 몸체를 정의한다. 무명클래스의 구현부를 다 작성하고 나면 메소드를 호출해준다.

[소스코드]

```
public class Test {  
    public static void main(String[] args) {  
        Controllable c = new Controllable() {  
            public void play() {  
                System.out.println("play()메소드 호출");  
            }  
            public void stop() {  
                System.out.println("stop()메소드 호출");  
            }  
        };  
        c.play();  
        c.stop();  
    }  
}
```

[결과화면]



The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output displays the following text:

```
<terminated> Test [Java Application] C:\Program Files\Java\jdk-13.0.2  
play()메소드 호출  
stop()메소드 호출
```