

# 과제(클래스)

문1) 다음은 도형을 정의한 Shape클래스이다. 이 클래스를 부모로 하는 Circle클래스와 Rectangle클래스를 작성하시오. 이 때, 생성자도 각 클래스에 맞게 적절히 추가해야 한다.

(1) 클래스명 : Circle

부모클래스 : Shape

멤버변수 : double r - 반지름

(2) 클래스명 : Rectangle

부모클래스 : Shape

멤버변수 : int width - 폭

int height - 높이

메서드 :

1. 메서드명 : isSquare

기능 : 정사각형인지 아닌지를 알려준다.

반환타입 : boolean

매개변수 : 없음

# 과제(클래스)

```
abstract class Shape {  
    Point p;  
    Shape() {  
        this(new Point(0,0));  
    }  
    Shape(Point p) {  
        this.p = p;  
    }  
}
```

```
abstract double calcArea(); // 도형의 면적을 계산해서 반환하는 메서드
```

```
    Point getPosition() {  
        return p;  
    }  
    void setPosition(Point p) {  
        this.p = p;  
    }  
}
```

# 과제(클래스)

```
class Point {  
    int x;  
    int y;  
    Point() {  
        this(0,0);  
    }  
    Point(int x, int y) {  
        this.x=x;  
        this.y=y;  
    }  
  
    public String toString() {  
        return "["+x+", "+y+"]";  
    }  
}
```

# 과제(클래스)

문2) 다음과 같은 실행결과를 얻도록 Point3D클래스의 equals()를 멤버변수인 x, y, z의 값을 비교하도록 오버라이딩하고, toString()은 실행결과를 참고해서 적절히 오버라이딩하시오.

```
class Exercise9_2 {  
    public static void main(String[] args) {  
        Point3D p1 = new Point3D(1,2,3);  
        Point3D p2 = new Point3D(1,2,3);  
  
        System.out.println(p1);  
        System.out.println(p2);  
        System.out.println("p1==p2?"+(p1==p2));  
        System.out.println("p1.equals(p2)?"+(p1.equals(p2)));  
    }  
}
```

# 과제(클래스)

```
class Point3D {  
    int x,y,z;  
    Point3D(int x, int y, int z) {  
        this.x=x;  
        this.y=y;  
        this.z=z;  
    }  
    Point3D() {  
        this(0,0,0);  
    }  
  
    public boolean equals(Object obj) {  
        /*  
        (1) 인스턴스변수 x, y, z를 비교하도록 오버라이딩하시오.  
        */  
    }  
}
```

# 과제(클래스)

```
public String toString() {  
    /*  
    (2) 인스턴스변수 x, y, z의 내용을 출력하도록 오버라이딩하시오.  
    */  
}  
}
```

[실행결과]

[1,2,3]

[1,2,3]

p1==p2?false

p1.equals(p2)?true

# 과제(클래스) - 상속

문3) 다음 코드의 실행했을 때 호출되는 생성자의 순서와 실행결과를 적으시오.

```
class Parent {  
    int x=100;  
    Parent() {  
        this(200);  
    }  
    Parent(int x) {  
        this.x = x;  
    }  
    int getX() {  
        return x;  
    }  
}
```

# 과제(클래스) - 상속

```
class Child extends Parent {  
    int x = 3000;  
    Child() {  
        this(1000);  
    }  
    Child(int x) {  
        this.x = x;  
    }  
}
```

```
class Exercise3 {  
    public static void main(String[] args) {  
        Child c = new Child();  
        System.out.println("x="+c.getX());  
    }  
}
```



# 과제(클래스) - 상속

문4) 아래 세 개의 클래스로부터 공통부분을 뽑아서 Unit이라는 클래스를 만들고, 이 클래스를 상속받도록 코드를 변경하시오.

```
class Marine { // 보병
    int x, y; // 현재 위치
    void move(int x, int y) { /* 지정된 위치로 이동 */ }
    void stop() { /* 현재 위치에 정지 */ }
    void stimPack() { /* 스팀팩을 사용한다.*/}
}
```

```
class Tank { // 탱크
    int x, y; // 현재 위치
    void move(int x, int y) { /* 지정된 위치로 이동 */ }
    void stop() { /* 현재 위치에 정지 */ }
    void changeMode() { /* 공격모드를 변환한다. */}
}
```

# 과제(클래스) - 상속

```
class Dropship { // 수송선
    int x, y; // 현재 위치
    void move(int x, int y) { /* 지정된 위치로 이동 */ }
    void stop() { /* 현재 위치에 정지 */ }
    void load() { /* 선택된 대상을 태운다.*/ }
    void unload() { /* 선택된 대상을 내린다.*/ }
}
```

# 과제(클래스)

문5) 다음과 같은 실행결과를 얻도록 코드를 완성하시오.  
[Hint] instanceof연산자를 사용해서 형변환 한다.

[실행결과]

춤을 춥니다.

노래를 합니다.

그림을 그립니다.

메서드명 : action

기능 : 주어진 객체의 메서드를 호출한다.

DanceRobot인 경우, dance()를 호출하고,

SingRobot인 경우, sing()을 호출하고,

DrawRobot인 경우, draw()를 호출한다.

반환타입 : 없음

매개변수 : Robot r – Robot인스턴스 또는 Robot의 자손 인스턴스

# 과제(클래스)

```
class Exercise5 {  
    /*  
        (1) action메서드를 작성하시오.  
    */  
    public static void main(String[] args) {  
        Robot[] arr = { new DanceRobot(), new SingRobot(),  
                        new DrawRobot()};  
        for(int i=0; i< arr.length;i++)  
            action(arr[i]);  
    } // main  
}
```

# 과제(클래스)

```
class Robot {}  
class DanceRobot extends Robot {  
    void dance() {  
        System.out.println("춤을 춥니다.");  
    }  
}  
class SingRobot extends Robot {  
    void sing() {  
        System.out.println("노래를 합니다.");  
    }  
}  
class DrawRobot extends Robot {  
    void draw() {  
        System.out.println("그림을 그립니다.");  
    }  
}
```

