

1. 원을 표현하는 다음 Circle 클래스가 있다.

```
class Circle {  
    private int radius;  
    public Circle(int radius) { this.radius = radius; }  
    public int getRadius() { return radius; }  
}
```

Circle 클래스를 상속받은 NamedCircle 클래스를 작성하여, 다음 main()을 실행할 때 다음 실행 결과와 같이 출력되도록 하라. [목적 - 간단한 클래스 상속] [난이도 하]

```
public static void main(String[] args) {  
    NamedCircle w = new NamedCircle(5, "Waffle");  
    w.show();  
}
```

Waffle, 반지름 = 5

힌트) NamedCircle 클래스에는 name 필드, 생성자, show() 메소드, main()이 작성되어야 한다.

2. 인터페이스 AddrInterface의 코드는 다음과 같다.

```
interface AddrInterface {  
    int add(int x, int y); // x와 y의 합 리턴  
    int add(int n); // 1에서 n까지의 정수 합 리턴. n은 0보다 큰 수로 가정  
}
```

AddrInterface를 상속받은 클래스 MyAdder를 작성하여, 다음 main()을 실행할 때 아래 실행 결과와 같이 출력되도록 하라. [목적 - 인터페이스를 구현한 클래스 만들기] [난이도 하]

```
public static void main(String[] args) {  
    MyAdder adder = new MyAdder();  
    System.out.println(adder.add(5, 10));  
    System.out.println(adder.add(10));  
}
```

15

55

힌트) Myadder 클래스에 두 add() 메소드를 구현할 때, public 타입으로 선언해야 한다.

4. 2차원 상의 한 점을 표현하는 Point 클래스는 다음과 같다.

```
class Point {  
    private int x, y;  
    public Point(int x, int y) { this.x = x; this.y = y; }  
    public int getX() { return x; }  
    public int getY() { return y; }  
    protected void move(int x, int y) { this.x = x; this.y = y; }  
}
```

다음 main()과 실행 결과를 참고하여 Point를 상속받은 ColorPoint 클래스(main() 포함)를 작성하라. [목적 - 클래스 상속] [난이도 중]

```
public static void main(String[] args) {  
    ColorPoint cp = new ColorPoint(5, 5, "YELLOW");  
    cp.setPoint(10, 20);  
    cp.setColor("GREEN");  
    cp.show();  
}
```

GREEN색으로(10, 20)
