

# 자바프로그래밍및실습 과제 5

214823 컴퓨터정보통신공학과 박종현

June 2022

코드 주석은 Javadoc 코드 문서화 스펙을 참조하여 작성함.

참조: <https://docs.oracle.com/en/java/javase/17/docs/specs/javadoc/doc-comment-spec.html>

## 1 과제 1

### 1.1 과제 1.1

#### 1.1.1 소스 코드

```
1  /**
2   * @author @ShapeLayer
3   */
4
5  // 문제 제시 코드
6  public class Main {
7      public static void main(String[] args){
8          Circle a = new Circle(2, 3, 5);
9          Circle b = new Circle(2, 3, 30);
10         System.out.println("원 a : " + a);
11         System.out.println("원 b : " + b);
12         if(a.equals(b))
13             System.out.println("같은 원");
14         else
15             System.out.println("서로 다른 원");
16     }
17 }
18
19 /**
20 * Circle 클래스
21 *
22 * Circle 클래스는 2차원 상의 원의 데이터를 처리할 수 있도록 작성되었습니다.
23 */
24 class Circle {
25     /**
26     * 원 중심의 x좌표
27     */
28     int x;
29
30     /**
31     * 원 중심의 y좌표
32     */
33     int y;
34
35     /**
36     * 원의 반지름
37     */
38     int radius;
39
40     /**
41     * Circle 객체를 생성합니다.
42     * @param x {@code int} 원 중심의 x좌표
43     * @param y {@code int} 원 중심의 y좌표
44     * @param radius {@code int} 원의 반지름
45     */
46     Circle(int x, int y, int radius) {
47         this.x = x;
48         this.y = y;
49         this.radius = radius;
50     }
51
52     /**
53     * Circle 객체 정보를 요약하여 반환합니다.
```

```

54     * @return {@code String} 요약 정보
55     */
56     public String toString() { return String.format("Circle(%d, %d)반지름%d", this.x, this.y,
↪      this.radius); }
57
58     /**
59     * 두 원이 동일한지 확인합니다.
60     * @return {@code boolean} 동일 여부
61     */
62     boolean equals(Circle circle) { return this.x == circle.x && this.y == circle.y; }
63 }

```

### 1.1.2 실행 예제

```

shapelaye@ShapLayer-MacBookAir Prob1.1 % java Prob1.1.java
원 a : Circle(2, 3)반지름 5
원 b : Circle(2, 3)반지름 30
같은 원

```

## 1.2 과제 1.2

### 1.2.1 소스 코드

프로그램 진입점 /java/Prob1/Prob1.2/main.java

```

1 package main;
2 import etc.Calc;
3
4 /**
5  * MainApp 클래스
6  *
7  * 일련의 자바 애플리케이션의 진입점입니다.
8  */
9 public class MainApp {
10     // 문제 제시 코드
11     public static void main (String[] args) {
12         Calc c = new Calc(10,20);
13         System.out.println(c.sum());
14     }
15 }

```

/java/Prob1/Prob1.2/etc/Calc.java

```

1 package etc;
2
3 // class Calc {
4 // main.java:2: error: Calc is not public in etc; cannot be accessed from outside package
5 /**
6  * Calc 클래스
7  *
8  * 간단한 수 계산에 대한 메서드를 포함하고 있습니다.
9  */
10 public class Calc {
11     // 문제 제시 코드
12     private int x,y;
13     public Calc(int x, int y) {this.x = x ; this.y = y; }
14     public int sum() { return x+y; }
15 }

```

### 1.2.2 실행 예제

```

shapelaye@ShapLayer-MacBookAir Prob1.1 % cd ../Prob1.2
shapelaye@ShapLayer-MacBookAir Prob1.2 % ls
README.md    buildrun.sh  etc          main.java
shapelaye@ShapLayer-MacBookAir Prob1.2 % sh buildrun.sh
30

```

## 1.3 과제 1.3

### 1.3.1 소스 코드

프로그램 진입점 /java/Prob1/Prob1.3/app/GraphicEditor.java

```
1 package app;
2 import base.Shape;
3 import derived.Circle;
4
5 /**
6  * GraphicEditor 클래스
7  *
8  * GraphicEditor 클래스는 일련의 애플리케이션의 진입점을 포함하고 있습니다.
9  */
10 public class GraphicEditor {
11     // 문제 제시 코드
12     public static void main(String[] args) {
13         Shape shape = new Circle();
14         shape.draw();
15     }
16 }
```

/java/Prob1/Prob1.3/base/Shape.java

```
1 package base;
2
3 // class Shape {
4 /**
5  * Shape 클래스
6  *
7  * Shape 클래스는 도형을 다루는 데 필요한 메서드가 포함되어 있습니다.
8  */
9 public class Shape {
10     public void draw() {System.out.println("Shape");}
11 }
```

/java/Prob1/Prob1.3/derived/Circle.java

```
1 package derived;
2 import base.Shape;
3
4 // class Circle extends Shape{
5 // 문제 제시 코드
6 /**
7  * Circle 클래스
8  * {@code Shape} 클래스를 상속함
9  */
10 public class Circle extends Shape{
11     public void draw() {System.out.println("Circle");}
12 }
```

### 1.3.2 실행 예제

```
shapelayer@Shapelayer-MacBookAir Prob1.2 % cd ../Prob1.3
shapelayer@Shapelayer-MacBookAir Prob1.3 % ls
README.md  app      buildrun.sh  derived
shapelayer@Shapelayer-MacBookAir Prob1.3 % sh buildrun.sh
Circle
```

## 2 과제2

### 2.1 소스 코드

```
1 import java.util.Scanner;
2 import java.util.Vector;
3
4 public class Main {
5     public static void main(String[] args) {
6         // 입력을 받기 위해 Scanner 객체 생성
7         Scanner sc = new Scanner(System.in);
8
9         // List<Integer> 벡터 객체 생성
10        Vector<Integer> gets = new Vector<Integer>();
11
12        // 입력 받을 변수 선언: nextInt()를 이 변수에 할당함
13        int get = -1;
14
15        System.out.print("정수(-1이 입력될 때까지) >> ");
16
17        // 반복문 시작
18        do {
19            get = sc.nextInt();
20            if (get == -1) continue; // 입력이 -1이라면 while()로 이동
21            gets.add(get); // 입력이 -1이 아니라면 리스트에 추가
22        } while (get != -1);
23        // -1이라면 반복문 중지
24
25        // 최대값 검색: 입력값은 모두 양의 정수이므로 음의 정수로 초기화 후 처리 시작
26        int max = -1;
27        for (int n: gets) // foreach문 사용
28            if (max < n) max = n;
29
30        System.out.println("가장 큰 수는 " + max);
31    }
32 }
```

### 2.2 실행 예제

```
shapelayer@Shapelayer-MacBookAir Prob2 % java Prob2.java
정수 (-1이 입력될 때까지) >> 10 6 22 6 88 77 -1
가장 큰 수는 88
```

## 3 과제3

### 3.1 소스 코드

```
1 import java.util.Scanner;
2 import java.util.List;
3 import java.util.ArrayList;
4
5 public class Main {
6     public static void main(String[] args) {
7         // 입력을 받아오기 위해 Scanner 객체 생성
8         Scanner sc = new Scanner(System.in);
9
10        // 리스트 객체 생성
11        List<Character> grades = new ArrayList<Character>();
12
13        System.out.print("6개의 학점을 빈 칸으로 분리 입력 (A/B/C/D/E/F) >> ");
14        for (int i = 0; i < 6; i++)
```

```

15 // sc.next()를 통한 입력은 한 문자(char형)만 받아오는 것이 아니기 때문에 String 형으로 반
↪ 환함.
16 // 따라서 charAt() 메서드를 사용하여 char 형 불러옴
17 grades.add(sc.next().charAt(0));
18
19 // 점수 합계용 변수 선언
20 double sums = 0;
21 for (char grade: grades) { // foreach문 사용
22     // switch-case문으로 처리 시작
23     switch (grade) {
24         case 'A':
25             sums += 4;
26             break;
27         case 'B':
28             sums += 3;
29             break;
30         case 'C':
31             sums += 2;
32             break;
33         case 'D':
34             sums += 1;
35             break;
36         case 'F':
37             // 학점 F가 존재한다는 맥락 차원에서 추가
38             sums += 0;
39             break;
40     }
41 }
42
43 System.out.println(sums/6);
44 }
45 }

```

## 3.2 실행 예제

```

shapelayer@Shapelayer-MacBookAir Prob3 % java Prob3.java
6개의 학점을 빈 칸으로 분리 입력 (A/B/C/D/E/F) >> A C A B F D
2.3333333333333335

```