

# 문제해결및실습:JAVA Homework #1

20011844

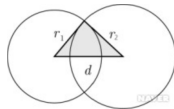
안수경

## < Hw1\_1 >

```
Console [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java (2021. 9. 16. 오후 2:3
<terminated> Hw1_1 [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java (2021. 9. 16. 오후 2:3
첫 번째 원의 중심과 반지름 입력 >> 10 10 3
두 번째 원의 중심과 반지름 입력 >> 12 12 2
두 원은 서로 겹친다.
```

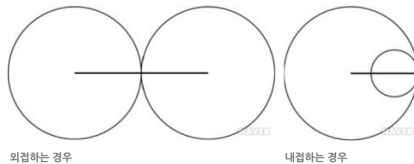
### 1) 두 점에서 만나는 경우

두 원의 교점의 개수는 반지름의 길이의 합 또는 차와 중심사이의 거리의 상대적인 크기에 의해 결정이 된다. 두 원의 반지름의 길이를 각각  $r_1, r_2$ 라 하고 중심간의 거리를  $d$ 라고 하면  $|r_1 - r_2| < d < r_1 + r_2$ 인 관계식을 만족할 때 두 원은 두 점에서 만나게 된다.



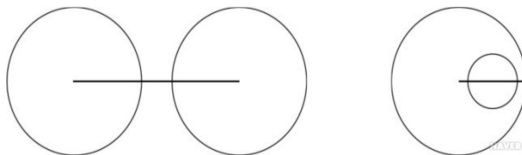
### 2) 접하는 경우

두 원 사이에  $r_1 + r_2 = d, |r_1 - r_2| = d$ 인 경우 두 원은 한 점에서 만난다. 전자의 경우를 외접한다고 하고 후자를 내접한다고 표현한다.



### 3) 만나지 않는 경우

두 원 사이에  $r_1 + r_2 < d, |r_1 - r_2| > d$ 인 관계식이 만족하는 경우 두 원은 교점을 갖지 않는다. 전자는 두 원이 멀리 떨어져 있어 만나지 않는 경우이고 후자는 한 원이 다른 원의 내부에 위치하여 공유점이 없는 경우이다.



두 원이 가질 수 있는 위치 관계는 위 자료에 나왔듯이 총 3가지이다. 이 중 ‘두 원이 겹친다’라고 할 수 있는 경우는 1번, 두 점에서 만나는 경우와 2번, 접하는 경우 중에서도 중심과 반지름이 전부 일치하는 경우이다. 1번 경우인  $|r_1 - r_2| < d < r_1 + r_2$  라는 관계식을 구현하기 위해서 전부 제곱을 해주면 더 쉽게 코드를 작성할 수 있다.

```

import java.util.Scanner;

public class Hw1_1 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int x1, y1, r1, x2, y2, r2;
        System.out.print("첫 번째 원의 중심과 반지름 입력 >> ");
        x1 = input.nextInt();
        y1 = input.nextInt();
        r1 = input.nextInt();
        System.out.print("두 번째 원의 중심과 반지름 입력 >> ");
        x2 = input.nextInt();
        y2 = input.nextInt();
        r2 = input.nextInt();

        int d = (x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2);
        int minus = (r1 - r2) * (r1 - r2);
        int plus = (r1 + r2) * (r1 + r2);

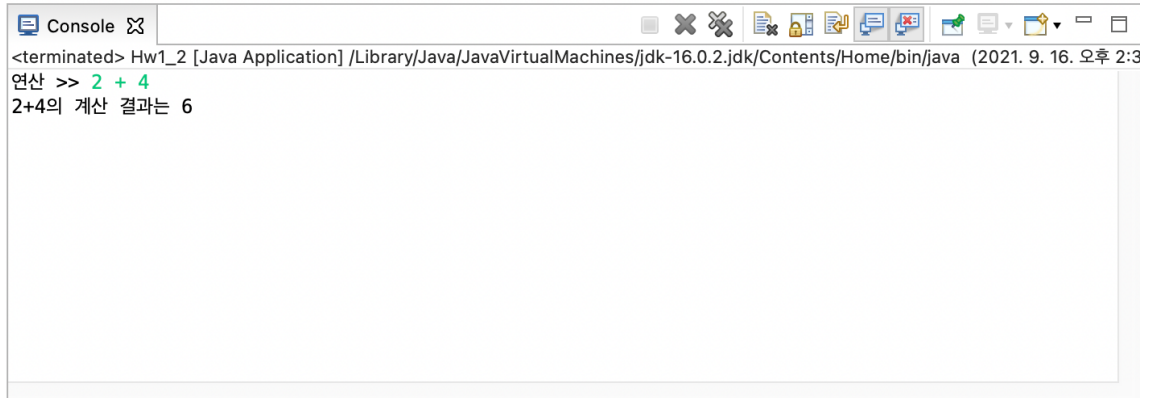
        if ((d > minus && d < plus) || ((x1 == x2)&&(y1 == y2)&&(r1 ==
r2)))

            System.out.println("두 원은 서로 겹친다.");
        else
            System.out.println("두 원은 서로 겹치지 않는다.");

    }
}

```

< Hw1\_2 >



문자형 변수를 입력받고 싶었지만 복잡해서 String 형으로 String o = input.next(); 이렇게 받았더니 정상적으로 작동되었다.

```
import java.util.Scanner;
```

```
public class Hw1_2 {

    public static void main(String[] args) {
        System.out.print("연산 >> ");
        Scanner input = new Scanner(System.in);
        int a = input.nextInt();
        String o = input.next();
        int b = input.nextInt();
        if (o.equals("/") && b == 0)
            System.out.println("0으로 나눌 수 없습니다");
        else {
            switch (o) {
                case "+":
                    System.out.println(a + "+" + b + "의 계산 결과는 " + (a + b));
                    break;

                case "-":
                    System.out.println(a + "-" + b + "의 계산 결과는 " + (a - b));
                    break;

                case "*":
                    System.out.println(a + "*" + b + "의 계산 결과는 " + (a * b));
                    break;

                case "/":
                    System.out.println(a + "/" + b + "의 계산 결과는 " + (a / b));
                    break;

            }
        }
    }
}
```

## < Hw1\_3 >

```
Console [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java (2021. 9. 16. 오후 2:3
정수 몇개? 24
97 3 42 72 68 41 79 99 43 94
49 26 83 57 70 82 29 73 15 76
60 98 39 75
```

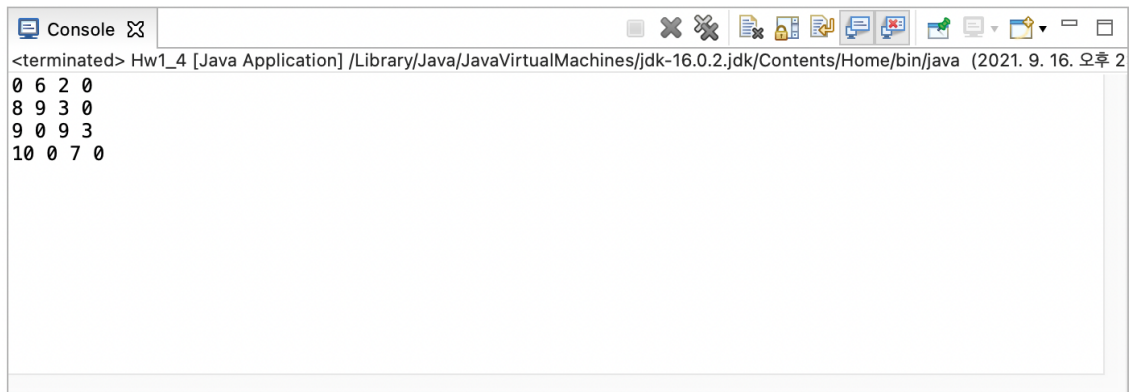
중복 제거를 위하여 랜덤으로 배정받은 1~99 사이의 수와 지금까지 배열에 저장한 수와 중복되지 않으면 배열에 추가를 하도록 하였다.

```
import java.util.Scanner;
```

```
public class Hw1_3 {

    public static void main(String[] args) {
        System.out.print("정수 몇개? ");
        Scanner input = new Scanner(System.in);
        int n = input.nextInt();
        int[] x = new int[n];
        int tmp;
        for (int i = 0; i < n; i++) {
            while (true) {
                tmp = (int) (Math.random() * 99 + 1);
                boolean flag = false;
                for (int j : x) {
                    if (j == tmp) flag = true;
                }
                if (flag == false)
                    break;
            }
            x[i] = tmp;
        }
        int cnt = 0;
        for (int i : x) {
            System.out.print(i + " ");
            cnt++;
            if (cnt % 10 == 0)
                System.out.println();
        }
    }
}
```

## < Hw1\_4 >



```
<terminated> Hw1_4 [Java Application] /Library/Java/JavaVirtualMachines/jdk-16.0.2.jdk/Contents/Home/bin/java (2021. 9. 16. 오후 2)
0 6 2 0
8 9 3 0
9 0 9 3
10 0 7 0
```

처음에 `index_i = (int)(Math.random()*3);`, `index_j = (int)(Math.random()*3);`으로 했더니 무한 루프에 빠졌다. 범위가 0~2까지만 도는 거여서 중복되지 않는 인덱스는  $3 \times 3 = 9$ 개밖에 가질 수가 없는데 10번을 돌려서 오류가 난 것이었다. 그리고 이중 for each문에서 `for(int j : x[i])`라고 하니까 오류가 났는데, `i` 자체가 `x`의 그 한 줄 배열 자체여서 `i`로 써줘야 한다.

```
import java.util.Scanner;
```

```
public class Hw1_4 {

    public static void main(String[] args) {
        int x[][] = new int[4][4];
        int index_i, index_j;
        for(int i = 0 ; i < 10; i++) {
            while(true) {
                index_i = (int)(Math.random()*4);
                index_j = (int)(Math.random()*4);
                if (x[index_i][index_j] == 0) break;
            }
            x[index_i][index_j] = (int)(Math.random()*10 + 1);
        }
        for(int[] i : x) {
            for(int j : i) {
                System.out.print(j + " ");
            }
            System.out.println();
        }
    }
}
```