

연습과제 1_ 1930022 최하원

<연습문제 4>

```
import java.util.Scanner;

class Odd {
    public int calculateOddSum(int n) {
        int sum = 0;
        for (int i = 1; i <= n; i += 2) {
            sum += i;
        }
        return sum;
    }
}

class OddEven extends Odd {
    public int calculateEvenSum(int n) {
        int sum = 0;
        for (int i = 2; i <= n; i += 2) {
            sum += i;
        }
        return sum;
    }
}

public class OddEvenTest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("숫자를 입력하세요: ");
        int n = scanner.nextInt();

        OddEven calculator = new OddEven();

        int oddSum = calculator.calculateOddSum(n);
        int evenSum = calculator.calculateEvenSum(n);

        System.out.println("짝수의 합: " + evenSum);
        System.out.println("홀수의 합: " + oddSum);

        scanner.close();
    }
}
```

<결과>

<terminated> OddEvenTest	<terminated> OddEvenTest
숫자를 입력하세요: 7	숫자를 입력하세요: 33
짝수의 합: 12	짝수의 합: 272
홀수의 합: 16	홀수의 합: 289

<연습문제 2>

```
import java.util.Scanner;

class SimpleCircle {
    static final double PI = 3.14159;
}

public class SimpleCircleTest1 {

    public static void main(String args[]) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("원의 반지름을 입력하세요: ");
        int radius = scanner.nextInt();

        double area = radius * radius * SimpleCircle.PI ;

        System.out.println("원의 반지름은: " + radius);
        System.out.println("원의 넓이는: " + area);

        scanner.close();

    }
}
```

<결과>

<terminated> SimpleCircleTest1	<terminated> SimpleCircleTest1
원의 반지름을 입력하세요: 10	원의 반지름을 입력하세요: 2
원의 반지름은: 10	원의 반지름은: 2
원의 넓이는: 314.159	원의 넓이는: 12.56636

<Section 5>

1. 메소드 오버로딩에 관해 간략하게 설명하시오.

➔ 같은 이름의 메소드를, arguments 의 개수와 데이터 타입을 바꾸어 여러 개 정의하는 방법이다. 코드의 가독성을 높이고 사용자가 더 직관적으로 메소드를 호출할 수 있다.

2. 두 수(프로그램에서 값을 지정)를 비교하여 큰 수를 출력하는 프로그램을 작성하시오. 두 수를 비교하는 메소드 max()를 정수(int)와 실수(double)로 구분하여 오버로딩 기법으로 구현하시오.

```
3. public class Comparison {
4.
5.     // 정수형 매개변수를 받는 max 메소드
6.     public static int max(int num1, int num2) {
7.         return (num1 > num2) ? num1 : num2;
8.     }
9.
10.    // 실수형 매개변수를 받는 max 메소드
11.    public static double max(double num1, double num2) {
12.        return (num1 > num2) ? num1 : num2;
13.    }
14.
15.    public static void main(String[] args) {
16.        int intNum1 = 25;
17.        int intNum2 = 20;
18.        double doubleNum1 = 15.5;
19.        double doubleNum2 = 18.3;
20.
21.        // 정수를 비교하는 max 메소드 호출
22.        int maxInt = max(intNum1, intNum2);
23.        System.out.println("두 정수 중 큰 수는: " + maxInt);
24.
25.        // 실수를 비교하는 max 메소드 호출
26.        double maxDouble = max(doubleNum1, doubleNum2);
27.        System.out.println("두 실수 중 큰 수는: " +
28.            maxDouble);
29.    }
```

<결과>

<terminated> Comparison [J]

두 정수 중 큰 수는: 25

두 실수 중 큰 수는: 18.3