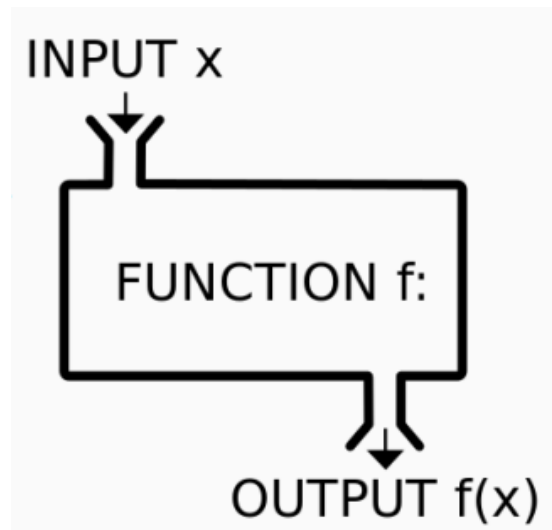


23.03.09 4강

메소드

코드의 부품화 하는 방법 중 한가지가 메소드이다

코드가 길어지거나, 같은 코드를 반복해야 할 때, 코드를 부품화하여 분리해 보기 좋은 코드로 만들 수 있다.



유지 보수, 재사용성, 사람의 실수로 인한 오류에 강하다.

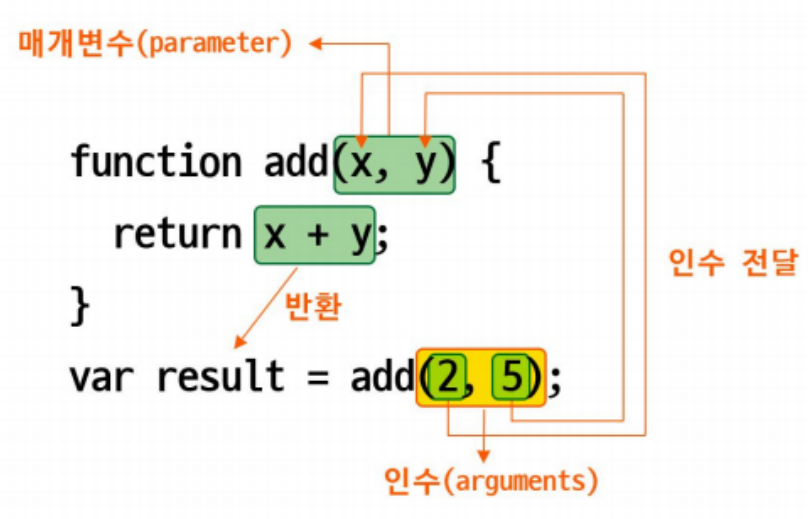
엄밀히 함수와 메소드는 다르지만, 일반적으로 비슷하다 여긴다.

호출

메소드를 실행한다.

main 메소드와 그 외 다른 메소드에서 호출이 가능하다.

인수(argument), 인자(parameter)



argument는 각각의 실행에서 메소드에 넣을 값, parameter는 각각의 argument가 들어가야하는 자리라고 생각하면 된다.

$$f(x, y) = (x + y)^2$$

$$f(2, 3) = (2 + 3)^2 = 25$$

예를 들어 위와 같은 메소드에서, 변수 x,y는 parameter 이고, 2,3은 argument라고 할 수 있다.

argument는 정확한 위치에 빠짐없이 넣어야한다. → 오버로딩을 고려하지 않는 경우

Return (반환값)

메소드 호출시 결과 값을 돌려주는 경우, 이 값을 반환 값 (return) 이라고 한다.

리스트 5-7 리턴 값의 예

survivalcoding.co

```
1 public class Main {
2     public static void main(String[] args) {
3         int ans = add(100, 10);
4         System.out.println("100 + 10 = " + ans);
5     }
6
7     private static int add(int x, int y) {
8         int ans = x + y;
9         return ans;
10    }
11 }
```

리턴값을 바로 argument로 사용이 가능하다.

리스트 5-8 리턴 값을 그대로 사용

survivalcoding.com

```
1 public class Main {
2
3     private static int add(int x, int y) {
4         int ans = x + y;
5         return ans;
6     }
7
8     public static void main(String[] args) {
9         System.out.println(add(add(10, 20), add(30, 40)));
10    }
11 }
```

return 문은 가급적 맨 마지막에 위치해야한다.

참조형 리턴도 가능하다. ex) 리턴 형식이 배열 등...

오버로딩

같은 이름의 메소드를 여러개 정의 하는 것

인수의 타입이나, 갯수가 다르게 설정할 수 있다.

call by reference vs call by value

리스트 5-12 배열이 인수일 때 (call by reference)

surviva

```
1 public class Main {
2     // int형 배열을 받아 모든 요소를 출력
3     public static void printArray(int[] array) {
4         for (int element : array) {
5             System.out.println(element);
6         }
7     }
8
9     public static void main(String[] args) {
10        int[] array = { 1, 2, 3 };
11        printArray(array);
12    }
13 }
```

리스트 5-13 같은 배열을 참조하고 있다는 것을 확인

survivalco

```
1 public class Main {
2     // int형 배열을 받아 배열내의 요소 전부 1을 더하는 메소드
3     public static void incArray(int[] array) {
4         for (int i = 0; i < array.length; i++) {
5             array[i]++;
6         }
7     } // 계산 결과를 리턴하지 않음
8
9     public static void main(String[] args) {
10        int[] array = { 1, 2, 3 };
11        incArray(array);
12
13        for (int i : array) {
14            System.out.println(i);
15        }
16    }
17 }
```

커맨드 라인 인수

```
public static void main(String[] args) {...}
```

커맨드 라인 인수는 String 형만 가능하다.

이를 int나 double로 변환하여 응용 가능하다.

중요 팁 - 메소드

메소드 내에서 println과 같은 리턴이 있는 메소드를 호출하지 말 것.

계산부와 출력부를 분리시킬 것

연습문제

연습문제 5-1

survivalco

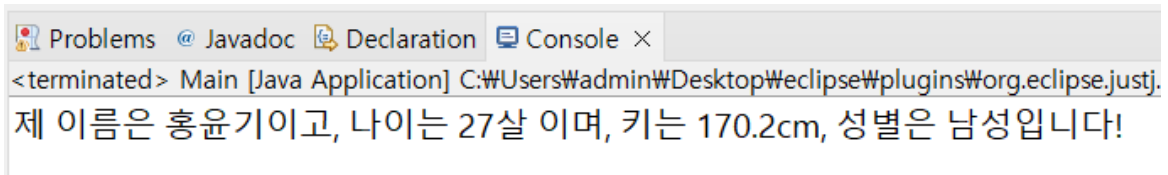
다음 사양을 참고하여 메소드 "introduceOneself" 를 정의하시오

메소드명	introduceOneself
리턴 값	없음
인수 항목	없음
처리내용	이름(문자열), 나이(정수), 키(부동소수점), 성별(1문자)를 대입하는 변수를 선언하고 값을 대입한다. 변수를 이용하여 자기소개를 표시한다. 표시할 데이터의 내용이나 방법은 자유롭게 정한다.

```
package com.example;

public class Main {
    public static void introduceOneself() {
        String name = "홍윤기";
        int age = 27;
        double height = 170.2;
        String gender = "남";
        System.out.println("제 이름은 " + name + "이고, 나이는 " + age + "살 이며, 키는 " + height + "cm, 성별은 "
            + gender + "성입니다!");
    }

    public static void main(String[] args) {
        introduceOneself();
    }
}
```



연습문제 5-2

survivalcod

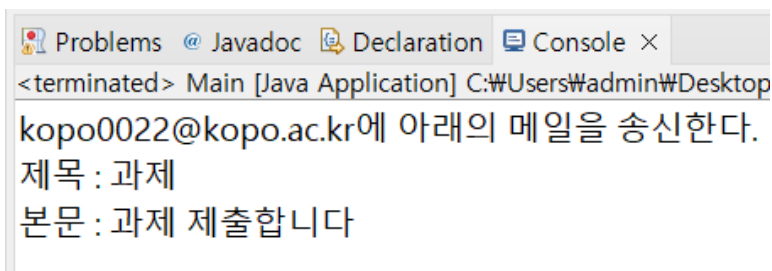
다음 사양을 참고하여 메소드 “email” 를 정의하시오

메소드명	email
리턴 값	없음
인수 항목	메일의 제목(String title) 메일을 받는 주소 (String address) 메일 본문 (String text)
처리내용	아래의 형식으로 표시를 한다. (빨간색 문자의 부분은 인수를 사용한다) 메일을 받는 주소 에 아래의 메일을 송신한다. 제목 : 메일의 제목 본문 : 메일 본문

```
package com.example;

public class Main {
    public static void email(String title, String address, String text) {
        System.out.println(address + "에 아래의 메일을 송신한다.");
        System.out.println("제목 : " + title);
        System.out.println("본문 : " + text);
    }

    public static void main(String[] args) {
        email("과제", "kopo0022@kopo.ac.kr", "과제 제출합니다");
    }
}
```



다음 사양을 참고하여 연습문제 5-2 의 코드에 메소드 “email” 를 오버로드하여 main메소드에서 호출하시오

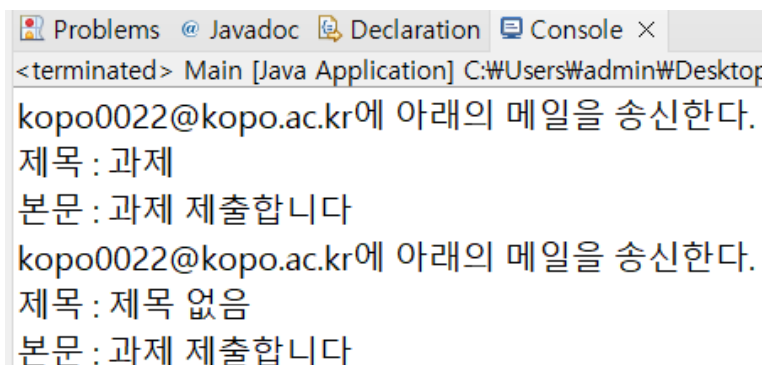
메소드명	email
리턴 값	없음
인수 항목	메일을 받는 주소 (String address) 메일 본문 (String text)
처리내용	아래의 형식으로 표시를 한다. (빨간색 문자의 부분은 인수를 사용한다) 메일을 받는 주소 에 아래의 메일을 송신한다. 제목 : 제목 없음 본문 : 메일 본문

```
package com.example;

public class Main {
    public static void email(String title, String address, String text) {
        System.out.println(address + "에 아래의 메일을 송신한다.");
        System.out.println("제목 : " + title);
        System.out.println("본문 : " + text);
    }

    public static void email(String address, String text) {
        email("제목 없음", address, text);
    }

    public static void main(String[] args) {
        email("과제", "kopo0022@kopo.ac.kr", "과제 제출합니다");
        email("kopo0022@kopo.ac.kr", "과제 제출합니다");
    }
}
```



```
Problems @ Javadoc Declaration Console ×
<terminated> Main [Java Application] C:\Users\Wadmin\Desktop
kopo0022@kopo.ac.kr에 아래의 메일을 송신한다.
제목 : 과제
본문 : 과제 제출합니다
kopo0022@kopo.ac.kr에 아래의 메일을 송신한다.
제목 : 제목 없음
본문 : 과제 제출합니다
```

다음 사양을 참고하여 메소드 “calcTriangleArea” 와 “calcCircleArea” 를 작성하시오.

적당한 값을 인수로 넘겨 올바른 면적이 표시되는지 확인하시오.

메소드명	calcTriangleArea
리턴 값	삼각형의 면적 (double)
인수 항목	삼각형의 밑변의 길이. 단위는 cm (double bottom) 삼각형의 높이. 단위는 cm (double height)
처리내용	인수를 사용하여 면적을 구하고, 그것을 반환 함

메소드명	calcCircleArea
리턴 값	원의 면적 (double)
인수 항목	원의 반지름. 단위는 cm (double radius)
처리내용	인수를 사용하여 면적을 구하고, 그것을 반환 함

```
package com.example;

import java.lang.Math;

public class Main {
    public static double calcTriangleArea(double bottom, double height) {
        double area = bottom * height / 2;
        return area;
    }

    public static double calcCircleArea(double radius) {
        double area = Math.pow(radius, 2) * Math.PI;
        return area;
    }

    public static void printArea(String figure, double area) {
        System.out.println(figure + "의 넓이는 : " + area + "cm^2");
    }

    public static void main(String[] args) {
        printArea("삼각형", calcTriangleArea(15.5, 4));
        printArea("원", calcCircleArea(3));
    }
}
```

Problems @ Javadoc Declaration Console ×

<terminated> Main [Java Application] C:\Users\wadmin\

삼각형의 넓이는 : 31.0cm^2

원의 넓이는 : 28.274333882308138cm^2

quiz

홀짝

```
package com.example;

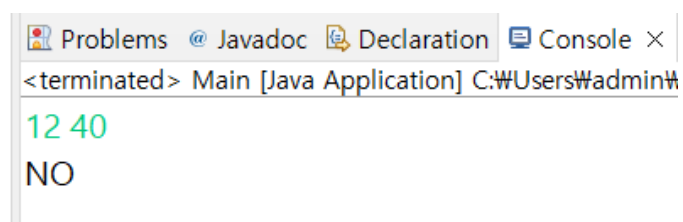
import java.util.*;

public class Main {
    public static int[] getInputs() {
        Scanner scanner = new Scanner(System.in);
        String[] inputs = scanner.nextLine().split(" ");
        scanner.close();
        int count = inputs.length;
        int[] answer = new int[count];
        for (int i = 0; i < count; i++) {
            answer[i] = Integer.parseInt(inputs[i]);
        }
        return answer;
    }

    public static String determine(int[] answer) {
        String result = "YES";
        int sum = 0;
        for (int num : answer) {
            sum += num;
        }
        if (sum % 2 == 0) {
            result = "NO";
        }
        return result;
    }

    public static void showOutput(String result) {
        System.out.println(result);
    }

    public static void main(String[] args) {
        showOutput(determine(getInputs()));
    }
}
```



등차수열

```
package com.example;

import java.util.*;
import java.lang.StringBuilder;

public class Main {
    public static int[] getInputs() {
        Scanner scanner = new Scanner(System.in);
        String[] inputs = scanner.nextLine().split(" ");
        scanner.close();
        int count = inputs.length;
```



```

        int[] answer = new int[count];
        for (int i = 0; i < count; i++) {
            answer[i] = Integer.parseInt(inputs[i]);
        }
        return answer;
    }

    public static int[] generate(int[] answer, int count) {
        int[] result = new int[count];
        int d = answer[1];
        result[0] = answer[0];
        for (int i = 1; i < count; i++) {
            result[i] = result[i - 1] + d;
        }
        return result;
    }

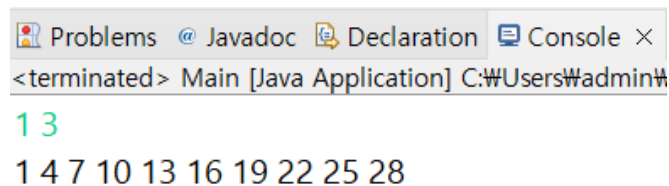
    public static int[] generate(int[] answer) {
        return generate(answer, 10);
    }

    public static StringBuilder converte(int[] result) {
        StringBuilder stringBuilder = new StringBuilder("");
        int count = result.length;
        for (int i = 0; i < count; i++) {
            stringBuilder.append(Integer.toString(result[i]));
            if (i != count - 1) {
                stringBuilder.append(" ");
            } else {
                stringBuilder.append("\n");
            }
        }
        return stringBuilder;
    }

    public static void showOutput(StringBuilder stringBuilder) {
        System.out.println(stringBuilder);
    }

    public static void main(String[] args) {
        showOutput(converte(generate(getInputs())));
    }
}

```



```

<terminated> Main [Java Application] C:\Users\admin#
1 3
1 4 7 10 13 16 19 22 25 28

```