

Java Programming

— Method —

Waseda University

Example

When the values of height and weight are given using the keyboard, display the standard weight and the message referring the following table.

Weight – Standard Weight	Message
Less than -10	Underweight
More than 10	Overweight
Otherwise	Normal

Equation for calculating Std. weight:

$$\text{Standard Weight(kg)} = \text{Height(m)}^2 \times 22$$

Method

In previous lessons ...

```
public class ClassName {  
    public static void main(String[] args) {  
        Main Body  
    }  
}
```

- Wrote statements in **Main Body**
- `public static void main(String[] args)` is also a method. (Main method)
- A Java method is a collection of statements that are grouped together to perform an operation.
- A Java method can be helpful when you write a large program.

Example : How to write and call a method

```
public class ClassName {
    public static void main(String[] args) {
        :
        c = method1(a, b);
        :
        method2(x);
        :
    }

    public static int method1(int a, int b) {
        ....
    }

    public static void method2(double x) {
        ....
    }
}
```

Example : How to write and call a method

```
public class ClassName {  
    public static void main(String[] args) {  
        :  
        c = method1(a, b);  
        :  
        method2(x);  
        :  
    }  
  
    public static int method1(int a, int b) {  
        .....  
    }  
  
    public static void method2(double x) {  
        .....  
    }  
}
```

Example : How to write and call a method

```
public class ClassName {
    public static void main(String[] args) {
        :
        c = method1(a, b);
        :
        method2(x);
        :
    }

    public static int method1(int a, int b) {
        ....
    }

    public static void method2(double x) {
        ....
    }
}
```

Example : How to write and call a method

```
public class ClassName {
    public static void main(String[] args) {
        :
        c = method1(a, b);
        :
        method2(x);
        :
    }

    public static int method1(int a, int b) {
        ....
    }

    public static void method2(double x) {
        ....
    }
}
```

Summary of a previous example

- Methods appear inside class bodies.
- In this example, the methods `method1` and `method2` are defined in the class `ClassName`.
- These methods are called by the method `main`.

Parameter and Return value

- Methods may have **parameter** and **return value**
- **Parameters** are supplied by the caller of the method.
- **Return value** is returned to the caller.

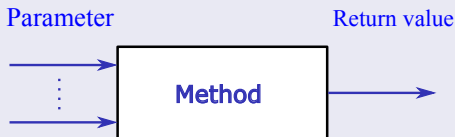


Image of a method

Method (1)

```
Modifier ReturnDataType MethodName(parameters) {  
    Main Body  
}
```

Modifier The modifiers (e.g., public or private) specify an accessibility. In this lesson, let Modifier be `public static`.

ReturnDataType A `ReturnDataType` is a data type of return value.

MethodName A `MethodName` is a name of a Method. `MethodName` should be easy to understand.

parameters When the method require parameters, you write Data types and variable names. You can write an arbitrary number of Data types and variable names as `parameters`.

Method (2)

```
return ReturnValue;
```

- **ReturnValue** : A return value is returned to the caller.

Example of a return value

```
public static double add(double x,double y){  
    double z;  
    z = x + y;  
    return z;  
}
```

Method (3)

Create the following method

[The method requires a value for height and
returns a value for the standard weight]

- Let the name of the method be `calcStdWeight`.
- Let the data type of the parameter be `double` and the name be `height`.
- Let the data type of the return value be `double`.

Equation for calculating Std. weight:

$$\text{Standard Weight(kg)} = \text{Height(m)}^2 \times 22$$

Method (4)

calcStdWeight

```
public static double calcStdWeight(double height) {  
    double weight;  
    height = height / 100;  
    weight = height * height * 22;  
    return weight;  
}
```

Examples of methods

```
public static int sampleMethod1(int a, int b)
```

- requires two `int` values
- returns `int` value.

```
public static double sampleMethod2(int a, double x)
```

- requires `int` value and `double` value
- returns `double` value.

```
public static boolean sampleMethod3(int a, int b, int c)
```

- requires three `int` values
- returns `boolean` value.

```
public static void sampleMethod4(double x)
```

- requires `double` values
- returns no values.

```
public static double sampleMethod5()
```

- requires no parameters
- returns `double` value.

Calling methods (1)

Calling the method `calcStdWeight`

```
sw = calcStdWeight(170.0);
```

- Input 170.0 as a value for a height.
- `calcStdWeight(170.0)` returns the Std. weight of 170cm.
- Assign the return value from `calcStdWeight(170.0)` to `sw`.

```
h = 170.0;
```

```
sw = calcStdWeight(h);
```

- You can also use variables as parameter arguments.

Calling methods (2)

The flow of calling the method `calcStdWeight`

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

```
...
```

```
h = 170.0;
```

```
sw = calcStdWeight(h);
```

```
...
```

```
}
```

170.0

```
public static double calcStdWeight(double height) {
```

```
double weight;
```

```
height = height / 100;
```

```
weight = height * height * 22;
```

```
return weight;
```

```
}
```

63.58

printMessage (1)

Create the following method.

The method requires the weight and the Std. weight and displays the message.

- Let the method name be `printMessage`.
- The parameters are two `double` variables for the weight and the Std. weight.
- The method `printMessage` returns no value.

printMessage (2)

Method : printMessage

```
public static void printMessage(double weight, double stdWeight) {  
    double diff = weight - stdWeight;  
    if (diff < -10) {  
        System.out.println("Underweight.");  
    } else if (diff < 10) {  
        System.out.println("Normal.");  
    } else {  
        System.out.println("Overweight.");  
    }  
}
```

printMessage (3)

Calling the method which returns no values

```
h = 170.0;
w = 70.0
sw = calcStdWeight(h); — The return value is assigned to sw
printMessage(w, sw); — The printMessage returns no value
```

- If a method returns no value, then we only call a method.

Local variable

```
public static void main(String[] args) {
    double x, y;
    ....
    y = method1(x);
    z = method1(y);
    ....
}

public static double method1(double t){
    double x, y, z;
    ...
}
```

— error

- **z** can be accessed only in the method `method1`
- If **z** is accessed in other methods, then an error occurs.
- If **z** is defined in other methods, this **z** is not the **z** in the method `method1`.

How to write the method which requires an array

```
public static int maxOfArray(int[] a)
```

— requires a `int` array

returns a `int` value which is the maximum of an array.

```
public static int maxOfArray(int[] a) {  
    int i, max;  
  
    max = a[0];  
    for (i=1; i<a.length; i++) {  
        if (a[i] > max) max = a[i];  
    }  
    return max;  
}
```

- Let `arr` be array of integer.
- The following code shows how to call the `maxOfArray`.

```
max = maxOfArray(arr);
```

How to write the method which returns an array

```
public static double[] ones(int n)
```

- requires a `int` value for the length of an array
- returns an array of n double values.

```
public static double[] ones(int n) {  
    int i;  
    double[] x = new double[n];  
  
    for (i=0; i<x.length; i++) {  
        x[i] = 1.0;  
    }  
    return x;  
}
```

- The valuable `n` means the length of an array.
- The following code shows how to call the `ones`

```
double[] x;  
x = ones(5);
```

How to get a keyboard input (1)

```
public static int inputInt()
```

— requires no value

returns the `int` value from a keyboard input.

```
public static int inputInt() {  
    Scanner sc = new Scanner(System.in);  
    int i = 0;  
  
    i = sc.nextInt();  
    return i;  
}
```

- The following code shows how to call the `inputInt`

```
a = inputInt();
```

How to get a keyboard input (2)

```
public static double inputDouble()
```

— requires no value

returns the `double` value from a keyboard input.

```
public static double inputDouble() {  
    Scanner sc = new Scanner(System.in);  
    double d = 0.0;  
  
    d = sc.nextDouble();  
    return d;  
}
```

- The following code shows how to call the `inputDouble`

```
x = inputDouble();
```

How to get a keyboard input (3)

SampleInput.java

```
import java.util.Scanner;           ← This statement is necessary.
```

```
public class SampleInput {
    public static void main(String[] args) {
        System.out.print("Input an int : ");
        a = inputInt();
        System.out.print("Input a double : ");
        x = inputDouble();
    }
    public static int inputInt() {
        ....
    }
    public static double inputDouble() {
        ....
    }
}
```


Methods

Weight3.java

```
import java.util.Scanner;

public class Weight3 {
    public static void main(String[] args) {
        ....
    }

    public static double calcStdWeight(double height) {
        ....
    }

    public static void printMessage(double weight, double stdWeight) {
        ....
    }

    public static double inputDouble() {
        ....
    }
}
```

Example : Main method

```
public static void main(String[] args) {
    double h, sw, w;
    System.out.print("Input height (cm) : ")
    h = inputDouble();

    System.out.print("Input weight (kg) : ")
    w = inputDouble();

    sw = calcStdWeight(h);
    System.out.println("Standard weight : " + sw + "kg");

    printMessage(w, sw);
}
```

calcStdWeight

- requires the height (**double**)
- returns the Std. Weight(**double**).

printMessage

- requires the weight (**double**) and the Std. Weight(**double**)
- displays the message.

inputDouble

- returns the value(**double**) from the keyboard.

calcStdWeight

```
public static double calcStdWeight(double height) {
    double stdWeight;

    height = height / 100;
    stdWeight = height * height * 22;
    return stdWeight;
}
```

printMessage

```
public static void printMessage(double weight, double stdWeight) {
    double diff;

    diff = weight - stdWeight;
    if (diff < -10) {
        System.out.println("Underweight.");
    } else if (diff < 10) {
        System.out.println("Normal.");
    } else {
        System.out.println("Overweight.");
    }
}
```

inputDouble

```
public static double inputDouble() {  
    Scanner sc = new Scanner(System.in);  
    double d = 0.0;  
  
    d = sc.nextDouble();  
    return d;  
}
```