Functions

Tobias Hanf, Maik Göken

November 21, 2022

Learn Programming with Java

Outline

Revision

Functions

Exercises

Revision

Small Quiz

https://pingo.coactum.de/647642

Functions

What are Functions?

$$\begin{aligned} f(x) &= x^2 \\ f(2) &= 4 \\ f(0) &= 0 \\ f(-1) &= 1 \\ f(\text{"}Hi\text{"}) &= ?? \end{aligned} \tag{1}$$

What are Functions?

What is function:

- Block of code
- Way of structuring your code
- Allows you to reuse code
- Can take arguments as Input
- Can provide a result as Ouput

Other Names

Other names:

- method
- procedure
- routine
- subprogram
- subroutine

Why use Functions?

Why should we use functions:

- Programmers are lazy -> less code is better

Why use Functions?

Why should we use functions:

- Programmers are lazy -> less code is better
- Makes programms more readable (better structur)

Why use Functions?

Why should we use functions:

- Programmers are lazy -> less code is better
- Makes programms more readable (better structur)
- Less changes needed

Why use Functions?

Why should we use functions:

- Programmers are lazy -> less code is better
- Makes programms more readable (better structur)
- Less changes needed
- Reduces errors/bugs

Why use Functions?

Why should we use functions:

- Programmers are lazy -> less code is better
- Makes programms more readable (better structur)
- Less changes needed
- Reduces errors/bugs
- Important for OOP
    - (and almost all other paradigms)

Functions in Java

```
1 // function delaration and implementation
2 <ret-type> <func-name>(<para-type> <para-name>, ...){
3     // function body
4     <code>
5     return <expression>;
6 }
7
8 // function call
9 <func-name>(<argument>,...);
```

- `ret-type`: type which will be returned by the function
  - `void` if no value is returned
- `func-name`: name of the functions
  - same rules as for variables names
- `para-type`: type of the parameter
- `para-name`: name of the parameter

Functions in Java

- Every has a return type
  - may be `void` if no value is returned

Functions in Java

- Every has a return type
  - may be `void` if no value is returned
- Every function has a name
  - should be unique
  - but can be the same in special cases (later unit)

Functions in Java

- Every has a return type
  - may be `void` if no value is returned
- Every function has a name
  - should be unique
  - but can be the same in special cases (later unit)
- Every function has 0 or more parameters

Functions in Java

- Every has a return type
  - may be `void` if no value is returned
- Every function has a name
  - should be unique
  - but can be the same in special cases (later unit)
- Every function has 0 or more parameters
- Every parameter has a type and name
  - name must be unique inside the list

Functions in Java

- Every has a return type
  - may be `void` if no value is returned
- Every function has a name
  - should be unique
  - but can be the same in special cases (later unit)
- Every function has 0 or more parameters
- Every parameter has a type and name
  - name must be unique inside the list
- Parameters can be used like normal variables

Functions in Java

- Every has a return type
  - may be `void` if no value is returned
- Every function has a name
  - should be unique
  - but can be the same in special cases (later unit)
- Every function has 0 or more parameters
- Every parameter has a type and name
  - name must be unique inside the list
- Parameters can be used like normal variables
- return retruns the value to the Output
  - and terminates the execution of the function
  - can have multiple `retrun`

Functions in Java

- Every has a return type
  - may be `void` if no value is returned
- Every function has a name
  - should be unique
  - but can be the same in special cases (later unit)
- Every function has 0 or more parameters
- Every parameter has a type and name
  - name must be unique inside the list
- Parameters can be used like normal variables
- return retruns the value to the Output
  - and terminates the execution of the function
  - can have multiple `retrun`
- Function call will be replaced with result

Example Function

```java
public static int add10_1(int n) {
    return n + 10;
}

public static int add10_2(int n) {
    n = n + 10;
    return n;
}

public static void main(String[] args) {
    int n = add10_1(3); // calling the function
    System.out.println("3+10 = ", n);
}
```

Scopes

- Every function has its own scope
- Variables declared inside a scope are only visible
  - inside the scope they were declared in
  - scopes inside the current scope (hierarchy)
- Important: parent scope of function
  - is scope of declaration
  - not scope of call
- Variables in the top most scope are called global Variables

Call by Reference vs Call by Value

Call by Value:

- Get a copy of the variable/value
- Changes in the function will no affect the outside variable
- Only done for primitive data types
    - `int`, `float`, `char`, ...

Call by Reference:

- Get a reference to the object
- Will affect the original object/value
- Done for everything else
    - `Object`, `String`, `Array`

Exercises

Sum of array

Declare and implement a function which takes an array of numbers as its argument and returns the sum of all values inside the array.

```
sum([30, 45, 10]) -> 85
```

Max Value of array

Declare and implement a function which takes an array of numbers as its argument and retruns the largest value of the array.

```
max([4,5,10,2,60,31]) -> 60
```

Use the same class/file as in the previous exercise

Sum of max values

Define and populate 3 or more arrays each containing atleat 3 different integers. Use the `sum` and `max` functions from the two previous execises to write a programm which extracts the max value of each array and stores it into a new array. The values of the new array should be summed up and printed to the console.

Use the same class/file as in the previous exercise