Java

Object Oriented Programming

Florian Kluge, Moritz Schulz

20. Oktober 2021

Florian.Kluge@mailbox.tu-dresden.de, Moritz.Schulz2@mailbox.tu-dresden.de

Overview

1. Organisation
2. Why Java?
3. Time to get started! ..almost
4. Let’s go!
5. That’s it (at least for today)

Organisation

Who are we?

Florian Kluge
Florian.Kluge@mailbox.tu-dresden.de

Moritz Schulz
Moritz.Schulz2@mailbox.tu-dresden.de
@schokotets auf Telegram

What are we doing here?

What are we doing here?

- Introduction to programming
- Getting to know the basics of Java
- Preparation for upcoming courses (e.g ’Softwaretechnologie’, 2nd Semester)

Structure

- 14 lessons

Structure

- 14 lessons
- Thursday, 13:00 - 14:30

Structure

- 14 lessons
- Thursday, 13:00 - 14:30
- APB/006 (right here)

Structure

- 14 lessons
- Thursday, 13:00 - 14:30
- APB/006 (right here)
- 3G-Rule

Structure

- 14 lessons
- Thursday, 13:00 - 14:30
- APB/006 (right here)
- 3G-Rule
- Attendance list

Structure

- This course is *voluntarily*

Structure

- This course is *voluntarily*
- Bored and want to leave? No problem!

Structure

- This course is *voluntarily*
- Bored and want to leave? No problem!

  .. but please contact us so we can invite students from the waiting list

Structure

- This course is *voluntarily*
- Bored and want to leave? No problem!

  .. but please contact us so we can invite students from the waiting list
- If you don’t attend the course for two weeks in a row without notice we will give your slot to other students

Our course philosophy

Our course philosophy

- We want to engage with *you*

Our course philosophy

Our course philosophy

- We want to engage with *you*
- We want to demonstrate theory by solving tasks

Our course philosophy

Our course philosophy

- We want to engage with *you*
- We want to demonstrate theory by solving tasks
- Don’t be afraid of mistakes

Our course philosophy

Our course philosophy

- We want to engage with *you*
- We want to demonstrate theory by solving tasks
- Don’t be afraid of mistakes
- No one is perfect

Our course philosophy

Our course philosophy

- We want to engage with *you*
- We want to demonstrate theory by solving tasks
- Don’t be afraid of mistakes
- No one is perfect
- We are also not perfect

Our course philosophy

Our course philosophy

- We want to engage with *you*
- We want to demonstrate theory by solving tasks
- Don’t be afraid of mistakes
- No one is perfect
- We are also not perfect
- Ask questions - always and every time!

Why Java?

Why Java?

- Widely used programming language
- Introduction to object oriented programming (OOP)
- Platform-independent
- ... and much more

Use cases

- Android development
- Web applications
- Desktop GUI applications
- … and much more

Do you have any programming experience already?

Do you have any programming experience already?

`https://trivo25.github.io/tud-java-course/poll.html`

or

`https://strawpoll.com/6uh45fcvx`

Time to get started! ..almost

Time to get started! ..almost

Java OpenJDK 11 https://adoptium.net/

Did you install it correctly? Time to find out!

```
        $ javac --version
        > javac 11.0.12

```

Time to get started! ..almost

Doesn’t work? :(Use an online compiler!

https://www.jdoodle.com/online-java-compiler/

Let’s go!

Your first task

- Create a new folder

Your first task

- Create a new folder
- Open the terminal and navigate into that folder using

  ```
  $ cd /to/my/folder
  ```

Your first task

- Create a new folder
- Open the terminal and navigate into that folder using

  ```
  $ cd /to/my/folder
  ```
- Create a new file by either typing

  ```
  $ touch helloWorld.java
  ```

  or right-clicking in your folder

  ```
  Right click -> new -> text document
  ```

  and save it as a `.java` file

Your first task

- now its time to write your first piece of code!

Your first task

- now its time to write your first piece of code!

```java
public class HelloWorld {
  public static void main (String[] args) {
    System.out.println("Hello World!");
  }
}
```

../code_samples/HelloWorld.java

How to execute a java program

what we have to do now..

- telling `javac` to compile our `helloWorld.java` file into a `helloWorld.class`

How to execute a java program

what we have to do now..

- telling `javac` to compile our `helloWorld.java` file into a `helloWorld.class`
- `.class` files are 'bytecode' for the Java Virtual Machine (JVM)

How to execute a java program

what we have to do now..

- telling `javac` to compile our `helloWorld.java` file into a `helloWorld.class`
- `.class` files are ’bytecode’ for the Java Virtual Machine (JVM)
- with `$ java helloWorld` we can finally execute our first program!

How to execute a java program

```
$ java helloWorld
> Hello World!

```

Time to play around

your next task

- change the text you want to print in the `helloWorld.java` file

Time to play around

your next task

- change the text you want to print in the `helloWorld.java` file
- re-compile it into a `.class` file and execute it again!

What are we actually doing?

- we are telling the computer what do to

What are we actually doing?

- we are telling the computer what do to
- we list instructions for the computer

Task numero 2!

Let’s add a variable of type `String`!

Task numero 2!

```java
public class VariableString {
  public static void main (String[] args) {
    // greeting is of type ’String’
    String greeting = "Hello"
    /*
      toGreet is also of type ’String’, but this comment is on multiple lines!
    */
    String toGreet = "everyone"
    System.out.println(greeting + " " + toGreet);
  }
}
```

../code_samples/VariableString.java

Task numero 2!

- We can re-use variables

Task numero 2!

- We can re-use variables
- We can store data in them

Task numero 3!

Let’s talk to the console and read our input!

Task numero 3!

```java
import java.util.Scanner;
public class VariableStringName {
  public static void main (String[] args) {
    Scanner myInputScanner = new Scanner(System.in);

    System.out.println("Hi, whats your name?");
    String name = myInputScanner.nextLine();
    System.out.println("Hello, " + name + " nice to meet you! :)");
  }
}
```

../code_samples/VariableStringName.java

Task numero 3!

- Besides `String`s we also have variables of type `int`

Task numero 3!

- Besides `String`s we also have variables of type `int`
- `int` represent whole numbers, like `1, 18, 1337 or 420360`

Task numero 3!

- Besides `String`s we also have variables of type `int`
- `int` represent whole numbers, like `1, 18, 1337 or 420360`
- We can calculate `int` with operators like +, −, * and many more

Task numero 3!

We now can..

- Display text in the console `System.out.println("Hello word!");`

Task numero 3!

We now can..

- Display text in the console `System.out.println("Hello word!");`
- Declare variables like `int` or `String`

Task numero 3!

We now can..

- Display text in the console `System.out.println("Hello word!");`
- Declare variables like `int` or `String`
- Read input from the conole

Task numero 3!

We now can..

- Display text in the console `System.out.println("Hello word!");`
- Declare variables like `int` or `String`
- Read input from the conole
- .. and know operators like `+, - or *`

Task numero 3!

We now can..

- Display text in the console `System.out.println("Hello word!");`
- Declare variables like `int` or `String`
- Read input from the conole
- .. and know operators like `+, - or *`

  Okay, what now?

Task numero 3!

We now can..

- Display text in the console `System.out.println("Hello word!");`
- Declare variables like `int` or `String`
- Read input from the conole
- .. and know operators like `+, - or *`

  Okay, what now?

  Let’s build a calculator!

That’s it (at least for today)

What will we do next lesson?

- Deep dive into (more) variables and their operators
- Introducing functions and control flow
- and build more cool things!

Links and resources

https://trivo25.github.io/tud-java-course/