

Das diesem Dokument zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 16OH21005 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor/bei der Autorin.

GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung

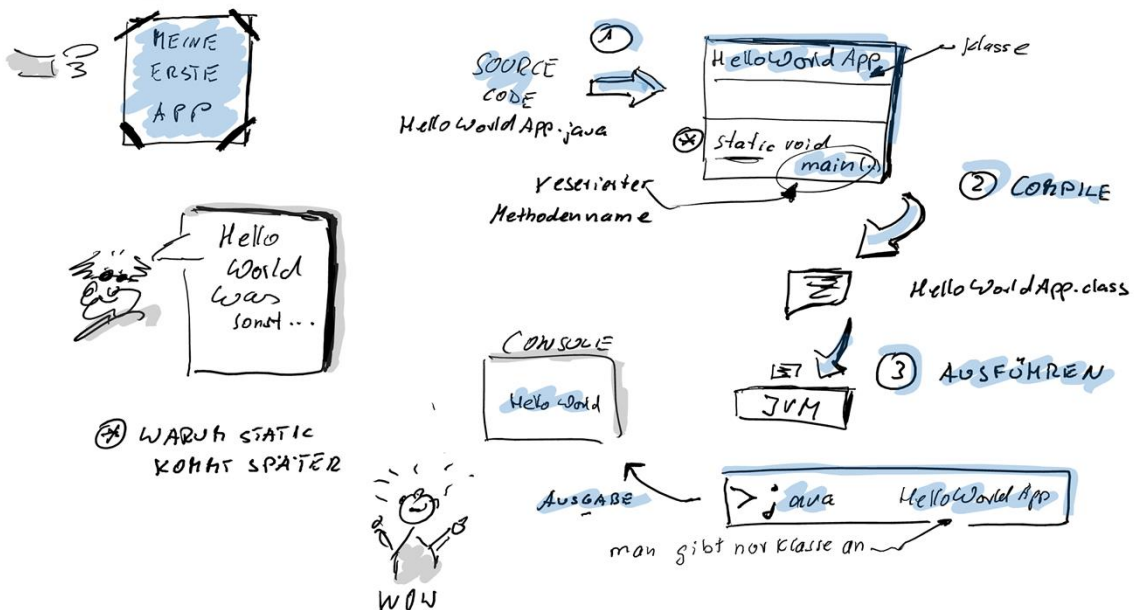


# Java meine erste App

## Grundlagen der Programmierung

- ➔ Beispiel einer einfachen Java App
- ➔ Die wichtigsten Elemente einer App

Um einen ersten Eindruck von Java zu bekommen und um eine Vorstellung von der Java-Syntax zu bekommen, schauen wir uns in diesem Abschnitt einmal ein sehr einfaches Java-Programm an. Dort werden wir auch gleich die wichtigsten Elemente einer Java-Anwendung kennenlernen.



18. Januar 2017

Modul 8338  
Grundlagen der Programmierung

- 3 -

Die einfachste App in Java ist eine „Hello-World“ App.

Dies ist eine App, die lediglich den Text „Hello World“ auf einer Konsole ausgibt.

Eine solche App besteht nur aus einer Java-Klasse und enthält lediglich den Quell-Code um den Text auf der Konsole auszugeben.

Um eine solche App zu erstellen, sind also folgende Schritte notwendig:

- 1) Java Quelle-Code erstellen
- 2) Java Quelle-Code übersetzen in eine Class-Datei
- 3) Ein JAR Datei einstellen, welche die Class-Datei und das entsprechende Manifest-File enthält.
- 4) Das Jar-file / die App mittels der JVM ausführen

Beginnen wir nun mit Schritt (1) und schauen uns einmal den Quell-Code in der nächsten Abbildung etwas genauer an.

**TIPP:**

\* Führen Sie hierzu die entsprechende LAB aus dem Übungsbuch durch.

```
1 package de.tespect.java1.basics;
2
3
4 /**
5  * TODO Lerninhalt: Hello World
6  *
7  * @author rje
8  *
9  */
10 public class HelloWorldApp {
11
12     /**
13      * @param args
14      */
15     public static void main(String[] args) {
16         System.out.println("Hello World");
17     }
18 }
19
20 }
21
```

18. Januar 2017 Modul 8338  
Grundlagen der Programmierung - 4 -

In dieser Abbildung sehen Sie den vollständigen Code der „Hello World“ App in Java.

Schauen wir uns den Code etwas genauer an:

### Zeile 1

Dort sehen wir eine PACKAGE-Anweisung mit einem Namen.

Der Name einer Package-Anweisung ist weltweit eindeutig. Er definiert einen Namensraum. Innerhalb des Namensraums (sprich innerhalb eines Packages) müssen die Klassennamen eindeutig sein. (→ Was genau eine Klasse ist, kommt später noch). Man kann sich leicht vorstellen, dass es nicht unwahrscheinlich ist, dass unterschiedliche Entwickler und Entwicklerinnen auf die Idee kommen, den gleichen Klassennamen zu erfinden.

**Zeile 4 bis 9** ist ein Kommentar

Die **Zeile 10** etwas genauer betrachtet besteht aus drei Angaben.

Public ← dies ist eine JAVA Schlüsselwort. Es gibt an, dass die Klasse

außerhalb des Packages sichtbar ein soll. Was diese genau bedeutet, sehen wir später noch.

Class ← dies gibt an, dass eine Klassendefinition folgt. Der Name der Klasse folgt unmittelbar. In unserem Beispiel ist der Klassenname „HelloWorldApp“.

{ ← Die geöffnete geschweifte Klammer zeigt an das ein Block beginnt. In diesem Fall ein Block, welches eine Klasse definiert.

**Zeile 12 bis 14** ist auch ein Kommentar

### **Zeile 15**

In dieser Zeile wird festgelegt, dass dort eine Methode mit dem Namen „main“ definiert wird.

Das Schlüsselwort „public“ zeigt hier wieder an, dass die Methode von anderen Klassen aufgerufen werden darf.

Das Schlüsselwort „static“ zeigt an, dass man für den Aufruf der Methode kein Objekt benötigt. Was ein Objekt ist kommt später noch genauer.

Alles was nach dem Namen der Methode „main“ steht und durch „(“ und „)“ begrenzt wird, sind die Parameter /Argumente, die man der Methode übergeben kann. Was dies genau bedeutet werden wir später kennenlernen.

### **Zeile 16**

In dieser Zeile beginnt der eigentliche Programmcode unserer Applikation. Diese Zeile besagt, dass der angegeben Text „Hello World“ auf der Konsole ausgegeben werden soll.

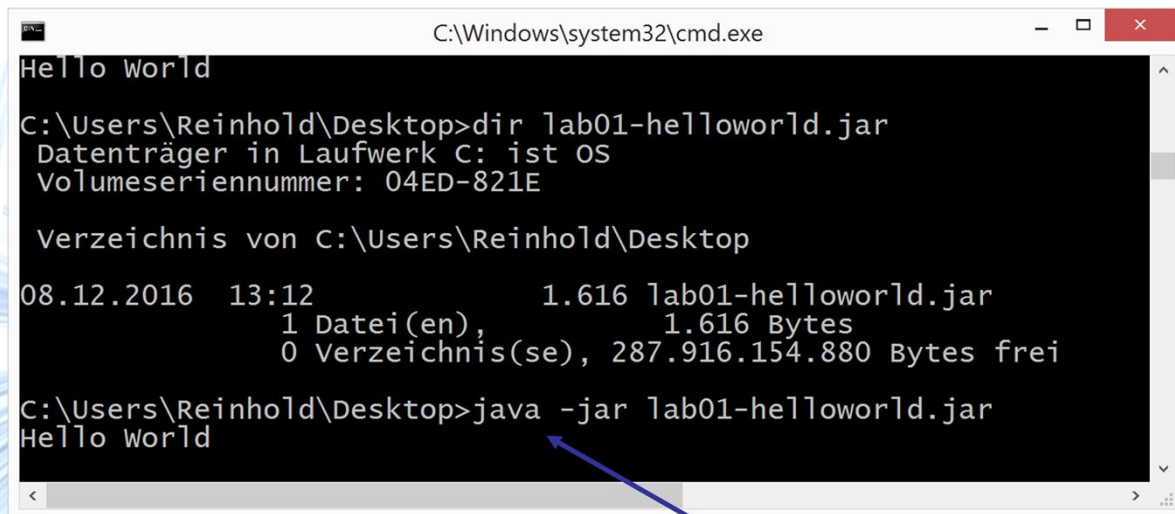
Was dieser Methodenaufruf genau bedeutet, werden wir ebenfalls später noch kennenlernen.

### **Zeile 20**

} ← Die schließende geschweifte Klammer zeigt an, dass der Block geschlossen wird, der zu der korrespondierende Klammer gehört, die den Block geöffnet hat.

Somit steht fest:

- Alle Zeilen zwischen **Zeile 10 und Zeile 20** gehören also zur Klassendefinition. Der Name der Klasse ist in Zeile 10 angegeben.



```
C:\Windows\system32\cmd.exe
Hello World
C:\Users\Reinhold\Desktop>dir lab01-helloworld.jar
Datenträger in Laufwerk C: ist OS
Volumeseriennummer: 04ED-821E

Verzeichnis von C:\Users\Reinhold\Desktop

08.12.2016  13:12                1.616 lab01-helloworld.jar
              1 Datei(en),                1.616 Bytes
              0 Verzeichnis(se), 287.916.154.880 Bytes frei

C:\Users\Reinhold\Desktop>java -jar lab01-helloworld.jar
Hello World
```

Start der JVM

Wenn wir nun eine JAR Datei erzeugen, dann können wir die App starten, wie in der Abbildung dargestellt.

Das Erzeugen einer JAR Datei erfolgt mit Hilfe von Eclipse sehr einfach.

- Exportiere deine App via Eclipse EXPORT als JAR-Datei
  - File->Export->JAR-File -> Projekt auswählen -> Next -> MAIN Class eintragen -> OK

TIPP:

\* Führen Sie nun die Übung „Hello World App“ aus dem Übungsbuch durch



