

Bestandteile einer einfachen Konsol-Anwendung

```
1 public class HalloWelt
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello World");
6     }
7 }
```

Zeile 1: **public class** HalloWelt

Erzeugung einer **Klasse** mit den Schlüsselwörtern **public** und **class**.

Sie erhält den Klassennamen **HalloWelt**.

Klassennamen werden groß geschrieben.

Wird eine Klasse als public deklariert, so muss der Dateiname der Quelldatei dem Namen der Klasse entsprechen, hier also **HalloWelt.java**

Ein Java-Programm kann aus einer oder mehreren Klassen bestehen.

Zeile 2 u. 7:

Der Inhalt einer Klasse wird in **geschweifte Klammern** gesetzt.

Öffnende und schließende Klammer gehören immer zusammen, hier 2 und 7. Sie werden jeweils in eine eigene Zeile geschrieben und stehen immer genau untereinander.

Achtung: hinter den Klammern kommt kein Semikolon.

Zeile 3: **public static void main**(String[] args)

Innerhalb der geschweiften Klammern der Klasse stehen die **Methoden dieser Klasse**. Eine Klasse kann mehrere Methoden enthalten.

Eine Methode hat einen Namen (hier: main) gefolgt von runden Klammern, in denen Parameter stehen können. In Methoden werden die Anweisungen des Programms geschrieben.

Eine Java-Anwendung braucht *genau eine* Methode `main` mit der Deklaration

`public static void`. (Erläuterungen dazu später.)

Der Interpreter führt alle Anweisungen der Methode `main` der Reihe nach aus. Das

Programm wird beendet, wenn die letzte Anweisung der Methode `main` ausgeführt wurde.

Zeile 4 u. 6:

Der Inhalt einer Methode (die Anweisungen) wird zwischen **geschweifte Klammern** gesetzt.

Öffnende und schließende Klammer gehören immer zusammen, hier 4 und 6. Sie werden jeweils in eine eigene Zeile geschrieben und stehen immer genau untereinander.

Zeile 5: **System.out.println**("Hello World");

Hier werden alle **Anweisungen** des Programms – der Programmcode - geschrieben.

In diesem Fall die Anweisung zum Ausgeben eines Textes am Bildschirm

(Eingabeaufforderung/Konsole).

Eine Anweisung wird mit einem Semikolon abgeschlossen.

Übersetzen und Ausführen einer Java Konsol-Anwendung in einer Windows Konsole (cmd bzw. "Eingabeaufforderung")

Im Allgemeinen werden Java Programme in einer IDE (Integrated Development Environment) erstellt und getestet. Um einmal an einem Beispiel deutlich zu machen, welche Schritte in der IDE im Hintergrund ausgeführt werden, erstellen wir unser erstes Programm ohne IDE.

- (1) Erstellen des Java-Quellcode mit einem beliebigen Editor (z.B. notepad++).
Achtung: der Dateiname muss gleich sein wie der Klassenname und bekommt die Endung `.java` (s.o.)
- (2) Starten einer Windows „Eingabeaufforderung“ (<Windows Taste> cmd <Enter>)
Wechseln in das Verzeichnis wo der Java-Quellcode liegt:
Zuerst ins Netzlaufwerk H: wechseln mit: `H:`
Dann mit dem Befehl `cd` ins Quelltextverzeichnis wechseln. Bsp.: `cd \PROG\Hallo`

- (3) Übersetzen des Quellcodes in den Java Byte-Code mit Hilfe des Befehl:
`javac <Dateiname>` (z.B. `javac HalloWelt.java`)
Wenn die Übersetzung ohne Fehler ausgeführt werden konnte, dann wurde in dem Verzeichnis nun die Datei `HalloWelt.class` erzeugt. Diese enthält den Byte-Code.
Anzeigen des Verzeichnisinhalts mit dem Befehl: `dir`

Hinweis: Möglicherweise wird der Befehl `javac` in der aktuellen Windows Installation nicht gefunden. Dann muss der komplette Pfad zum Java Compiler (`javac`) angegeben werden: z.B.

`C:\Program Files\Java\jdk1.8.0_144\bin\javac HalloWelt.java`

(oder die Umgebungsvariable `Path` entsprechend konfiguriert werden)

- (4) Der Interpreter (JVM Java Virtual Machine) interpretiert den Byte-Code und führt ihn aus.
Befehl: `java <Klassenname>` (z.B. `java HalloWelt`)