

Einführung in die Java Grundlagen

Als Einstieg in die Java Grundlagen siehst du hier einen einfachen Programmcode. Dieser fängt mit `class` an. Doch was ist das eigentlich? Eine `class` (oder auch Klasse genannt) ist ein Abschnitt eines Programms. Je nach Größe des Programms kann es mehrere Klassen geben. Für ein einfaches Hello-World-Programm reicht dir doch eine einzige. Es gibt Klassen, die aus nur einer Zeile bestehen, und solche, die mehrere Zeilen enthalten. Damit der Compiler weiß, wo eine Klasse anfängt und wo sie aufhört, werden geschweifte Klammern am Anfang und am Ende gesetzt. Es ist dir überlassen, wie die Klasse heißt. Du musst aber folgende Dinge beachten: Erstens, der Name wird entweder nur aus Buchstaben oder aus Buchstaben und Ziffern gebildet. Zweitens, das erste Zeichen muss ein Buchstabe sein, und drittens, der Name darf keine Leer- oder Sonderzeichen enthalten. Wie wir wissen, gibt das Beispielprogramm die Worte „Hello World“ auf dem Bildschirm aus. Warum also der ganze Aufwand für so einen kleinen Output? Der eigentliche Output, also das, was auf dem Bildschirm erscheint, wird mit `System.out.println` in der fünften Zeile beschrieben. Die anderen Zeilen haben mit dem Output wenig zu tun, sind aber dennoch wichtig. Die dritte Zeile ist der Startpunkt des Programms. Das Wort „Main“ ist der Name der Methode und bedeutet für Java, dass das der Startpunkt des Programms ist. Die Main-Methode sieht für Einführungsprogramme immer so aus, deswegen solltest du dir den Code dazu einfach merken. Die geschweiften Klammern der Main-Methode geben den genauen Programmstart und das Programmende an. Dazwischen befinden sich Anweisungen, wie sich das Programm verhalten soll. In diesem Fall besteht das aus einer einzigen Anweisung – `System.out.println Hello World`. Dabei gibt der Text innerhalb der Anführungszeichen an, was auf dem Bildschirm ausgegeben wird. Wie dir bestimmt schon aufgefallen ist, heißt die Klasse des Programms „Hello World“. Das bedeutet, dass der Name der Datei „Hello World.java“ heißen muss, damit die Namen übereinstimmen und der Compiler keinen Error meldet. Zur Info: Der Compiler ist dafür da, den Quellcode, den du geschrieben hast, in einem Bytecode umzuwandeln, damit der Java-Prozessor diesen ausführen kann und somit das Programm startet. So wird dann die Datei „Hello World.class“ erzeugt. Wenn dein Quellcode keinen Sinn ergibt, dann gibt der Compiler einen Error als Antwort. Doch selbst, wenn man ein Programm fehlerfrei schreibt und keine Syntaxfehler auftauchen, kann es trotzdem passieren, dass es seinen Zweck nicht erfüllt und somit nicht korrekt ist, z.B. wenn der Output des Programms „Hello“ statt „Hello World“ ist. Dabei handelt es sich dann um einen sogenannten Bug. Das Programm lässt sich zwar problemlos kompilieren, aber das Ergebnis schweift vom eigentlichen Ziel ab. Deshalb ist es wichtig, vor allem bei größeren Programmen zu planen, was dieses überhaupt machen sollen und solche Bugs zu vermeiden. Wenn du möchtest, kannst du auch mehrere Zeilen Text auf dem Bildschirm erscheinen lassen. Dazu musst du die `System.out.println` Anweisung einfach mehrmals benutzen. Das Ganze könnte dann zum Beispiel so aussehen.

Außerdem steht jedes „Hello World“ in einer neuen Zeile. Das liegt daran, dass wir die Anweisung `println` verwenden, wodurch jeder String eine eigene Zeile zugewiesen bekommt. String? Ja, du hast richtig gehört. Bei den Sätzen, die sich in den Anführungszeichen befinden, handelt es sich um strings. Vielleicht ist dir schon aufgefallen, dass in der `main`-Methode ein String eingebunden wurde. Dadurch können wir Zeichen aneinanderreihen, die dann als String erkannt werden. So können wir diese dann auf dem Bildschirm ausgeben. Damit dein Quellcode übersichtlich und leicht zu verstehen ist, solltest du dir angewöhnen, ein sauberes und konsistentes Layout zu verwenden. Ansonsten sieht das ganze irgendwann so aus.

Auch kann es nützlich sein, Teile des Quellcodes zu kommentieren, damit man den Überblick nicht verliert, soweit das Programm etwas größer wird. Dabei gibt es zwei Arten von Kommentaren – einzeilige und mehrzeilige Kommentare. Das Ganze sieht dann so aus:

Dir ist bestimmt schon aufgefallen, dass die Kommentare grün markiert werden. Das ist in den meisten Programmiersprachen und Texteditoren der Fall. Ein einzeiliger Kommentar beginnt mit einem doubleSlash. Alles was hinter dem doubleSlash steht, wird von dem Compiler ignoriert und dient nur zum Verständnis für dich als Programmierer oder für andere, die dein Programm verstehen wollen. Für mehrzeiliges Kommentieren muss man das Gebiet angeben, in dem der Text als Kommentar zugeordnet wird. Dabei handelt es sich um ein ähnliches Schema, wie bei den geschweiften Klammern nur, dass man hier für den Kommentar Start `/*` benötigt und für das Ende ein `*/`. Und zum Schluss noch zwei wichtige Hinweise. Achte immer darauf, dass du die geschlossene geschweifte Klammer nicht vergisst, sobald du eine Klasse oder Methode schreibst und zweitens – Anweisungen, wie zum Beispiel `System.out.println` Befehl, haben immer ein Semikolon am Ende der Syntax. So, die Grundlagen von Java sind jetzt hoffentlich kein Problem mehr für dich und du bist in der Welt des Programmierens angekommen