

Java-Main

Schauen wir uns zunächst einmal den Aufbau der Main-Methode in Java an. Sie sieht wie folgt aus: `public static void main(String[] args){`. Der Aufbau ist immer gleich: Am Anfang schreibst du einen `public`. Dabei handelt es sich um einen sogenannten Modifier. Er liegt die Sichtbarkeit von Programmelementen fest. `Public` erlaubt, dass du von außerhalb der Klasse auf die Methode zugreifen kannst. Als nächstes kommt das Wort `static`. Hiermit kennzeichnest du die Methode als statisch und machst sie unabhängig von der Klasse, in der sie steht. Später mehr dazu. Danach schreibst du ein `void`. Es kennzeichnet, dass die gesamte Methode keinen Rückgabewert besitzt. Jetzt kommt der Name der Methode, dieser ist hier `main()`. In den Klammern des Namens steht der Parameter `String args` – er ist eine Art der Kommunikation zwischen der Konsole und dem Java-Programm. Wenn du nun dein geschriebenes Programm über die Kommandozeile startest, kannst du hier noch etwas dazu schreiben, was dann als Array gespeichert und direkt in das Java-Programm überreicht wird. Das erklären wir dir später noch genauer. Wenn du noch nicht weißt, was `Array` und `Java-Klassen` sind, dann schau dir unsere Videos dazu an. Mit ihnen kannst du super dein Wissen weiter vertiefen. Jetzt hast du auch schon den groben Aufbau und die Funktionen der `Main-Methode` kennengelernt. Nun schauen wir uns den Ablauf mal genauer am Beispiel eines „Hello World“ Programms an. Den Code dafür hattest du ja in unserem Video zu den `Java Grundlagen` schon einmal kurz gesehen. Zunächst startest du den Code von deiner Kommandozeile aus mit dem Befehl „Java Hello World“. Wenn du hier beispielsweise schreiben würdest, „Java Hello World blablab“, so würdest du dem Programm die `string` Werte „bla“ und „blub“ mitgeben. Diese sind dann in den Arrays `args` gespeichert und du kannst auch Sie über `args` innerhalb der `Main Methode` zugreifen. Hier wäre jetzt `Args` an der Stelle 0 gleich „bla“ und `Args` an der Stelle 1 gleich „blub“. Du musst jedoch keine Werte mitgeben, dann bleibt das Array `args` einfach leer. Mit dem Befehl „Java Hello World“ startest du dein Programm von der Konsole aus. Weißt du noch, wie das geht? Richtig, du hast die Methode als `static` markiert, das bedeutet, sie ist unabhängig von der Klasse, und dein Computer kann direkt auf die `Main-Methode` zugreifen, ohne erst ein Objekt der Klasse erzeugen zu müssen, aber machen wir langsam. Was das genau bedeutet, wirst du in den nächsten Videos lernen. Methoden können auch verschiedene Werte zurückgegeben. Das ist bei der `Main-Methode` jedoch nicht der Fall. Sie kann lediglich Werte ausgeben oder neue Objekte im Rahmen der Objektorientierung erzeugen. Das besagt gerade das Schlüsselwort `Void`. Nun kommt die eindeutige Namensgebung der Methode mit `main ()`. In die Klammern kommt der Parameter `String args`, den kennst du ja schon. Nachdem das Programm sich jetzt in der `Main-Methode` befindet, werden die Befehle, die du dort eingegeben hast, von oben nach unten durchgearbeitet. Das Programm beendet sich automatisch, wenn es das Ende der `Main-Methode` erreicht hat. Super! Jetzt hast du den Ablauf einmal durchgearbeitet. Anwenden kannst du das Ganze vor allem in der Objektorientierung in Java. Generell ist es so, dass du mit einer `Main-Methode` ein neues Objekt einer Klasse erzeugen kannst. Schauen wir uns das zusammen an einem Beispiel an: Wir programmieren dafür zwei Klassen: Einmal eine Klasse mit einer `Main-Methode` und einmal eine Klasse ohne eine `Main-Methode`. Wir nennen die zwei Klassen deshalb mit `Main` und `ohne Main`. Nun erzeugt das Programm in der `Main-Methode` ein neues Objekt der Klasse `ohne Main` mit dem Operator `new` und speichert es in einer Variablen `x`. Die Klasse `ohne Main`, von der das Objekt erzeugt wird, ist hier einfach leer. Mehr zu dem Thema, was Objekte genau sind und wie du mit ihnen arbeiten kannst, erfährst du in unserem Video zu den `Java Objekten`.