

If else Java

Aber was ist eine if-Anweisung überhaupt? Sie hilft dabei, etwas Leben in dein Programm zu bringen, indem sie z.B. Input verarbeitet und damit den Verlauf des Programms verändert. So weit, so gut. Und wie sieht eine Entscheidung in der Programmierung aus? Wie du sicher weißt, besteht eine Entscheidung aus mindestens zwei Optionen. Ob man sich für eine Option entscheidet, wird über die booleschen Werte true und false entschieden. Das bedeutet, dass es eine oder mehrere if-Anweisungen gibt, die verschiedene Optionen darstellen können. Und wie entscheidet sich das Programm dann für eine Option? Jede Option besitzt eine Bedingung, die erfüllt werden muss. Dann gibt die Anweisung den Wert true zurück und verändert damit den Programmverlauf. Das ist dir immer noch nicht ganz klar? Nehmen wir mal ein Beispiel: Du schreibst eine App, die die Nutzer mitteilen soll, ob die Sonne scheint. Dafür verwendest du eine if-Anweisung, die sich nur mit der Bedingung ausführt, wenn die Sonne scheint. Dann ist der Wert also true und der Nutzer bekommt eine Nachricht. Wenn sie nicht scheint, also der Wert false ist, bekommt er keine Nachricht. Der Code, um die Nachricht für Sonnenschein zu senden, ist also in dem Programmcode vorhanden. Er wird aber nur unter bestimmten Umständen ausgeführt. Schauen wir uns das Ganze etwas genauer an. Am Anfang steht der Befehl if, um die if-Anweisung auszuführen. Dem if muss immer eine Bedingung folgen, die in runden Klammern geschrieben wird. In diesem Fall wird die boolesche Variable „Sunshine“ überprüft. Falls der Wert true ist, wird die nächste Anwendung, die der if-Anweisung folgt, ausgeführt. Ob der Wert true ist, wird mittels dem booleschen Ausdruck Doppel gleich geprüft. In diesem Fall handelt es sich um einen Output, der mitteilt, dass die Sonne scheint. Und was bedeutet jetzt dieses else? Was nach einer neuen Anweisung aussieht, gehört eigentlich zur gleichen if-Anweisung dazu. Man kann dieser ganz am Schluss ein else, also ein andernfalls, anhängen. Das bedeutet, falls alle anderen Optionen der if-Anweisung false ausgegeben haben, wird die Anweisung, die dem else folgt, immer ausgeführt. Man muss sich dabei aber nicht immer nur auf eine ausgeführte Anweisung beschränken. Mit Hilfe von geschweiften Klammern kann man dem Compiler mitteilen, welcher Bereich zu jeweiligen if-Anweisung gehört. Somit kann man dann mehr Code unter der if Bedingung ausführen lassen. Wenn du willst, kannst du auch if-Anweisungen in eine if-Anweisung einbauen. Dadurch bekommt dein Programm mehr Dimension und wird komplexer. In diesem Beispiel wird der Output „Zeit an den Strand zu gehen“ nur ausgeführt, wenn die Sonne scheint und die Temperatur größer oder gleich 25 Grad ist. Bei Temp handelt es sich dabei um eine Variable des Typs double. Und hier ist auch schon der nächste und letzte Befehl der if-Anweisung. Vergiss nicht, dass alle Befehle, bis auf das if selbst, optional sind und nicht immer eingebaut werden müssen. Der letzte Befehl ist das else if. Wie der Name ansonsten „falls“ schon sagt, wird er nur ausgeführt, wenn seine eigene Bedingung erfüllt wird und wenn alle Bedingungen davor false waren. Das bedeutet in diesem Beispiel, falls die Sonne scheint, die if Bedingung also erfüllt ist, kann die else if Bedingung gar nicht mehr ausgeführt werden. Das gilt selbst, wenn es regnet. Denn sie gehören der gleichen if-Anweisung an. In einer if-Anweisung kann immer nur eine Bedingung ausgeführt werden selbst, wenn mehrere von diesen erfüllt sind. Wichtig ist, dass immer ein if existieren muss, um eine if-Anweisung zu starten. Else if und else sind optional.

Eine if-Anweisung darf nur ein else-Befehl enthalten, welcher immer am Schluss der if-Anweisung stehen muss. Du kannst aber so viele else if wie du willst in eine if-Anweisung hinter das If packen, solange der Syntax stimmt. Doch was, wenn du beide Bedingungen ausfüllen möchtest, wenn sie erfüllt sind? Also, wenn die Sonne scheint und wenn es regnet? Wie du jetzt weißt, kann ja nur die erste Bedingung ausgeführt werden, weil damit alle anderen automatisch übersprungen werden. Der Schlüssel hierzu ist, mehrere if-Anweisungen zu verwenden. Dadurch können beide Bedingungen ausgeführt werden, solange sie erfüllt sind. Das Ganze sieht dann so aus. Du musst dabei beachten, dass es sich hier um zwei verschiedene if-Anweisungen handelt und nicht um eine einzige. Deshalb besitzen beide ihr eigenes optionales else if und else und können sich diese nicht teilen. Du fragst dich, warum wir bisher immer nur die system.out.println-Anweisungen als Ausführung der Bedingung verwendet haben? Das ist nur eins von vielen Beispielen. Du kannst natürlich alle möglichen Anweisungen in die if-Anweisung packen, z.B. kannst du mittels Bedingung überprüfen, ob die Sonne scheint und dann als Ausführung die Zuweisungsanweisung verwenden, die die jetzige Uhrzeit in einer Variablen speichert.