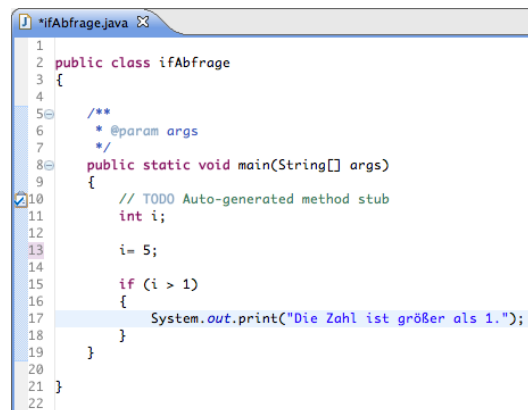


Fallunterscheidungen

Beim Programmieren kommt es häufig vor, dass manche Anweisungen nur ausgeführt werden sollen, wenn eine bestimmte Bedingung erfüllt ist. Dies erreicht man mit einer if-Abfrage.

Ein einfaches Beispielprogramm dazu sieht wie folgt aus:



```
1
2 public class ifAbfrage
3 {
4
5     /**
6      * @param args
7      */
8     public static void main(String[] args)
9     {
10        // TODO Auto-generated method stub
11        int i;
12
13        i = 5;
14
15        if (i > 1)
16        {
17            System.out.print("Die Zahl ist größer als 1.");
18        }
19    }
20 }
21 }
22 }
```

- Die if-Abfrage befindet sich in den Zeilen 15 bis 18.
- In Zeile 15 steht nach if in runden Klammern die Bedingung für die Ausführung des folgenden Programmcodes. Die Bedingung ist eine Aussage; ist die Aussage wahr, wird der nachfolgende Programmcodes ausgeführt, ist die Aussage falsch, wird er nicht ausgeführt.
- Der zur if-Abfrage gehörende Programmcodes wird durch die geschweiften Klammern in den Zeilen 16 und 18 zu einem Block zusammengefasst, der in diesem Beispiel nur aus einer Programmzeile besteht.
- Der Code in diesem Block wird nur dann ausgeführt, wenn die Aussage in der Bedingung wahr ist.

Notiere hier deine Beobachtungen für den Fall, dass der Variable *i* in einem Programmlauf der Wert 5 zugewiesen wird, im nächsten Programmlauf jedoch der Wert -5 zugewiesen wird.

Im obigen Beispiel wird der Programmcode nur ausgeführt, wenn die Bedingung erfüllt ist – falls nicht, wird nichts ausgeführt. Häufig hat man aber die Situation, dass im Falle einer nichterfüllten Bedingung andere Anweisungen ausgeführt werden sollen. Dies erreicht man, indem man die if-Abfrage um einen else-Zweig erweitert, wie das folgende Beispielprogramm zeigt:

```
if (i > 1)
{
    System.out.print("Die Zahl ist größer als 1.");
}
else
{
    System.out.print("Die Zahl ist kleiner oder gleich 1.");
}
```

Der hier gezeigte else-Zweig ist optional – es gibt ihn also nur, wenn man ihn auch tatsächlich braucht.

Der Programmcode im ersten Block wird also nur ausgeführt, wenn die Bedingung $i > 1$ wahr ist, andernfalls wird der Programmcode im zweiten Block ausgeführt.

Gängige Abfragen sind:

- größer als >
- kleiner als <
- größer oder gleich >=
- kleiner oder gleich <=
- gleich ==
- ungleich !=

Es ist auch möglich, dass links und/oder rechts des Vergleichsoperators ein Ausdruck steht, der zunächst berechnet werden muss, bevor der Wahrheitswert der Aussage ermittelt werden kann.

Der Rechenausdruck $i \% 2$ liefert den Divisionsrest der Division des Wertes der Variablen i durch 2. Ist dieser Rest 0, dann enthält die Variable i offenbar eine gerade Zahl.

Modifiziere die if-Abfrage so, dass das Programm ausgibt, ob die Variable i eine gerade oder ungerade Zahl enthält. Halte den wesentlichen Programmteil hier fest: