Java Cheat Sheet für Einsteiger

Einleitung

Ein *Java*-Programm wird immer von oben nach unten abgearbeitet.

Jede Anweisung in *Java* muss mit einem Semikolon (;) abgeschlossen werden. Jede öffnende Klammer benötigt eine schließende Klammer.

Variablen

Variablen haben unterschiedliche Datentypen (z.B. *boolean*, *int*, *double*, *String*). Eine Variable muss deklariert werden. Anschließend kann für die Nutzung der Variable ein Wert zugewiesen werden (Definition).

Datentypen

Ein Datentyp legt fest, welche Daten in der Variable gespeichert werden können:

- ganze Zahlen (z.B. *short*, *int*)
- Fließkommazahlen (z.B. *float*, *double*)
- Wahrheitswerte (z.B. *boolean*)
- Zeichenketten (z.B. *String*)

Deklarierung

Bei der Deklarierung wird zuerst der Datentyp und anschließend der Name der Variable angegeben. Mit dem Namen kann die Variable später im Programm verwendet werden. Der Name der Variable ist frei wählbar.

```
boolean running;
int zahl;
double pi;
String text;
```

Definition

Nachdem eine Variable deklariert wurde, kann ihr ein Wert zugewiesen werden. Dies geschieht mit dem Zuweisungsoperator:

```
running = true;
zahl = 42;
pi = 3.1415;
text = "Hallo, Welt!";
```

Deklaration und Definition

Die Deklaration und Definition kann in einem Schritt erledigt werden:

```
boolean running = true;
int zahl = 42;
double pi = 3.1415;
String text = "Hallo";
```

Operatoren

In *Java* gibt es eine Reihe von Operatoren. Operatoren verknüpfen die Operanden, wie z.B. zwei Variablen:

Operator	*Beschreibung*
=	Zuweisungsoperator
<	Kleiner-Operator
==	Gleichheitsoperator
!=	Ungleichoperator
>	Größer-Operator
+	Additionsoperator
-	Subtraktionsoperator
*	Multiplikationsoperator
/	Divisionsoperator

Im Beispiel würde die Anwendung von Operatoren wie folgt aussehen:

```
int sum = 4 + 7;
int result = 4 * 2;
```

Verzweigungen

In Programmen müssen Entscheidungen getroffen werden und anhand dieser muss der Programmfluss verzweigt werden. Dazu kann in *Java* unter anderem die Anweisung *if* genutzt werden:

```
if(count == 42) {
    // count ist 42
}
```

Mit *if* und *else* bzw. *else if* können auch mehrere Fälle abdeckt werden:

```
if(text.equals("Hallo")) {
```

```
  // Variable ist "Hallo"
} else
if(text.equals("Welt")) {
  // Variable ist "Welt"
}
else {
  // Variable ist etwas
anderes
}
```

Schleifen

In *Java* gibt es eine Reihe von Schleifen. Mit Schleifen können Programmteile mehrfach durchlaufen werden.

for-Schleife

Die *for*-Schleife ist eine Zählschleife. Im Schleifenkopf wird eine Variable deklariert und definiert. Diese Variable trägt meist den Namen *i* und ist die Zählvariable.

Nach der Definition der Zählvariable, wird die Bedingung definiert, welche festlegt wann die Schleife abgebrochen wird.

Danach wird der Befehl definiert, welcher nach einem Durchlauf der Schleife ausgeführt wird.

```
for(int i = 0; i < 100; i++)
{
  // Führe Anweisungen aus
}
```

while-Schleife

In der *while*-Schleife wird eine Bedingung definiert. Solange diese Bedingung erfüllt ist, wird die Schleife ausgeführt:

```
int z = 100;

while(z > 0) {
  z--; // Reduziere z
}
```

Kommentare

Mit Kommentaren kann der Quelltext mit Hinweisen und Informationen versehen werden. Diese werden vom Compiler ignoriert. Die am häufigsten verwendeten Kommentare sind Zeilenkommentare:

```
// Ich bin ein Kommentar
```

Ausgabe

Für die Ausgabe von Werten auf der Konsole kann die Methode *System.out.println()* genutzt werden:

```
System.out.println("Hallo,
Welt!");
```

Eingabe

Neben der Ausgabe von Informationen lesen Programme oft auch Informationen ein. Unter *Java* kann hierfür die Klasse *Scanner* genutzt werden:

```
Scanner scanner = new
Scanner(System.in);
String line =
scanner.nextLine();
```

In der Variable *line* ist enthalten was der Nutzer eingegeben hat.

Zufall

Für bestimmte Programme wird Zufall benötigt. Hierfür eignet sich die Klasse *Random*, welche Pseudozufallszahlen generiert:

```
Random random = new
Random();
int randomNumber =
random.nextInt();
```

Der Methode *nextInt()* kann ein Wert übergeben werden, welcher den Zahlenbereich der Zufallszahlen einschränkt:

```
int randomNumber =
random.nextInt(1000);
```

Damit werden nur noch Zahlen zwischen 0 und 999 generiert.