

# Kapitel 4 – Variablen

**Übersicht** Du weißt aus den vorherigen Kapiteln schon ungefähr, was Variablen sind. Jetzt wollen wir uns Variablen noch einmal genauer ansehen. Um die Eigenschaften von Variablen besser zu veranschaulichen, wollen wir Ihren Inhalt am Bildschirm ausgeben. Dafür müssen wir uns noch mal mit der Ausgabe am Bildschirm befassen.

Wir wollen in diesem Kapitel **interaktive Programme** schreiben. Das bedeutet, dass die kompilierten Programme während der Ausführung über den Bildschirm und die Tastatur mit dem Benutzer kommunizieren. Um ein interaktives Programm schreiben zu können, brauchen wir also noch die Möglichkeit Eingaben über die Tastatur einzulesen und zu verarbeiten.



## Lernziel

Wir wollen Variablen noch mal genauer unter die Lupe nehmen. Du lernst, was Variablen sind, wie man sie deklariert und verwendet. Obwohl das Kapitel Variablen heißt, wirst du auch lernen, welche Möglichkeiten man hat, Ausgaben am Bildschirm zu machen und Eingaben vom Benutzer über die Tastatur im Programm zu verwenden.



## Vorbereitung

Bevor wir nun mit den Variablen anfangen können, wollen wir einige Vorbereitungen treffen:

### A) Ausgabe am Bildschirm

Wie du weißt, wollen wir interaktive Programme schreiben. Wie Texte während der Ausführung des Programms ausgegeben werden, wissen wir, nämlich mit

```
System.out.print("irgendein String")
```

Durch den Befehl

```
System.out.println("irgendein String")
```

(erkennst du den Unterschied???) wird der in den Klammern angegebene String ausgegeben und dann in die nächste Zeile gesprungen.

### Beispiel:

Programmtext	Ausgabe am Bildschirm
<pre>System.out.println("**Einkaufsliste**"); System.out.println(""); System.out.println("3 Liter Milch"); System.out.println("500gr Weintrauben"); System.out.println("5 x Joghurt"); System.out.println("2 x Brot");</pre>	<pre>**Einkaufsliste**  3 Liter Milch 500gr Weintrauben 5 x Joghurt 2 x Brot</pre>

Schreibt man in einen String

`\n`

dann bewirkt das bei der Ausgabe des Strings einen Zeilenumbruch an der Stelle wo `\n` steht.

Ein

`\t`

in einem String bewirkt einen Zeilenvorschub, wie, wenn man in einem Textverarbeitungsprogramm die Tabulatortaste verwendet.

Programmtext	Ausgabe
<code>System.out.print("1 \n 2 \n 3 \n 4");</code>	1 2 3 4

Oft lassen sich Zeichen wie ö, ü, ä und ß nicht mit `System.out.print...` ausgeben. Deshalb schreibt man stattdessen in der Regel `oe`, `ue`, `ae` und `ss`.

## B) Eingabe über die Tastatur

Wie bekommen wir ein Programm nun dazu während der Ausführung Eingaben über die Tastatur entgegenzunehmen, um sie dann weiterzuverarbeiten? Das ist nicht so einfach. Woher soll ein Programm wissen, ob eine Zahl oder ein Zeichen gemeint ist, wenn man die 3 auf der Tastatur drückt? Wir haben bis jetzt noch nicht das Wissen, um ein Programm zu schreiben, was das kann. Also holen wir uns Hilfe:

**Kopiere dir die Datei `Kon.java` aus dem Jahrgangsstufenordner bzw. Teams in den Ordner, in dem du deine Javaquelltexte liegen hast und kompiliere sie!**

Wenn du jetzt ein Programm schreibst, und es im **gleichen Ordner** wie die `Kon.java` ablegst, hast du die Möglichkeit, Eingaben von der Tastatur zu lesen.

Mit den folgenden Methodenaufrufen kannst du einen Wert vom jeweiligen Datentyp über die Tastatur einlesen lassen:

`Kon.readInt();`

Einlesen von ganzen Zahlen (int)

`Kon.readDouble();`

Einlesen von Dezimalzahlen (double)

`Kon.readChar();`

Einlesen von Zeichen (char)

`Kon.readString();`

Einlesen von Texten (String)

**Aufgabe:** Dann wollen wir das mal ausprobieren.

1. **Schreibe ein Programm, das nach dem Namen des Benutzers fragt!** Die Ausgabe am Bildschirm könnte zum Beispiel so aussehen:

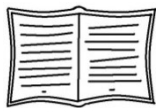
Wie heisst du ?

**Tipp:** Wenn du nicht genau weißt, wie du das hinbekommen sollst, dann orientiere dich an dem Programm `Mein_erstes_Programm.java`.

2. **Wenn das geklappt hat, dann erweitere dein Programm so, dass bei der Ausführung ein Wert über die Tastatur eingelesen wird!**

**Arbeite erst weiter, wenn dein Programm funktioniert.**

Dein Programm liest jetzt bei der Ausführung einen String über die Tastatur ein. Allerdings wird dieser String noch nicht verarbeitet. Das Programm ist in dieser Form also relativ sinnlos! Der String muss weiterverarbeitet werden. Dazu muss er zunächst einmal gespeichert werden. Wir brauchen also Variablen.



## Theorie

Du hast schon nebenbei einiges über Variablen gelernt. Wir wollen Sie uns jetzt genau ansehen. **Variablen sind benannte Behälter für Daten.** Sie können ihren Wert ändern und haben einen Datentyp. Das bedeutet, dass man in eine Variable nur Werte speichern kann, die den gleichen Datentyp wie die Variable haben. Man sagt, Wert und Variable müssen **zuweisungskompatibel** sein.

Beispiel:

3	'a'	"Text"	1,4	123
x	y	z	merke	v
int	char	String	double	int

In der Darstellung gibt es die Variablen `x`, `y`, `z`, `merke` und `v`. `x` ist vom Datentyp `int` und enthält den Integerwert 3, `y` ist vom Datentyp `char` und enthält den Wert `'a'` usw..

Jede Variable muss vor ihrer Verwendung **deklariert** werden. Das hast du im Laufe der letzten Kapitel sicherlich mitbekommen. Es bedeutet, dass Namen und Datentyp der Variablen bekannt gemacht werden. Der Compiler reserviert Speicherplatz für den Wert, der in der Variable abgelegt werden soll.

Die Deklaration für die obigen Variablen sieht so aus:

```
int x;
char y;
String z;
double merke;
int v;
```

Übrigens: Wenn in einer Deklaration mehrere Variablen von demselben Datentyp angelegt werden sollen, dann kann man diese auch alle durch Komma getrennt hinter die Typangabe schreiben.

Die Deklaration für die obigen Variablen kann also auch so aussehen:

```
int x, v;  
char y;  
String z;  
double merke;
```

Mit dem = - Zeichen bewirkt man eine **Wertzuweisung**. Damit kann man einen Wert in einer Variablen speichern.

Die Wertzuweisungen für die obigen Variablen sehen wie folgt aus:

```
x = 3;  
y = 'a';  
z = "Text";  
merke = 1,4;  
v = 123;
```

Da Variable und Wert zuweisungskompatibel sein müssen, müssen die linke und rechte Seite vom Gleichheitszeichen, in der Regel denselben Typ haben. Zum Beispiel kann man in eine Variable, die für int-Werte vorgesehen ist, keine Strings speichern.

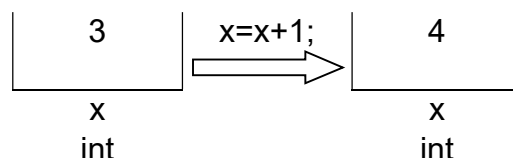
```
x = "Gymnasium Siegburg Alleestraße";
```

**geht nicht**

Nach der Wertzuweisung kann man die Variable anstelle des Werts benutzen. Man kann zum Beispiel folgendes machen:

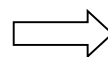
```
x = x+1;
```

Zunächst wird der Ausdruck  $x + 1$  ausgewertet. Da 3 in x gespeichert ist, wird  $x+1$  zu 4 ausgewertet. Dann wird der ermittelte Wert der Variablen x zugewiesen. Also enthält x jetzt den Wert 4.



Man kann den Wert der Variablen am Bildschirm ausgeben lassen.

```
System.out.print(x);
```



4

Programmzeile

Ausgabe am Bildschirm

Außerdem kann man mit `Kon.readInt()`, `Kon.readDouble()`, `Kon.readChar()`, und `Kon.readString()` Werte, die über die Tastatur eingegeben werden, in einer Variablen speichern.

