

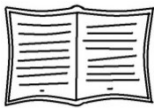
Kapitel 3 – Datentypen und Operationen

Übersicht Wir haben in Kapitel 2 schon gelernt, dass der Rechner Daten von unterschiedlichen Typen (z.B. `double`) verarbeiten kann. Wie funktioniert nun die Verarbeitung? Was kann man also mit diesen Daten machen? Man kann mit ihnen operieren. Beispiele dafür sind die Grundrechenarten. Wenn man nun eine Operation auf Daten bestimmter Datentypen ausführt, von welchem Datentyp ist das Ergebnis dann? Das musst du beim Programmieren wissen, damit du weißt, mit welchem Datentyp die Variable, in die das Ergebnis gespeichert werden soll, deklariert sein muss.



Lernziel

In diesem Kapitel wirst du lernen, welche einfachen Datentypen es gibt, welche Operationen man auf ihnen ausführen kann und von welchem Datentyp die Ergebnisse sind.



Theorie

Ein Computer muss viele Daten verarbeiten können. Du weißt, dass es verschiedene Arten von Daten, sogenannte **Datentypen** oder einfach nur **Typen**, gibt. Ein Computer kennt zum Beispiel unter anderem Zahlen, Zeichen oder auch Texte.

In den letzten beiden Kapiteln sind wir schon auf Daten und Datentypen gestoßen. Im Programm `Mein_erstes_Programm.java` hatten wir mit Texten zutun.

Die Ausgabe mit

```
System.out.print( )
```

ist u.a. für Daten vom Datentyp Text (bei Java „String“ genannt) vorgesehen. Um dem Computer zu signalisieren, dass es sich bei einem Datum um einen Text handelt, muss er vom Programmierer als Text gekennzeichnet werden. Dazu setzt man Texte in Anführungszeichen (oben).

```
"Dieses ist ein Text!"
```

Die Anführungszeichen signalisieren dem Rechner also, dass es sich um einen Text handelt. In der Fachsprache bezeichnet man einen so gekennzeichneten Text als **String**.

Im letzten Kapitel wurde im Programm `Kreisumfang.java` mit Dezimalzahlen gerechnet, um zu einem Radius den Kreisumfang zu bestimmen. In Java sind die Werte vom Datentyp `double`.

Welche Datentypen gibt es sonst noch bei Java?

Zunächst wollen wir nur die sogenannten „Einfachen Datentypen“ betrachten.

Einfache Datentypen

Datentyp	in Java	Beispiele
ganze Zahlen (Integer)	int	1 -2 0 2 42
Gleitkommazahlen, Dezimalzahlen	double	3.14159 -2.71828
Wahrheitswerte	boolean	true false
Zeichen (Character)	char	'a' '3' '+' '\$' '.'
Zeichenkette, Text	String	"Hallo Welt" "3 ist eine ganze Zahl"

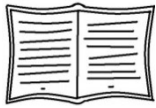
- Ganze Zahlen und Dezimalzahlen, sind dir aus der Mathematik bekannt. Auch bei Java können sie negativ oder positiv sein.
- Es gibt nur zwei Wahrheitswerte: wahr (`true`) oder falsch (`false`).
- Zeichen werden in Apostrophe (' ') und
- Texte in Anführungszeichen (" ") gesetzt.

Bei Zeichen und Texten muss man beachten, dass zwischen Groß- und Kleinbuchstaben unterschieden wird. Für einen Rechner sind ein Kleinbuchstabe und sein zugehöriger Großbuchstabe völlig verschiedene Zeichen.

Mit Hilfe von Java kann man Daten verarbeiten. Zum Beispiel kann man eine einfache Addition zweier ganzer Zahlen durchführen. Die Berechnung wird durch einen sogenannten **Operator** bewirkt. In diesem Fall handelt es sich bei diesem Operator um ein `+`.

Diese Kombination von Daten eines bestimmten Datentyps mit Hilfe von Operatoren, nennt man **Ausdrücke**. Die Daten bezeichnet man dabei als **Operanden**. Bei der Auswertung eines Ausdrucks ergibt sich ein Datum, das wiederum einen Datentyp hat. Dieser ist der **Ergebnisdatentyp**. Addiert man beispielsweise zwei Werte vom Datentyp `int`, so erhält man wieder einen Wert vom Datentyp `int`.

Beispiel: 3 und 2 sind Daten vom Datentyp `int`.
+ ist ein Operator für Daten vom Datentyp `int`.
3 + 2 ist ein Ausdruck mit den Operanden 3 und 2 und dem Operator +.
Der Ergebnisdatentyp des Ausdrucks 3 + 2 ist vom Datentyp `int`.



Theorie

Was gibt es nun für Operationen? Die meisten Operationen sind Dir bekannt.

1. Grundrechenarten

Das sind zunächst einmal die **Grundrechenarten** (+, -, *, /). Diese kann man sowohl auf int- als auch auf double-Werte anwenden, z.B.

$$3 + 2 \quad \text{oder} \quad 3.2 + 2.1$$

Die Bedeutung dieser Operatoren ist bis auf die Division bei `int`, die sogenannte ganzzahlige Division, klar.

Bei der **ganzzahligen Division**, werden einfach die Stellen hinterm Komma vernachlässigt.

Beispiel:

3/2	wird ausgewertet zu 1
4/7	wird ausgewertet zu 0
2/2	wird ausgewertet zu 1

Die ganzen Zahlen sind eine Teilmenge der rationalen Zahlen. Deshalb kann man auch Operationen durchführen, wo einer der beiden Operanden vom Typ `int` und der andere vom Typ `double` ist. Das Ergebnis ist dann natürlich vom Typ `double`.

Beispiel: $1.2 + 2$ wird ausgewertet zu 3.2

Intern formt der Computer den Integerwert zu einem Wert vom Typ `double` um und führt dann die Operation durch. Das Umformen zum passenden Typ nennt man **Typkonversion**. Sie kommt auch bei anderen Datentypen vor.

2. Verkettung

Beim Datentyp `String` hat der Operator + eine ganz andere Bedeutung. Es verkettet zwei Strings.

"Hal" + "lo" wird ausgewertet zu "Hallo"

3. Vergleichsoperationen

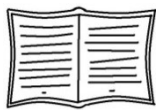
Um zwei Werte miteinander zu vergleichen gibt es **Vergleichsoperationen**, z.B.

$$3 < 2 \quad \text{oder} \quad 3 \neq 4$$

Im einzelnen sind das die folgenden Operationen:

<	kleiner
<=	kleiner oder gleich (kleinergleich)
>	größer
>=	größer oder gleich (größergleich)
==	gleich
!=	ungleich

Diese Operationen können natürlich auf Zahlen angewendet werden. Zum Teil lassen sie sich aber auch auf die anderen Datentypen anwenden. Zum Beispiel können Texte oder Buchstaben auf Gleichheit überprüft werden. Der Ergebnistyp einer Vergleichoperation ist immer boolean.



Theorie Boolesche Operationen

Boolesche Operationen werden auf Wahrheitswerte angewendet. Es gibt das logische Und (&&), das logische Oder (||) und das logische Nicht (!).

A && B ist nur dann erfüllt, wenn sowohl A als auch B erfüllt ist.

A || B ist dann erfüllt, wenn A oder B oder beides gilt.

!A ist erfüllt, wenn A nicht gilt.

Wenn dir das nicht verständlich genug ist, dann schau in der Tabelle nach!

	Ausdruck	Auswertung
logisches Und	true && true false && true true && false false && false	true false false false
logisches Oder	true true false true true false false false	true true true false
logisches Nicht	! true ! false	false true

Jetzt hast du alle wichtigen Operationen kennengelernt. Die Tabelle auf der nächsten Seite stellt eine Übersicht davon dar.

Du brauchst die Tabelle jetzt nicht auswendig zu lernen. Nutze sie zum Nachschauen.

Datentyp Operanden	Operator	Bedeutung	Ergebnis- datentyp	Beispiele
int	+	Addition	int	3 + 2
	*	Multiplikation	int	3 * 2
	-	Subtraktion	int	3 - 2
	-	Negation	int	-3
	/	ganzzahlige Division	int	3 / 2
	==	gleich	boolean	3 == 2
	!=	ungleich	boolean	3 != 2
	<	kleiner	boolean	3 < 2
	>	größer	boolean	3 > 2
	<=	kleinergleich	boolean	3 <= 2
	>=	größergleich	boolean	3 >= 2
double	+	Addition	double	3.2 + 2.0
	*	Multiplikation	double	3.2 * 2.0
	-	Subtraktion	double	3.2 - 2.0
	-	Negation	double	-3.0
	/	Division	double	3.2 / 2.0
	==	gleich	boolean	3.2 == 2.0
	!=	ungleich	boolean	3.2 != 2.0
	<	kleiner	boolean	3.2 < 2.0
	>	größer	boolean	3.2 > 2.0
	<=	kleinergleich	boolean	3.2 <= 2.0
	>=	größergleich	boolean	3.2 >= 2.0
boolean		logisches Oder	boolean	true false
	&&	logisches Und	boolean	true && false
	!	logisches Nicht	boolean	! true
	==	gleich	boolean	true == false
	!=	ungleich	boolean	true != false
char	==	gleich	boolean	'a' == 'b'
	!=	ungleich	boolean	'a' != 'b'
String	+	Verkettung	String	"Te" + "xt"
	==	gleich	boolean	"Te" == "xt"

Mit den unterschiedlichen Operatoren können auch komplexere Ausdrücke erstellt werden, z.B.:

```
(2 > 3) && (3 < 9)
```

Wie in der Mathematik werden die Ausdrücke in den Klammern zuerst ausgewertet. Daraus ergibt sich für den obigen Ausdruck

```
false && true
```

und das wird zu `false` ausgewertet.