

Variablen und Datentypen in Java

Variablen sind **Behälter** für Eingabe-, Ausgabe- und Rechenwerte. Sie sind vor der ersten Verwendung stets mit Angabe des Datentyps und eines Namens festzulegen. Wir beginnen den Variablennamen stets mit einem Kleinbuchstaben.

Datentypen legen fest, welche Größe der Behälter für die Variablen haben muss. Dabei orientiert man sich stets an der **Art der zu speichernden Werte**. Wir unterscheiden folgende Datentypen

Datentyp	Art der zu speichernde Werte
int	ganze Zahlen
double	(Gleit-) Kommazahlen
boolean	Wahrheitswerte (true/false)
char	ein einzelnes Zeichen
String	Zeichenfolgen/Zeichenketten

Beispiele

Festlegung und Initialisierung einer Variablen für den Wohnort:

```
String ort = "Ribnitz-Damgarten";
```

Festlegung und Initialisierung einer Variablen für die Jahrgangsstufe:

```
int stufe = 10;
```

Festlegung und Initialisierung einer Variablen für das Geschlecht:

```
boolean weiblich = true;
```

Ein- und Ausgaben mit Java

Benutzung der Eingaben

Alle Eingaben folgen stets dem Schema:

```
variable = IO.getDatentyp("Hinweistext");
```

Anstelle des Wortes Datentyp ist die Bezeichnung des zugehörigen Typs mit Großbuchstaben beginnend einzutragen:

Beispiele:

```
Spieler1 = IO.getString("Name von Spieler Nr. 1? ");  
Spieler2 = IO.getString("Name von Spieler Nr. 2? ");  
anzahl = IO.getInt("Spieler "+Spieler1+" nimmt: ");
```

Benutzung der Ausgabe

Alle Ausgaben folgen stets dem Schema:

```
IO.show("Hinweistext");
```

Sollen neben Zeichenketten auch die Inhalte von Zahlenvariablen ausgegeben werden, so sind diese durch das Pluszeichen mit den Zeichenketten zu verketteten.

Beispiel:

```
IO.show("Es bleiben "+hoelzer+".");
```

Operatoren in Java

Arithmetische Operatoren für Berechnungen

Die wichtigsten arithmetischen Operatoren sind:

=	zuweisen	/	ganzer Teil bei Division zweier ganzer Zahlen (DIV)
+	addieren	%	Rest bei Division zweier ganzer Zahlen (MOD)
-	subtrahieren	+=	erhöhen um
*	multiplizieren	-=	vermindern um
/	dividieren	++	inkrementieren (erhöhen um 1)
		--	dekrementieren (vermindern um 1)

Weitere mathematische Funktionen liefert die Systemklasse `Math`. Mit `Math.random()` wird eine Zufallszahl zwischen 0 und 1 bestimmt. Soll diese Zahl zwischen 1 und 6 liegen, so muss die Anweisung lauten: `zahl = (int) (6 * Math.random() + 1)`.

Boolesche Ausdrücke zur Definition von Bedingungen

Boolesche Ausdrücke liefern entweder den Wert *false* oder den Wert *true*. Sie werden in den Bedingungen von Verzweigungen und Wiederholungen verwendet.

In Java sind folgende Vergleichsoperatoren implementiert:

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

Boolesche Ausdrücke lassen sich mit Hilfe boolescher Operatoren kombinieren.

Die Operatoren sind:

&&	und
	oder

Zusätzlich kann mit dem Operator

!	nicht
---	-------

ein boolescher Ausdruck negiert werden.