

# JAVA-Spicker

## Elementare Datentypen

Typ	Name	Bereich	Speicher
Ganze Zahlen	byte	-128 bis 127	8 Bit
	short	-32768 bis 32767	16 Bit
	int	-2.147.483.648 bis 2.147.483.647	32 Bit
	long	-9223372036854775808 bis 9223372036854775807	64 Bit
Fließkommazahlen	float	$1,4 \cdot 10^{-45}$ bis $3,4028235 \cdot 10^{38}$	32 Bit
	double	$4,9 \cdot 10^{-324}$ bis $1,7976931348623157 \cdot 10^{308}$	64 Bit
Wahrheitstyp	boolean	true, false	8 Bit
Zeichen	char	alle Unicode-Zeichen	16 Bit
Zeichenkette	String	kein elementarer Datentyp (Objekt)	variabel

## Arithmetische Operatoren

Operator	Operation	Kurzform	Langform
+	Addition	a++ a+=b	a=a+1 a=a+b
-	Subtraktion	a-- a-=b	a=a-1 a=a-b
*	Multiplikation	a*=b	a=a*b
/	Division	a/=b	a=a/b
%	Modulo (Restklassendivision)	a%=b	a=a%b

## Boolesche Operatoren

Operator	Operation	
!	logische NICHT	!a ergibt false, wenn a wahr ist, und true, wenn a falsch ist.
&&	logisches UND	a && b ergibt true, wenn sowohl a als auch b wahr sind.
	logisches ODER	a    b ergibt true, wenn mindestens einer der beiden Ausdrücke a oder b wahr ist.
^	Exklusiv-ODER	a ^ b ergibt true, wenn beide Ausdrücke einen unterschiedlichen Wahrheitswert haben.

## Vergleichs-Operatoren

Operator	Operation
==	gleich
!=	ungleich
<	kleiner als
>	größer als
<=	kleiner/gleich
>=	größer/gleich

## Stringoperator

Operator	Operation	Kurzform	Langform
+	Stringverkettung	a+=b	a=a+b

## Variablen Deklarationen

Typname VariablenName;

## Ergibtanweisung:

variablenName=ausdruck;

der Variablen wird im Programmtext ein Variablenwert zugewiesen

**Beachte:** Jede Variable muss vor ihrer Benutzung deklariert werden, d.h. Es wird mitgeteilt, von welchem Typ sie ist.  
Häufig ist es notwendig und auch nützlich, dass sie unmittelbar nach der Deklaration initialisiert, also mit einem Anfangs- oder Initialwert belegt wird.

## Anweisungsblock:

```
{
  Anweisung1;
  Anweisung2;
  ...
}
```

## Alternativanweisungen (Verzweigung):

if (logischer Ausdruck)

ja-anweisung

else nein-anweisung;

Wenn logischer Ausdruck den Wahrheitswert TRUE hat, wird ja-anweisung abgearbeitet, sonst wird nein-anweisung abgearbeitet.

Der else-Zweig ist nicht zwingend notwendig.

switch (verteilerVariable)

```
{
  case wert1: anweisung break;
  case wert2: anweisung break;
  case wert3: anweisung break;
```

default: anweisung break;

```
}
```

Beachte: Für die verteilte Variable sind nur die Typen byte, short, int und char zulässig. Entsprechendes gilt für die werte nach case.

## Wiederholungsanweisungen (Schleifen):

1. mit bekannter Anzahl von Durchläufen (ZÄHL-Schleife):

for (initialisierung laufVariable; test laufVariable; update laufVariable) anweisung;

Beachte:

initialisierung und test der Laufvariablen sind so zu wählen, dass die Schleife im Zusammenspiel mit dem update der Laufvariablen terminiert  
update der laufVariablen organisiert die Änderung der Laufvariablen und kann beliebig gewählt werden

**Beachte:** anweisung kann bei Bedarf und wird in der Regel ein anweisungsblock sein

# JAVA-Spicker

2. mit vorangestellter Abbruchbedingung (SOLANGE-Schleife):

```
while (logischer Ausdruck)
    anweisungsblock;
anweisungsblock wird solange wiederholt ausgeführt, wie die logischer Ausdruck den Wahrheitswert
TRUE hat
Schleife ist abweisend
```

3. mit Abbruchbedingung nach dem Schleifendurchlauf (WIEDERHOL- Schleife):

```
do
    anweisungsblock
while (logischer Ausdruck);
anweisungsblock wird auch hier solange wiederholt, wie logischer Ausdruck den Wahrheitswert TRUE
hat
Schleife ist nicht abweisend
```

### Mathematische Funktionen:

sind in der Klasse Math zusammengefasst und werden entsprechend mit **Math.methodenname** aufgerufen

Funktion	mathemat. Bedeutung	Beispiel	Ergebnis
abs(arg)	Betragsfunktion	abs(-2.5)	⇒ 2.5
sqrt(arg)	Quadratwurzel	sqrt(9)	⇒ 3
round(arg)	rundet arg auf ganze Zahl	round(2.499)	⇒ 2
floor(arg)	rundet arg ab	floor(13.538) floor(-13.538)	⇒ 13.0 ⇒ -14.0
random()	liefert Zufallszahl mit 0<=Ergebnis<=1		
srand()	initialisiert den Zufallsgenerator		
sin(arg), asin(arg), cos(arg), acos(arg), tan(arg), atan(arg),	Winkelfunktionen und ihre Umkehrfunktionen (Arcussinus ...)		
log(arg)	ln(x)		
exp(arg)	e <sup>x</sup>		
pow(basis, exponent)	basis <sup>exponent</sup>	pow(2,3)	⇒ 8
PI	Kreiszahl Pi =3.141592653589793		
E	Eulersche Zahl e =2.718281828459045		

### String-Funktionen:

Für die Benutzung von Strings wurde eine Klasse definiert, in der viele Methoden zur Stringmanipulation enthalten sind. Wird später im Unterricht erweitert.

Benutzung:

```
String wort;
wort = "Willibald";
wort = wort.toUpperCase();
laenge = wort.length();
```

Funktion	Bedeutung
length()	ermittelt die Anzahl der Zeichen des Strings
toUpperCase()	setzt einen String in Grossbuchstaben um
toLowerCase()	setzt einen String in Kleinbuchstaben um
trim()	Entfernt Leerzeichen am Anfang und am End