



Vergleich Bausteine Scratch – Programmiersprache Java

Scratch	Java
<p>Variablen</p> <p>Nach dem Klick auf „Neue Variable“ gibt man den Namen der Variable ein und entscheidet, ob sie nur für die aktuelle Figur oder für alle Figuren gültig sein soll.</p>  <p>Erstellen einer neuen Variable liefert:</p> 	<p>Variablen werden entweder global (ganz zu Beginn) oder lokal (in einer Methode oder Schleife) deklariert.</p> <p>Globale Variable kennt das ganze Programm, lokale nur die Methode (Schleife), in der sie deklariert wurden.</p> <p>Bei der Deklaration der Variable müssen wir deren Typ angeben.</p> <p>Beispiele:</p> <pre>// Deklaration von Variablen int a = 4; // ganze Zahlen float b = 6.01; // Kommazahlen char c = 'A'; // ein Zeichen String text = "IMP 9"; // eine Zeichenkette boolean stimmt = false; // ein Wahrheitswert</pre> <p>Im Beispiel wurden alle Variablen gleich initialisiert. (Das könnte auch an anderer Stelle passieren.)</p> <pre>// Deklaration int zaehler; // Initialisierung zaehler = 0; // Erhöhen des Wertes der Variable zaehler = zaehler + 1; // oder zaehler += 1;</pre> <p>Der Wert der Variable kann genutzt oder verändert werden (s. Abb.).</p>
<p>Zufallszahlen</p> 	<pre>float z1 = random(100); float z2 = random(1,10); int z3 = round(random(1,6));</pre> <p>Mit <code>random(a)</code> werden Kommazahlen von 0 bis a erzeugt. Man kann auch den Bereich eingeben. Möchte man ganze Zahlen, muss man runden. (In den Materialien nutzen wir eine Methode <code>getZufallszahl(int min, int max)</code>.)</p>

```
if (klassenstufe == 9) {
    fachwahl="IMP";
}
```

Scratch

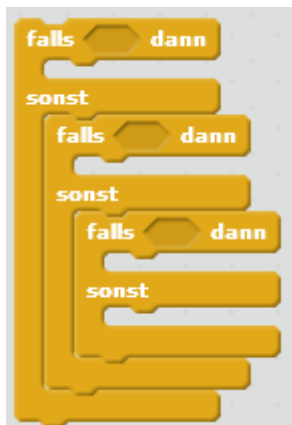
Java

Verzweigungen

Es gibt einen einfachen „falls ... dann“ - Block und einen „falls ... dann ... sonst“ - Block.



Hat man mehr als zwei Möglichkeiten, müssen Blöcke ineinander gesetzt werden.



```
if (bedingung_erfuellt) {
    // hier passiert dann etwas :)
}
```

```
if (bedingung_erfuellt) {
    // Anweisungen, wenn Bedingung erfüllt ist
} else {
    // Anweisungen, wenn sie nicht erfüllt ist
}
```

```
if (bedingung_erfuellt) {
    // Anweisungen, wenn Bedingung erfüllt ist
} else if (andere_bedingung_erfuellt) {
    // 1. Bed. ist nicht erfüllt, 2. Bed. ist erfüllt
} else {
    // keine der vorherigen Bedingungen ist erfüllt
}
```

Innerhalb der geschweiften Klammern können weitere Kontrollstrukturen stehen.

Es können auch weitere „else if“ eingefügt werden.

Eine andere Art der Verzweigung:

```
switch (klasse) {
    case 6:
        ausgabe = "Sprachwahl";
        break;
    case 8:
        ausgabe = "Profilwahl";
        break;
    case 11:
        ausgabe = "Kurswahl";
        break;
    default:
        ausgabe = "Es steht keine Wahl an.";
}
```

Lässt man das „break“ weg, werden alle Zweige durchlaufen.

Trifft kein Fall zu, werden die Anweisungen unter „default“ ausgeführt.

(wird im Material nicht genutzt)

```
if (klassenstufe == 9) {
    fachwahl="IMP";
}
```

Scratch

Java

Bedingungen

Bei den Operatoren findet man Vergleichsoperatoren und logische Verknüpfungen.

Beispiele:



Vergleichsoperatoren: >, <, >=, <=, ==

Achtung:

Ein einfaches „=" bedeutet eine Zuweisung (wie oben bei den Variablen)!

Möchte man aber überprüfen, ob eine Variable einen bestimmten Wert hat, braucht man „==“.

Logische Verknüpfungen:

&& (und); || (oder); ! (nicht)

(Bei *und* und *oder* würde auch jeweils ein Zeichen ausreichen. Nimmt man allerdings das Zeichen doppelt, wird die zweite Bedingung nur noch ausgewertet, wenn eine wahre Aussage noch möglich ist.)

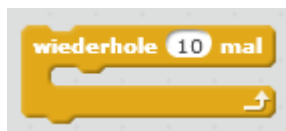
Beispiele:

```
if ( a < 3 && b > 5 ) {
    // tue etwas
}

if ( ! (a == b) ){
    // tue etwas
}
```

Schleifen

Zählschleife:



```
for (int i=start; i<=ende; i=i+schritt){
    // tue etwas
}
```

Die Zählvariable *i* läuft im Intervall [start; ende] mit der Schrittweite *schritt*.

Sie kann in der Schleife verwendet werden.

Beispiel:

```
for (int i=0; i<10; i++) {
    ellipse(30, 30, 100-i*10, 100-i*10);
}
```

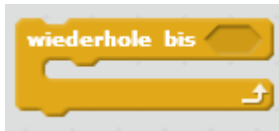
(i++ entspricht i=i+1)

Hier werden 10 konzentrische Kreise mit Mittelpunkt (30|30) gezeichnet.

Scratch

Java

Bedingte Schleife:



Die Anweisungen im Schleifenkörper werden so lange wiederholt, bis die Bedingung erfüllt ist.

Wir verwenden hier die while-Schleife, bei der die angegebene Bedingung erfüllt sein muss, um auszuführen.

```
while (bedingung_erfuellt) {
    // tue etwas
    // verändere die Voraussetzung für die Bedingung
}
```

Achtung:

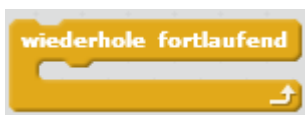
Im Schleifenkörper muss darauf geachtet werden, dass die Bedingung irgendwann nicht mehr erfüllt ist und die Schleife damit abbricht.

Beispiel:

```
while (x < width) {
    ellipse(x, 50, 20, 20);
    x=x+10; // der Wert der x-Koordinate wird erhöht
}
```

(Hier wird die x-Koordinate des Mittelpunktes so lange um 10 Pixel erhöht, bis sie größer oder gleich der Fensterbreite ist. Dann wird abgebrochen.)

„Endlosschleife“:



Eine Endlosschleife ist ein Programmierfehler und muss in Java auf jeden Fall vermieden werden.

Listen

Definition der Liste:



Auch in Java gibt es die Möglichkeit, eine Liste zu erzeugen und mit ihr zu arbeiten.

Wir beschränken uns aber auf **Felder** (Arrays).

Deklaration des Feldes:

```
Typ [] Feldname = new Typ[Elementanzahl];
```

Mit Initialisierung:

```
Typ [] Feldname = {Element 1, Element 2,...};
```

Scratch

Eingabe über „+“ oder im Code mit

füge thing zu meineliste hinzu

oder

füge thing als 1 in meineliste ein

Element löschen:

lösche 1 aus meineliste

Element ersetzen:

ersetze Element 1 von meineliste durch thing

Auf ein bestimmtes Element der Liste kann mit

Element 1 von meineliste

auf die Länge der Liste mit

Länge von meineliste

zugegriffen werden.

Java

Beispiele:

```
float[] kommazahlen = new float[10];
int[] zahlenfeld = {1, 2, 4, 8, 16, 32};
char[] zeichenfeld = {'I', 'M', 'P', '9', '!'};
```

Felder haben eine feste Größe, die mit der Eigenschaft `Feldname.length` abgefragt werden kann.

Achtung: Die Zählung beginnt bei 0!

Beispiele zum Durchlaufen von Feldern bzw. dem Füllen eines Feldes mit reellen Zufallszahlen zwischen 1 und 10:

```
for (int i=0; i<zahlenfeld.length; i++) {
    println(zahlenfeld[i]);
}

String z="";
for (int i=0; i<zeichenfeld.length;i++){
    z=z+zeichenfeld[i];
}
println(z);

for (int i=0; i<kommazahlen.length; i++){
    kommazahlen[i]=random(1,10);
    println(kommazahlen[i]);
}
```

(Ausgabe Abbildung)

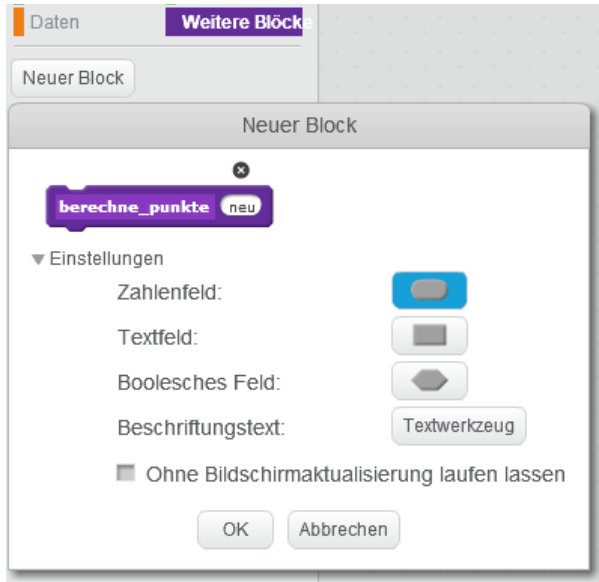
```
1
2
4
8
16
32
IMP9!
6.4687505
6.905039
2.7421758
4.0916686
4.104682
3.7311764
2.4759123
9.931702
2.7038841
6.714201
```

Scratch

Java

Unterprogramme / Methoden

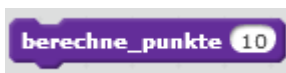
Erstellen eines neuen Unterprogramms:



Anhängen des Codes:



Aufruf des Unterprogramms:



In Java unterscheidet man Methoden mit und ohne Rückgabewert. (Siehe auch: Infoblatt zu Methoden.)

Methoden ohne Rückgabewert:

```
void name(<Übergabeparameter>) {
    ...
};
```

Methoden mit Rückgabewert:

```
Typ name(<Übergabeparameter>) {
    ...
    return rückgabewert;
}
```

Beispiele:

```
void zeichneKreis(int x, int y, int r, int farbe){
    fill(farbe);
    ellipse(x,y,r,r);
}

String begruessung(String vorname, String name){
    return "Hallo "+vorname+" "+name+"!";
}

int groesser(int zahl1, int zahl2){
    int erg;
    if (zahl1>zahl2) {
        erg = zahl1;
    } else {
        erg = zahl2;
    }
    return erg;
}
```

Aufruf der Methoden:

```
zeichneKreis(100, 200, 50, #FF0000);
text(begruessung("Hieronymus", "Hase"), 200, 100);

int max;
max = groesser(40, 56); // max erhält hier den Wert 56
```

```
if (klassenstufe == 9) {  
    fachwahl="IMP";  
}
```

Scratch wird von der Lifelong-Kindergarten-Group am MIT-Media-Lab entwickelt. Siehe <http://scratch.mit.edu>.
Scratch ist lizenziert unter *CC BY-SA 2.0* (<https://creativecommons.org/licenses/by-sa/2.0/deed.en>).