Parametertypen

Ein Attributwert, der von einer Methode/Konstruktor eingelesen wird heißt **aktueller Parameter** .

Formaler Parameter heißt die in der Signatur festgelegte temporäre Variable, die **den aktuellen Parameter kurzzeitig abspeichert**. Er steht einem Objekt **nur im Rumpf des Konstruktors** oder **der Methode** zur Verfügung. Er kann also nicht an anderen Stellen einer Klasse genutzt werden, wenn er nicht an eine globale Variable übergeben wird.

```
public void anstreichen (String farbe, int anzahl, boolean fertig)
{    ...    }
```

Anzahl, Reihenfolge und **Datentyp** der **aktuellen** Parameter müssen den **formalen** Parametern entsprechen.

```
anstreichen ("rot",3, true)  nicht: anstreichen ("rot")  oder anstreichen (true,"rot",3)
```

Parameter übergeben den Methoden die Attributwerte.
Formaler Parameter
Die temporäre Variable, die in der Signatur einer Methode in die Parameterliste geschrieben wird.
Aktueller Parameter
Der Wert oder die Variable, die beim Aufruf der Funktion übergeben wird.

Manchmal werden innerhalb einer Methode für Werte kurzeitige Speicherplätze benötigt oder der Wert ist nur für eine einzelne Aufgabe innerhalb einer Methode erforderlich. (z.B. Index bei Schleifen).
--> **Lokale Variable** werden **innerhalb einer Methode definiert**.
Ihre **Sichtbarkeit** und **Lebensdauer sind auf die definierende Methode/Konstruktor beschränkt.**

	Globale Variable	*Formaler Parameter*	*Lokale Variable*
Wo definiert?			
Wie definiert?			
Lebensdauer/Verfügbarkeit			
Beispiel			

Kontrollstrukturen.

1.Alternativen/Verzweigung

Bedingte Anweisung (wenn ... dann ... sonst)

einseitige Abfragen

```
if (prüfung) {Anweisungen, wenn prüfung true; };


z.B.
if(x==5) {System.out.println("Fünfe")};
```

zweiseitige Abfragen

```
if (prüfung)   {Anweisungen, wenn prüfung true; }
else {Anweisungen, wenn prüfung false; };


z.B.
if(x!=5){System.out.println("Nisch Fünf!")}
else {System.out.println("Is Fünf")};
```

Mehrfachabfrage (switch Struktur)

```
switch (variable) {
        case wert1: anweisungen; break;
        case wert2: anweisungen; break;
        case wert3: anweisungen; break;
        ...
       default: ..  weitere anweisungen;     };
```

```
z.B. char wahl;

switch (wahl) {
case 'a':System.out.println("Amsel");break;
case 'b':System.out.println("Biber");break;
case 'c':System.out.println("Crocodil");break;
default: System.out.println("Wees net");break;
```

2.Wiederholungsanweisungen (Schleifen)

Die vorprüfende Wiederholungsanweisung (solange ...tue)

while (prüfung**) {** anweisungen; wertänderung; **} ;**
z.B. int a=-3; while (a<5) {System.out.println(a);a++;} ;

Die nachprüfende Wiederholungsanweisung (tue... solange)

do { anweisungen ; **} while (**prüfung**);**
z.B. int a=-3;do {System.out.println(a);a++;} while (a<5);

Die Zählschleife (oft auch **for-Schleife** genannt)

for (startwert **;** prüfung **;** änderung des startwertes**) {** anweisungen**; } ;**
z.B. for (int i=15,i>=6,i--) {System.out.println(i);};

Hinweis:

++	**Inkrement**	**erhöht einen Indexwert um 1**	z.B. x++;
--	**Dekrement**	**erhöht einen Indexwert um 1**	z.B. x--;
Beliebiges Verändern des Indexwertes			z.B. x=2*x+3;

Für die Planung der Kontrollstrukturen einer Methode verwendet man Struktogramme.
(siehe Arbeitsblatt „Kontrollstrukturen")

Boolesche Ausdrücke

Die Bedingung muss einen Wahrheitswert haben ⇒ ***Boolescher Ausdruck*** (nur true und false), entstehen durch Vergleichen.

- **a > b** **größer**
- **a == b** **gleich**
- **a<= b** **kleiner gleich**
- **a != b oder !a = b** **ungleich**

Boolesche Ausdrücke können mit Operatoren verknüpft werden

- a && b	**und** Verknüpfung	beide Bedingungen müssen true sein
- a || b	**oder** Verknüpfung	eine der beiden Bedingungen muss true sein
- !b	**nicht**	Bedingung muss false sein

Die Klasse Math-Besondere Rechenoperationen in Java

Schreibe über den Klassenkopf: import java.lang.Math;

double wert = **Math.PI;**	**Pi**
double wert = **Math.sqrt (**x**) ;**	**x-te Wurzelziehen**
double wert = **Math.pow (**x,y**);**	**Potenz x^y**
double wert = **Math.abs (**x**);**	**Betrag von x**
double wert = **Math.sin (**x**);**	**Trigonometrische Fkt. (auch cos, acos...)**
double wert = **Math.min (**x**);**	**Minimum**
double wert = **Math.max (**x**);**	**Maximum**...
double wert = **Math.log (**x**) ;**	**Logarithmus**
int wert = **(int) (Math.random( )***d**)**+untereGrenze**;**	**Zufallszahl erzeugen zwischen** untereGrenze **und** untereGrenze+d
int wert = (int) (Math.random( )*20)+50	Zufallszahl zwischen 50 und 70