

Methodendeklaration

```
public void methodenname()  
{  
    befehl();  
}
```

Beispiel:

```
public void move5()  
{  
    kara.move();  
    kara.move();  
    kara.move();  
    kara.move();  
    kara.move();  
}
```

if-Anweisung

```
if (boolescherAusdruck) {  
    befehl();  
}  
else {  
    alternativerBefehl();  
}
```

Beispiel:

```
if (kara.onLeaf())  
    kara.removeLeaf();  
else  
    kara.putLeaf();
```



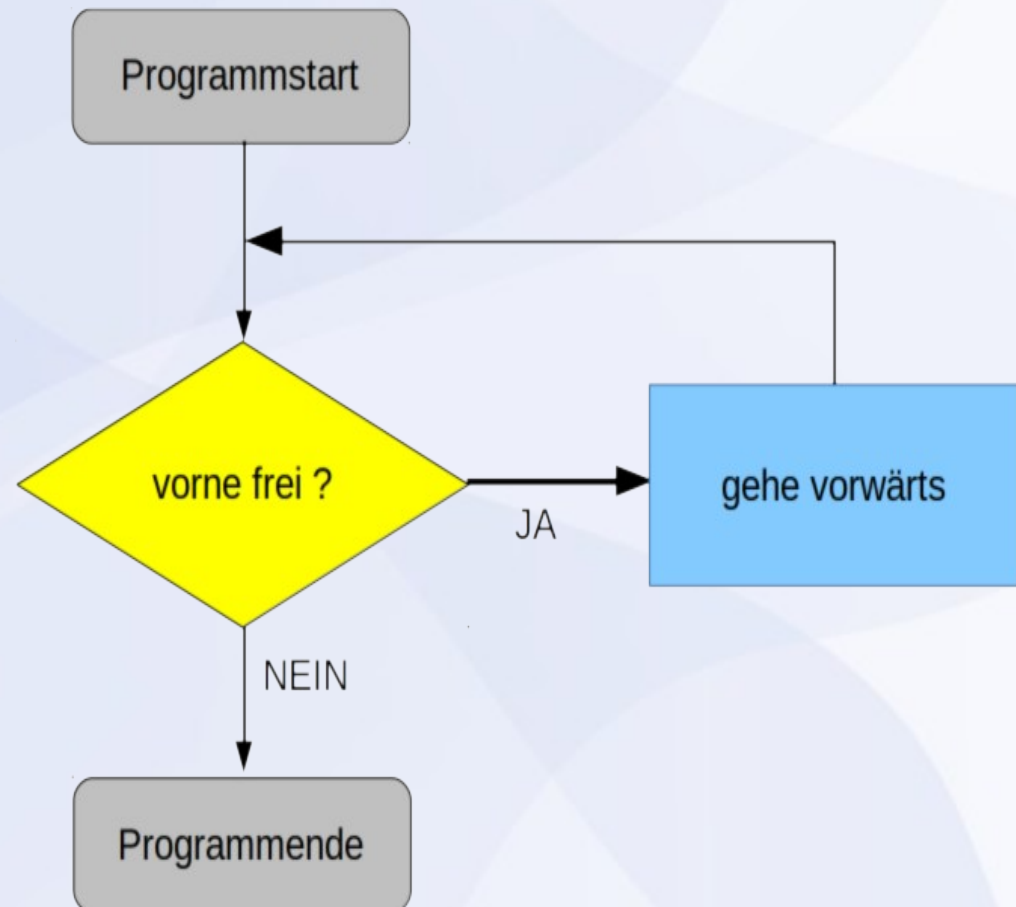
while-Schleife

```
while (bedingung)
{
    Anweisungsblock
}
```

Wiederhole den Anweisungsblock,
solange die Bedingung erfüllt ist.
bedingung steht für einen
booleschen Ausdruck,
z.B. eine Testfunktion

Beispiel:

```
while (!kara.treeFront())
    kara.move();
```



for-Schleife

Die `for`-Schleife hat einen eingebauten Schleifenzähler. Man verwendet sie, wenn vor vorneherein `jakar` ist, wie oft wiederholt werden soll.

```
for ( int i=a; i<=e; i++ ) {  
Anweisungsblock; }
```

Schleifenzähler : `i`
Anfangswert : `a`
Endwert : `e`

Beispiel:

```
for ( int i=1; i<=10; i++ )  
    tools.showMessageDialog(""+i*i);
```

i	Ausgabe
1	1
2	4
...	
10	100

Integer-Variablen

- **Deklaration**

Namesgebung und Reservierung von Speicherplatz

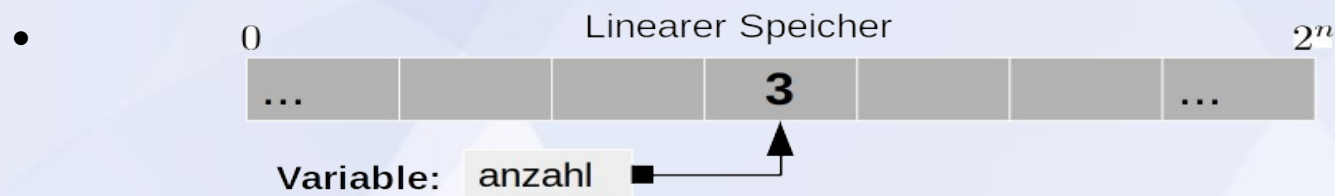
```
int anzahl;
```

- **Variablentyp:** int
Variablenname: anzahl

- **Wertzuweisung**

Veränderung des abgespeicherten Wertes

```
anzahl = 3;  
anzahl = anzahl + 1;
```



Sinus-Maschine

π



Eingabe

$x \rightarrow \sin(x)$

Ausgabe



0

Funktionen

Funktionen sind spezielle Methoden. Funktionen haben gegebenenfalls **Übergabeparameter** und **einen Rückgabewert**.

```
public rueckgabetyp funktionsname (eingabetyp x) {  
    → Berechnung des Rückgabewerts y  
    return y;  
}
```

x : Übergabeparameter **y** : Rückgabewert

Beispiel:

```
public double quadrat (double x) {  
    return x * x;  
}
```

Aufruf: `System.out.println(quadrat(2));`

Array (Tabelle)

Deklaration:

```
int[] eineTabelle;
```

Initialisierung:

```
eineTabelle = new int[anzahlFelder];
```

Zugriff:

```
eineTabelle[position] = wert;
```


Klassen, Objekte, Methoden

Klasse	Eine Klasse umfasst eine Sammlung von Objekten mit gemeinsamen Eigenschaften. Die zu einer Klasse gehörigen Objekte haben gemeinsame Attribute und Methoden .
Objekt	Ein Objekt (eine Instanz) ist die Realisierung einer Klasse. Es können beliebig viele zu einer Klasse gehörige Objekte existieren.
Methode	Eine Methode wird auf ein Objekt angewendet, um dessen Zustand abzufragen oder zu verändern.

Klassendeklaration

```
public class Klassenname { ... }
```

Der Klassenname beginnt immer mit einem Großbuchstaben.

Die Klassendeklaration umfasst:

- Attribute
- Konstruktoren
- Methoden

Beispielklasse

```
public class Beispielklasse {  
  
    // Attribute  
    private Object einAttribut;  
  
    // Konstruktor (Initialisierung der Attribute)  
    public Beispielklasse (...) {  
        einAttribut = new Object();  
    }  
  
    // Methoden  
    public Object getAttribut(){  
        return einAttribut;  
    }  
}
```