

# Informatik 10

**Basisinformationen zu Informatik 10. In der Jahrgangsstufe 10 ist der große Themenbereich das Erlernen des objektorientierten Programmierens.**

## Python auch im Jahr 2020 im tiobe-index weiterhin aufsteigend

Python als Programmiersprache und Gewinner des Jahres 2018 ist weiterhin in der Relevanz der Programmiersprache aufsteigend.

vgl. [tiobe.com/tiobe-index](https://tiobe.com/tiobe-index)

## Aufgaben in Informatik 10

- **Grundlagen der Informatik** (Wdh)  
Klasse, Objekt, Attribut, Methode, Klassen- und Objektkarte, Klassendiagramm, HTML, Datentypen und Datenbanken (Wdh)
- **Datenmodellierung** (Wdh): ZWEI Datenbanken (DB) mit den benötigten Tabellen (LibreOffice Calc, Microsoft Access) zu den Themen "Musik-DB" und "Photo/ Filmarchiv" anlegen  
(wenige (auch fiktive) Datensätze reichen für eine SQL-Abfrage)
- Digitales Audio, vgl. [learningmusic.ableton.com](https://learningmusic.ableton.com) (zur Klärung eines persönlich musikalisch favorisierten Klassendiagramms mit relevanten Attributen)
- <https://metmuseum.org/about-the-met/policies-and-documents/open-access> (Kunstgeschichts-Bilder mit public-domain-Lizenz)
- SQL Lite (py), LibreOffice Base, Microsoft Access (DB)
- **HTML** (Wdh) webSeite mit Wordpress ([wordpress.com](https://wordpress.com)) editieren
- **Java/ Processing** | OOP (Objektorientierte Programmierung) mit Java/  
Processing: <https://processing.org> (Vereinfachte Java-Version), [openprocessing.org](https://openprocessing.org)  
Processing ist eine objektorientierte, stark typisierte Programmiersprache mit zugehöriger integrierter Entwicklungsumgebung. Die Programmiersprache ist auf die Einsatzbereiche Grafik, Simulation und Animation spezialisiert. Processing wird in einem quelloffenen Projekt entwickelt, das am Massachusetts Institute of Technology in Boston von Ben Fry (Broad Institute) und Casey Reas (UCLA Design | Media Arts) initiiert wurde. Processing hat den Charakter einer stark vereinfachten Version der Programmiersprache Java, ermöglicht Interaktionen und visuelle Elemente zu programmieren und richtet sich vorwiegend an Gestalter, Künstler und Programmieranfänger.
- **Python** | OOP  
vgl. [cscircles.cemc.uwaterloo.ca](https://cscircles.cemc.uwaterloo.ca)  
[cscircles.cemc.uwaterloo.ca > cheatsheet.pdf](https://cscircles.cemc.uwaterloo.ca/cheatsheet.pdf)

OOP Referate (**Python** | Java) (Lit. z.B. [python4kids.net](https://python4kids.net) > Wie ein Informatiker denken lernen)

Besonders relevant sind ALLE Glossare zur Begriffs-Definition.

Kap02 **Variablen, Ausdrücke und Anweisungen** | Datentypen, = Zuweisung == Gleichheit, 29 Schlüsselwörter

Kap03 **Funktionen** | 3.6 Neue Funktionen, Variable lokal/global, Rekursion - Abbruchbedingung wegen Gefahr der Endlosschleife, Vergleichsoperatoren

Kap04 **Verzweigung und Rekursion** | Mehrfache Verzweigung

Kap05 **Funktionen mit Wert** | return-Anweisung

Kap06 **Iteration** | Funktionen - Vorteile

Kap07 **Strings** | Diff. Strings/ Listen, String-Abschnitte

Kap08 **Listen** | 8.1 Werte, 8.2 Zugriff, 5.5 Listen und For-Schleifen, 8.8 Listen veränderbar, Elemente löschen, Objekte und Werte, Alias, Klone, Listen als Parameter, Geschachtelte Listen -8.15 Matrix

Kap09 **Tupel**

Kap10 **Datentyp Dictionary**

Kap11 **Dateien und Ausnahmen**

Kap12 **Klassen und Objekte** | Klassendefinitionen

Kap13 **Klassen und Funktionen**

Kap14 **Klassen und Methoden**

Kap15 **Mengen von Objekten**

Kap16 **Vererbung**

Kap17 **Verkettete Listen**

Kap18 **Stacks**

Kap19 **Queues**  
 Kap20 **Bäume**

## py | Literatur zu Python und Games (englisch)

Teil 1

inventwithpython.com/invent4thed

Teil 2

inventwithpython.com/pygame

- **OOB Programmierprojekt** mit objektorientierter Programmierung z.B. turtle Grafik (Python)  
 #Codierung aufschlussreich kommentieren.  
 #Eigene Projektdokumentation erstellen
- Alternativ: **5-Seiten-Artikel schreiben**. Programmier- bzw. Rechercheprojekt  
 Ihr könnte einen Artikel, d.h. gut recherchierten Aufsatz schreiben, der etwas umfangreicher ist als ein Referat und das in der Erstellungsphase im Kurs wie in einem Kolloquium vorgestellt wird - von Internet-Kunst, webDesign, Anwendung von QR-codes, pygame py-Bibliotheken, jupyter-notebooks for python oder dergleichen - einfach ein informatisches Spezialthema eures Interesses. Seitenumfang: ca. 5 Seiten Text (ohne Abb. oder Programm-Code).  
 Die Vorstellung im Kurs/ in der Klasse kann sein:
  - Vorstellung der Kurz-Gliederungspunkte mit einem Beispiel-Thema
  - Vorstellung einer Argumentation oder eines Kernaspekts
  - Vorstellung einer geschriebenen Textpassage
  - Vorstellung von Code-Beispielen

Wir wiederholen Grundlagen aus der Informatik der Vorjahre und programmieren mit objektorientierter Programmiersprache (Python | Java), OOP.

Konkret fangen wir mit kleineren Übungen an, z.B. mit dem Blocksystem von Scratch (<http://scratch.mit.edu>) und der graphischen Python turtle Programmierung

Danach gehts systematisch in Programmierkenntnisse von Python und später auch Java - wir behandeln hier die Unterschiede von Python und Java.

Jeder hat als Aufgabe, ein größeres oder mehrere kleinere Informatik-Projekte mit Python oder Java umzusetzen.

## python turtle Grafik

Mit der python IDLE grafische Objekte zeichnen

vgl. Skript Kurzbefehle

Befehl, Langform	Kurzformen	Bedeutung
<b>Turtle-Aktionen</b>		
<b>Bewegen und Zeichnen</b>		
forward(px)	fd()	bewegt T. um <i>px</i> Pixel nach vorne
backward(px)	bk(), back()	bewegt T. um <i>px</i> Pixel nach hinten
right(winkel)	rt()	dreht T. um <i>winkel</i> im Uhrzeigersinn
left(winkel)	lt()	dreht T. um <i>winkel</i> gegen den Uhrzeigersinn
setposition(x,y)	setpos(), goto()	positioniert T. auf <i>x- und y</i> -Koordinaten
setx(x), sety(y)		positioniert T. auf <i>x- oder y</i> -Koordinate
setheading()	seth()	setzt absolute Blickrichtung der T.
home()		positioniert T. auf Startkoordinaten
circle(radius, winkel, ecken)		zeichnet Kreis gemäß <i>radius</i> ; falls <i>winkel</i> entspr. Kreisbogen; falls <i>ecken</i> entspr. Polygon
dot(größe, farbe)		zeichnet Punkt ggf. mit entspr. <i>größe</i> und <i>farbe</i>
stamp(), clearstamp()		
speed(zahl)		steuert T.geschwindigkeit gemäß <i>zahl</i> : 1-10
undo()		macht letzte Anweisung rückgängig
<b>Status abfragen</b>		
position()	pos()	gibt Position als Koordinatentupel zurück
towards()		
xcor(), ycor()		gibt <i>x- oder y</i> -Koordinate zurück
heading()		gibt T.orientierung zurück; 0 = ost, 90 = nord
distance(x,y)		berechnet Abstand der T. zum Punkt P(x y)

Kontrolle des Stifts		
Stift setzen		
pendown()	down(), pd()	bringt S. aufs Papier
penup()	up(),up()	nimmt S. vom Papier
pensize(breite)	width()	setzt Stiftbreite auf Pixelbreite
pen()		gibt Stiftstatus als Dictionary zurück
isdown()		gibt <i>True</i> zurück, falls Stift schreibbereit
Farben		
color()		gibt pencolor() und fillcolor() zurück
pencolor(), fillcolor()		gibt Strich-/Füllfarbe als Namen oder RGB-Tupel zurück
pencolor(farbnamen)		setzt Strichfarbe gemäß Farbnamenszeichenkette
pencolor(r,g,b)		setzt Strichfarbe gemäß RGB-Komponenten (1-255)
filling()		gibt <i>True</i> zurück, falls Füllmodus aktiviert
begin_fill(), end_fill()		Füllmodus (de-)aktivieren
Weitere Zeichenoptionen		
reset()		Gezeichnetes löschen und T. in Ausgangsposition zurück
clear()		nur Gezeichnetes löschen
write(text, move, align, font)		gibt <i>text</i> entspr. <i>align</i> = 'left' mittels <i>font</i> = ('Arial',8,'normal') aus; <i>m</i>
Turtle state		
Sichtbarkeit und Erscheinung		
showturtle()	st()	zeige Turtle
hideturtle()	ht()	verstecke Turtle
isVisible()		gibt <i>True</i> oder <i>False</i> zurück
shape() resizeMode() shapesize()   turtlesize() shearfactor() settiltangle() tiltangle() tilt() shapetransform() g		
Mausevents nutzen		
onclick()		
onrelease()		
ondrag()		
Spezielle Turtle-Methoden		
		begin_poly() end_poly() get_poly() clone() getturtle()   getpen() getscreen() setundobuff
Turtle-Fenster		
Fensterkontrolle		
bgcolor()		setzt Hintergrundfarbe; Standard = "white"
bgpic(picname)		setzt/liest Hintergrundbild
clearscreen()	clear()	
resetscreen()	reset()	
screenSize(x,y, bgcolor)		setzt/liest Fensterbreite <i>x</i> und Fensterhöhe <i>y</i> ; auch Angabe von <i>bg</i>
setworldcoordinates(lux,luy,rox,roy)		setzt Koordinatensystem neu: lu=links unten; ro=rechts oben

## Scratch

<http://scratch.mit.edu> > Entwickeln

## OOP Java/ Processing | openprocessing

Processing ist eine objektorientierte, stark typisierte Programmiersprache mit zugehöriger integrierter Entwicklungsumgebung. Die Programmiersprache ist auf die Einsatzbereiche Grafik, Simulation und Animation spezialisiert. Processing wird in einem quelloffenen Projekt entwickelt, das am Massachusetts Institute of Technology in Boston von Ben Fry (Broad Institute) und Casey Reas (UCLA Design | Media Arts) initiiert wurde. Processing hat den Charakter einer stark vereinfachten Version der Programmiersprache Java, ermöglicht Interaktionen und visuelle Elemente zu programmieren und richtet sich vorwiegend an Gestalter, Künstler und Programmieranfänger.

[processing.org](http://processing.org) | OOP (Objektorientierte Programmierung) mit Java/ Processing, Vereinfachte Java-Version

[processing.org/examples](http://processing.org/examples) | Beispiele

[hello.processing.org/editor](http://hello.processing.org/editor) | Tutorial

[py.processing.org/reference](http://py.processing.org/reference) | Python/ Processing

[openprocessing.org](http://openprocessing.org)

[openprocessing.org/browse/#](http://openprocessing.org/browse/#)

## Python-Referat I | (Bsp-Aufgaben, Wdh, mit py computer science circles (Uni Waterloo))

Es gibt hier 18 gut verständliche Kapitel zur Objektorientierten Programmierung mit Python anhand von Programmier-Beispielen.

Referat eines der Kapitel:

Vortrag auf deutsch mit Verwendung [cscircles.cemc.uwaterloo.ca/de](http://cscircles.cemc.uwaterloo.ca/de)

Präsentation in englisch [cscircles.cemc.uwaterloo.ca](http://cscircles.cemc.uwaterloo.ca)

Übungen mit Codierung direkt ausführen und kommentieren.

## Python-Referate II (Überblick, Wdh, in Teamarbeit zu zweit)

Recherche z.B. in Online-Quellen, Lit:

**Python für kids:** <http://python4kids.net/how2think/index.html>

**Kapitel 1: Über das Programmieren**

**Kapitel 2: Variablen, Ausdrücke und Anweisungen**

**Kapitel 3: Funktionen**

**Kapitel 4: Verzweigung und Rekursion**

**Kapitel 5: Funktionen mit Wert**

**Kapitel 6: Iteration**

**Kapitel 7: Strings**

**Kapitel 8: Listen**

**Kapitel 9: Tupel**

**Kapitel 10: Der Datentyp Dictionary**

**Kapitel 11: Dateien und Ausnahmen**

**Kapitel 12: Klassen und Objekte**

**Kapitel 13: Klassen und Funktionen**

**Kapitel 14: Klassen und Methoden**

**Kapitel 15: Mengen von Objekten**

**Kapitel 16: Vererbung**

**Kapitel 17: Verkettete Listen**

**Kapitel 18: Stacks**

**Kapitel 19: Queues**

**Kapitel 20: Bäume**

**Python Lernen | Literatur in Englisch | [w3schools.com](http://w3schools.com)**

w3schools.com/python | py  
und

w3schools.com/python/python\_examples.asp | py examples

w3schools.com/python/exercise.asp | py exercise

w3schools.com/quiztest/quiztest.asp?qtest=PYTHON | py Quiz

w3schools.com/python/showpython.asp?filename=demo\_default | py run example

## Python-Projekt, eigenes (kleines) | Taschenrechner oder Memory/ Puzzle-App

**Taschenrechner** (vgl. Klassenordner auf cloud.sgm.de) mit kinter

oder Memory/ Puzzle-App

vgl.

pythonprogramming.altervista.org/memory-game-with-python-in-no-time/

oder

inventwithpython.com/pygame/chapter3.html

## Java/ Processing | openprocessing

openprocessing.org/browse/#

OOP (Objektorientierte Programmierung) mit Java/ Processing: <https://processing.org> (Vereinfachte Java-Version), openprocessing.org

## Referate Informatik 10 I (5 min Vortrag)

wichtig ist hierbei jeweils: Beschreibung | Definition | Beispiel | Abbildung

Bei jedem Referat soll am Ende ein kurzer Hefteintrag (Textpassagen zum Abschreiben bzw./und Graphik oder Abbildung zum Abzeichnen) die letzte Seite sein.

## **Informatik | Objektorientierte Programmierung mit Python oder Java**

if/else - Bedingung  
 while - Schleife  
 Objekte | py  
 Klassen | py z.B. Liste (unsortiert, sortiert), Liste einfach verkettet, Schlange, Stapel, Baum, Graph (ungewichtet, gewichtet)  
 Klassenkarten | py  
 Objektkarten | py  
 Klassendiagramm  
 Objektdiagramm  
 Zustandsdiagramm  
 Variablen | py  
 Methoden | py  
 Anweisungen | py  
 Strings (Zeichenketten) | py  
 Vererbung | py  
 Verzweigung/ Rekursion  
 Bsp. Koch'sche Schneeflocke  
 Bsp. Sierpinski-Dreieck  
 Bsp. Fibonacci-Funktion  
 Lastenheft/ Pflichtenheft in der Informatik  
 Agile Softwareentwicklung (vs Wasserfall-Modell)  
 Objektorientierte Programmierung - Diff. python, Java  
 Biometrische Daten als Schlüssel  
 Künstliche Intelligenz (Artificial Intelligence AI)  
 Intuitive Bedienbarkeit/ User-Dialog/ GUI - Graphische Oberfläche und optisches Design  
 Eclipse-Programmierungsumgebung  
 Android-Programmierung  
 python-Turtle-Programmierung  
 App-Programmierung

## **Grundlagen der Informatik | basic**

Open Source  
 CC, Copyright und Lizenzen  
 KI - Künstliche Intelligenz  
 Big Data - Überwachung, Online-Tracking, Datenschutz  
 Serious Games, technologiegestütztes Lernen

Geschichte/ Meilensteine der Informatik  
 Klasse Objekt Attribut Methode. Punktnotation  
 Klassenkarte Objektkarte Klassendiagramm | py  
 Beziehungen und Kardinalitäten  
 Methoden | py  
 HTML | atom.io oder notepad++  
 Algorithmus/ algorithmisches Denken  
 Struktogramme  
 Baum und Graph | Hierarchische Strukturen  
 Binäres Zahlensystem  
 Rechnerarchitektur

Mögliche Themen als Wiederholung aus Informatik 09, Funktionale Programmierung (Excel/ Calc) und Datenbanken (Access/ Base)

mit Blick auf Umsetzungen in Python

## **Funktionale Programmierung | py**

Funktionen  
 Vordefinierte Funktionen

Funktionen *UND*, *ODER*, *NICHT*  
 Logische Funktionen  
 Datenflussdiagramm  
 Bedingte Funktion: *WENN*-Funktion  
 while-Schleife  
 if-/else - Bedingte Anweisung  
 Zusammenfassen von Funktionen  
 Verknüpfung von Tabellen  
 Verzweigung im Datenfluss  
 Geschichte der Tabellenkalkulation  
 Tabellenkalkulation - Buchhaltung  
 Diagramme aus Tabellendaten  
 Kapital-Berechnung (Kredit-Aufnahme, Geld-Anlegen, Zinsberechnung mit Zinstagen)  
 Zusammengesetzte Daten (z.B. Brüche)  
 Verkaufspreis mit Mengenrabatt  
 Verkaufspreis mit Skonto  
 Kostenvoranschlag (Rechnung) mit Tabellenkalkulation  
 Adress-Listen mit Tabellen  
 Diagramme in Tabellenkalkulation: Entwicklung der Weltbevölkerung  
 Kopieren - relative und absolute Adressen  
 Links - lokal und online z.B. in Tabellen

## Datenbanken | py

DB statt Fkt - warum?  
 Datentypen | py  
 Relationales Datenbanksystem  
 SQL-Abfragen  
 Datenbankschema  
 Datenpflege (UPDATE, INSERT, DELETE)  
 Datensicherheit und Datenschutz  
 Klassendiagramm und Kardinalitäten  
 n:m - Beziehung  
 Das kartesische Produkt  
 Aggregat-Funktionen von SQL (COUNT, AVG, MAX, MIN, SUM)  
 Schlüssel, Fremdschlüssel und referentielle Integrität  
 Datenpflege (Ändern, Einfügen, Löschen von Datensätzen | UPDATE, INSERT, DELETE)  
 Änderungsanomalien bei der Datenpflege  
 Datenkonsistenz  
 Datensicherheit und Datenschutz  
 Datenbankschema  
 Datenbank-Design  
 Prüfziffer am Bsp. der ISBN-Nummer

## Scratch | [scratch.mit.edu](https://scratch.mit.edu) (Wdh) oder Snap

### Programmieren mit Puzzle-Bausteinen

[scratch.mit.edu](https://scratch.mit.edu)

> Entwickeln

[de.scratch-wiki.info/wiki/Einsteiger-Tutorials](https://de.scratch-wiki.info/wiki/Einsteiger-Tutorials)

## Grundwissen Informatik 10

Generell ist eine sich vertiefende und vermehrt systematische Kenntnis einer objektorientierten Programmiersprache (Python | Java) als Grundwissen in Informatik 10 relevant.

### Objekte:

Als Objekte bezeichnet man alle Gegenstände, Dinge, Lebewesen, Begriffe oder Strukturen unserer Welt (z. B. Autos, Räume, Bakterien, Lehrer, Schüler, Kunden, Unterrichtsstunden, Buchungen, Konten, ...). Jedes Objekt besitzt einen eindeutigen Bezeichner (Objektnamen), mit dem es identifiziert werden kann.

### Attribut:

Als Attribute bezeichnet man die Eigenschaften, durch die sich ein Objekt von anderen unterscheidet. Bei jedem Objekt haben die Attribute bestimmte Werte, Attributwerte.

### Methode:

Eine Methode ist eine Handlungsanweisung.

Zur Methodendefinition: *Aufbau und Rückgabetypen von Methoden*

```

Rückgabotyp void: void = leer      Eingabe-Attribut mit domain
public void setzeFarbe(int farbnrNeu){
    farbnr = farbnrNeu;
}

Rückgabotyp int: int = ganzzahl
public int gibUmfang(int laenge, int breite){
    int umfang = 2*laenge + 2*breite;
    return umfang;
}
Definition der lokalen Variablen umfang

```

Damit ein Objekt eine Methode ausführt, muss ihm dazu eine Handlungsanweisung gegeben werden, d. h. eine Methode aufgerufen werden.

Zum Methodenaufruf:

objektname.methodenname( wert1, wert2, ...)      z. B.:    max.oeffne(Fenster)

### Konstruktor

Ein Konstruktor ist eine spezielle Methode, die ein Objekt der zugehörigen Klasse erstellt und dessen Attributen Startwerte zuweist.

Der Aufruf des Konstruktors mit dem new-Operator erzeugt ein neues Objekt einer Klasse:

```

Schueler max = new Schueler("Max", "Mustermann");

```

Diagramm zur Erklärung des Aufrufs:

- Objektname: max
- new-Operator: new
- Klassenname: Schueler
- Parameter: "Max", "Mustermann"

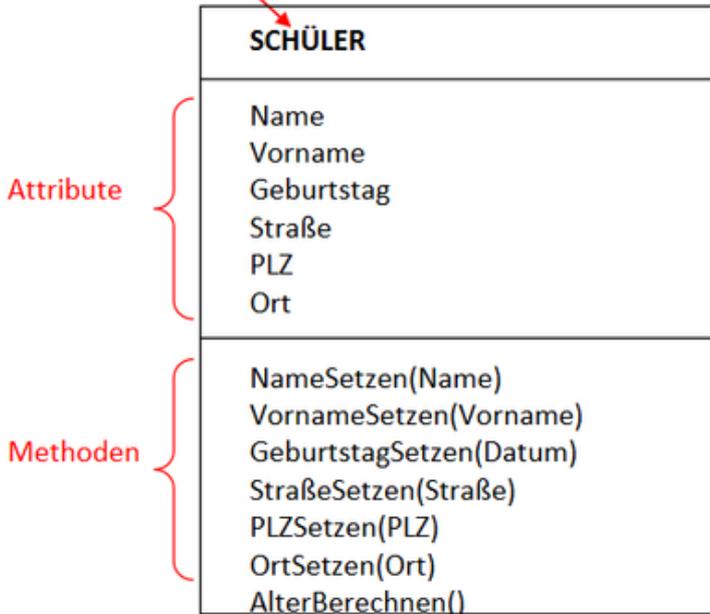
### Klasse (Objekttyp):

Eine Klasse ist ein "Bauplan" für Objekte mit gleichen Attributen und gleichen Methoden (alle Auto-Objekte gehören zur Klasse Auto). Wichtig: Alle Objekte einer Klasse, haben die gleichen Attribute, aber nicht die gleichen Attributwerte!

### Klassenkarte:

z. B.

Klassenname

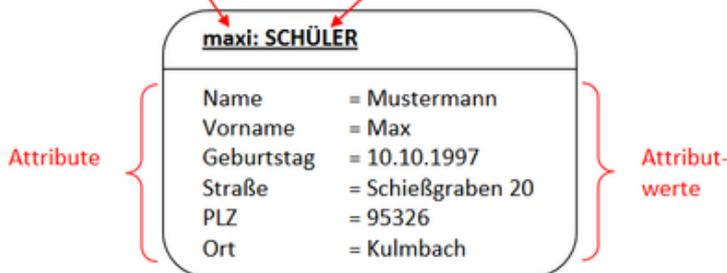


**Objektkarte:**

z. B.

Objektname

Klassenname



**Algorithmus**

Ein Algorithmus ist eine Verarbeitungsvorschrift, die aus **endlich vielen eindeutig ausführbaren Anweisungen** besteht.

**Struktogramm**

Ein Struktogramm ist eine Darstellung für Algorithmen, die von der konkreten Schreibweise einer bestimmten Programmiersprache unabhängig ist.

**Sequenz**

Eine Folge von Anweisungen, die nacheinander abgearbeitet werden.

Struktogramm:

z. B. in Java:

```
Schueler max = new Schueler();
max.vornameSetzen("Max");
max.nachnameSetzen("Mustermann");
...
```

Struktogramm:



**Wiederholung mit fester Anzahl**

Soll eine Sequenz mehrmals wiederholt werden und steht die Anzahl der Wiederholungen fest, verwendet man zur Vereinfachung die Wiederholung mit fester Anzahl.

z. B. in Java:

```
for (int i = 0; i < 10; i = i + 1) {
    ...
}
```

Struktogramm:



**Bedingte Wiederholung mit Anfangsbedingung**

Die Anweisungen werden wiederholt, solange eine bestimmte Bedingung erfüllt ist. Ist die Bedingung zu Beginn nicht erfüllt, wird die Wiederholung kein einziges Mal ausgeführt.

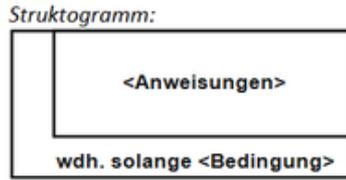
```
Java:
while (Bedingung) {
    ...
}
```



**Bedingte Wiederholung mit Endbedingung**

Die Anweisungen werden auf jeden Fall einmal ausgeführt, egal ob die Bedingung zutrifft oder nicht. Sie werden wiederholt, solange die Bedingung erfüllt ist.

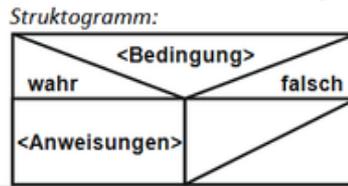
```
Java:
do {
    ...
} while (Bedingung);
```



**Einseitig bedingte Anweisung**

Die einseitig bedingte Anweisung wird verwendet, um eine Sequenz abhängig von einer Bedingung auszuführen.

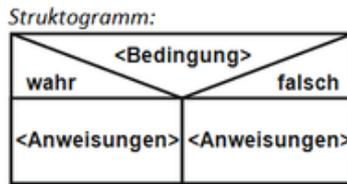
```
Java:
if (Bedingung) {
    ...
}
```



**Zweiseitig bedingte Anweisung**

Die zweiseitige bedingte Anweisung wird verwendet, wenn abhängig von einer Bedingung unterschiedliche Sequenzen ausgeführt werden sollen.

```
Java:
if (Bedingung) {
    ...
} else {
    ...
}
```



**Mehrfachauswahl**

Mit der Mehrfachauswahl können beliebig viele Fälle untersucht werden. Die zu überprüfende Variable muss als Datentyp byte, short, int, char, String haben.

```
Java:
switch (Variable) {
    case wert1: ... break;
    case wert1: ... break;
    default: ...
}
```



**Feld (Array)**

In einem Feld können mehrere Werte eines Datentyps abgespeichert werden. Deklaration eines Felds:

```
private double[] zahlen;
```

Initialisierung eines Felds mit der Länge 10:

```
zahlen = new double[10];
```

Speichern eines Wertes unter dem Index i:

```
zahlen[i] = 10.3;
```

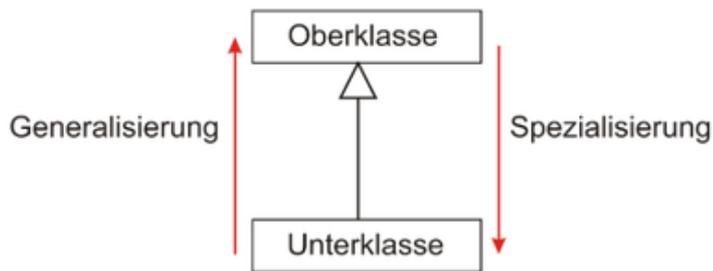
Feld durchlaufen und Inhalt ausgeben:

```
for (int i = 0; i < zahlen.length; i = i+1) {
    System.out.println( (i+1) + ": " + zahlen[i] );
}
```

**Vererbung**

Vererbung beschreibt eine neue Beziehung zwischen einer Oberklasse (Superklasse) und einer Unterklasse (Subklasse)

oder abgeleitete Klasse). Die Unterklasse erbt (erhält) von der Oberklasse alle Attribute und Methoden. Hat die Oberklasse eine Beziehung zu einer anderen Klasse, so hat auch die Unterklasse diese Beziehung.



Umsetzung in Java:

```

public class Unterklasse extends Oberklasse {
...
    public Unterklasse(...) {
        super();
    }
...
}
  
```

Durch den Aufruf `super()`; wird zuerst der Konstruktor der Oberklasse aufgerufen.

### Polymorphie

Man spricht von Polymorphie, wenn gleichnamige Methoden unterschiedliche Definitionen bzw. Implementierungen haben. Polymorphie wird durch Überschreiben der geerbten Methoden bewirkt

Quelle: angelehnt an: <http://www.mgf-kulmbach.de/neu/images/medien/unterrichtsfächer/Informatik/grundwissen/inf10-grundwissen.pdf>

## s Tipps | Informatik

Programmiere ein eigenes Informatik-Projekt - zusätzlich zu den Unterrichts-Projekten in Informatik. Bei Fragen, die du trotz eigenem Ausprobieren bzw. Nachlesen alleine nicht klären kannst, wendest du dich an mich.

Sonnja Genia Riedl