# Begriffe u. Prinzipien der OOP

In der OOP werden die Nachteile   
der bisherigen statischen, prozeduralen Programmierung überwunden.

Wir bemühen wieder einen Vergleich,

an dem wir die Begriffe und Sachverhalte einführen:

Die Weihnachtsbäckerei.

Wenn ich Plätzchen backe, gehe ich dabei wie folgt zweistufig vor:

1. Aus Blech biege ich mir ein Förmchen,  
   mit dem ich später Plätzchen ausstechen möchte,  
   z.B. einen Stern.  
   Dabei kann ich sehr genau jedes Detail definieren:  
   Die Größe, die Anzahl der Zacken, ob er einen Schweif haben soll oder nicht u.s.w.
2. Wenn das Förmchen fertig ist,  
   kann ich damit beliebig viele Plätzchen dieser Form ausstechen,  
   nur begrenzt durch die Menge des Teiges.  
   Alle damit ausgestochenen Plätzchen haben die gleiche Form.

Dies übertragen wir jetzt auf die Programmierung u. führen dabei die wichtigen Begriffe ein:

(abhängige u. unabhängige) Klassen, Objekt u. Instanz:

Nehmen wir als Beispiel wieder die Kundendatenbank.

1. Ich definiere in einer abhängigen Klasse,  
   welche Informationen ich zu Kunden speichern möchte,  
   also z.B. Name, Vorname, Email u.ä.
2. Von dieser abhängigen Klasse erzeuge ich Objekte bzw. Instanzen.  
   Beide Begriffe sind Synonyme.  
   Erst in dem Moment,

indem ich mit dem schon bekannten "Zauberwort" "new" eine Instanz erzeuge,

wird Speicher belegt,

und auch nur für dieses eine Objekt.  
Dies kann zur Laufzeit beliebig oft geschehen,  
ohne eine "künstlich auferlegte Obergrenze" wie die Feldgröße,  
nur begrenzt durch den verfügbaren Arbeitsspeicher.

Wir arbeiten in der OOP mit mindestens zwei Klassen:

1. Der unabhängigen Klasse,   
   die die Hauptmethode enthält  
   u. von daher ausführbar ist.  
   Hier können Instanzen erzeugt werden.
2. Die eine oder auch beliebig viele abhängige Klassen,  
   der "Bauplan",

nach dem Instanzen in der Hauptklasse erzeugt werden können.  
Sie enthalten keine Hauptmethode.

Zur Übersichtlichkeit positionieren wir

* die unabhängige Klasse mit der Hauptmethode links,
* die eine oder auch mehrere abhängigen Klassen ohne Hauptmethode rechts

Die abhängigen Klassen haben nur den Klassenrahmen ohne Hauptmethode.

Die Benennung ist beliebig.

Es macht aber Sinn,

dass schon aus dem Namen von Haupt- u. abhängigen Klassen hervorgeht,

welche Rolle sie spielen,

z.B. durch Aufnahme der Silbe "Haupt" in den Datei- u. Klassennamen.

Es ist nicht notwendig,

die eine oder die vielen abhängigen Klassen einzeln zu compilieren.

Es genügt, die Hauptklasse zu compilieren.

Bezieht sich diese auf abhängige Klassen,

werden diese automatisch mit compiliert.

Sie müssen jedoch darauf achten,

dass Sie mit dem Cursor in der Hauptklasse stehen,

wenn Sie den Build-Befehl absetzen.

Im bisherigen statischen Kontext haben wir von "Variablen" gesprochen.

Wenn wir bisher Variablen auf oberster Ebene definiert haben, auf Klassenebene,

haben wir sie "globale Variablen" genannt.

Hier im dynamischen Kontext nennen wir sie "Attribute".

Attribute einer abhängigen Klasse sind deren Eigenschaften.

Sie sind der erste Aspekt von (abhängigen) Klassen.

Später werden wir in Methoden den zweiten Aspekt kennen lernen.

Wenn wir die bisherige Art zu Programmieren vergleichen mit der neuen,

nennen wir die bisherige prozedurale/imperative Programmierung auch "statisch",

die neue, objektorientierte hingegen "dynamisch".

Sie ist flexibler, wenn auch aufwändiger.

Erstes Beispiel:

Wir wollen eine (abhängige) Klasse "Rechteck" definieren

mit den beiden int-Attributen "laenge" u. "breite".

Von der Hauptklasse aus erzeugen wir ein Rechteck namens "r1"

u. wollen dieses mit Werten belegen.

Wie erzeugen wir ein Objekt oder eine Instanz einer abhängigen Klasse?

Hier im Beispiel wollen wir ein Objekt "r1" der Klasse "Rechteck" erzeugen.

Wir leiten die Anweisung aus der zur Erzeugung von Feldern her,

weil Ähnlichkeit besteht:

Wir wollen ein Feld vom Typ "int" mit dem Namen "meinFeld" der Größe 5 erzeugen:

**int[] meinFeld = new int[5];**

Welche Infos brauchen wir zur Erzeugung eines Rechteckes namens "r1"?

1. Name der (abhängigen) Klasse: "Rechteck"
2. Name der zu erzeugenden Instanz: "r1"

Der Befehl lautet:

**Rechteck r1 = new Rechteck();**

Erzeugung eines Objektes "r1" der Klasse "Rechteck":

**Rechteck r1 = new Rechteck();**

**­­**

Wie sieht es im Speicher aus?

**Rechteck**

int laenge = 0

int breite = 0

r1

r1 ist wie ein Zeiger auf ein Objekt vom Typ "Rechteck"

mit zwei Attributen u. Platz für die zweite Komponente.

Jedes Objekt hat einen eigenen Satz von Attributen.

Wie greife ich schreibend u. lesend darauf zu?

Ähnlich wie bei Feldern u. bei Straßen:

* Bei Straßen u. Häusern erwähnen wir erst den Straßennamen,  
  dann ein trennendes Leerzeichen  
  u. schließlich selektieren wir das Haus durch Nennung der Nummer.
* Bei Feldern erwähnen wir den Feldnamen  
  u. selektieren dann das Element durch Nennung der Indexnummer  
  in eckigen Klammern.
* Hier bei Objekten erwähnen wir zuerst den Objekt- bzw. Instanznamen,  
  schreiben dann einen Punkt u. erwähnen danach das Attribut,  
  auf das ich mich schreibend oder lesend beziehen möchte.

Hier im Beispiel:

Ich setze im Rechteck "r1" das Attribut "laenge" auf 3, "breite" auf 5:

**r1.laenge** = 3;

**r1.breite** = 5;

Lesend genauso:

System.out.println ("Laenge: "+**r1.laenge**);

System.out.println ("Breite: "+**r1.breite**);