

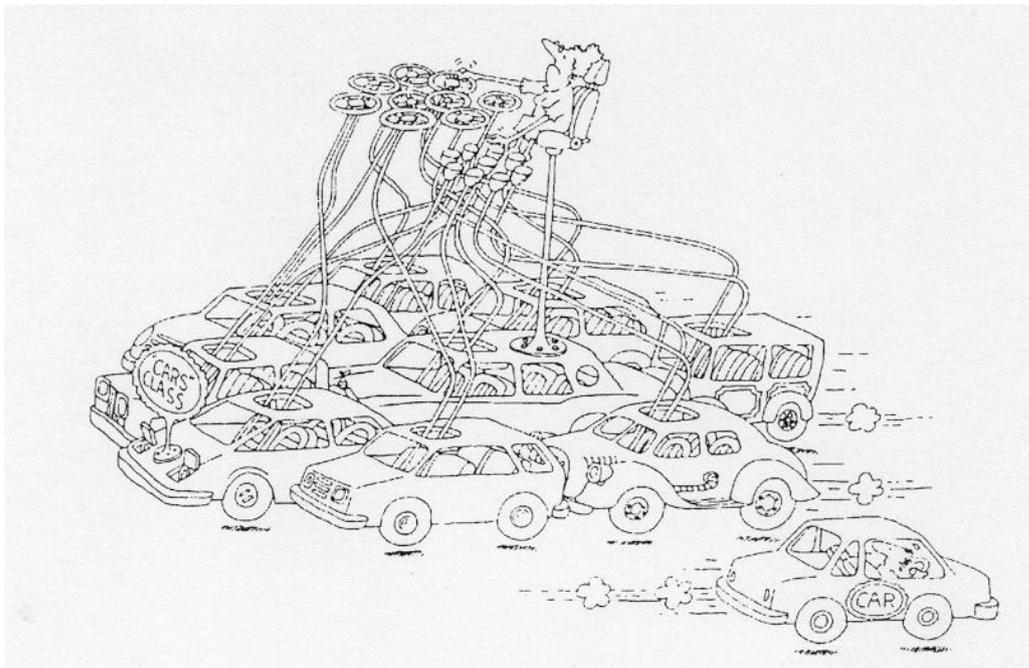
# Klassen

## Objekte und Klassen

### Objekte

Was bedeutet nun Objektorientierung für uns und in Java?

Wir haben es im Alltag fast immer mit sehr komplexen Dingen zu tun. Wer von uns weiss schon genau, wie ein Kühlschrank, eine Fernsehgerät oder ein Auto funktioniert? Um ein Auto fahren zu können, brauchen wir auch gar nicht alles im Detail wissen.



Was wir aber wissen müssen, um Fahren zu können, das sind die wesentlichen Eigenschaften und Funktionen, die sich hinter dem - Objekt Auto als solches - verbergen.

Wir können also ein Auto starten, es hat ein Lenkrad, wir können beschleunigen und bremsen und vieles mehr.

Damit haben wir für uns eine Abstraktion des Objektes Auto hergestellt. Wir kommen damit gut zurecht und können Autofahren. Wir haben dadurch die Komplexität eines Autos für uns in den Griff bekommen. Ein Objekt ist im Zusammenhang mit Objekt-orientierter Programmierung die Realisierung einer Abstraktion in einem Computer

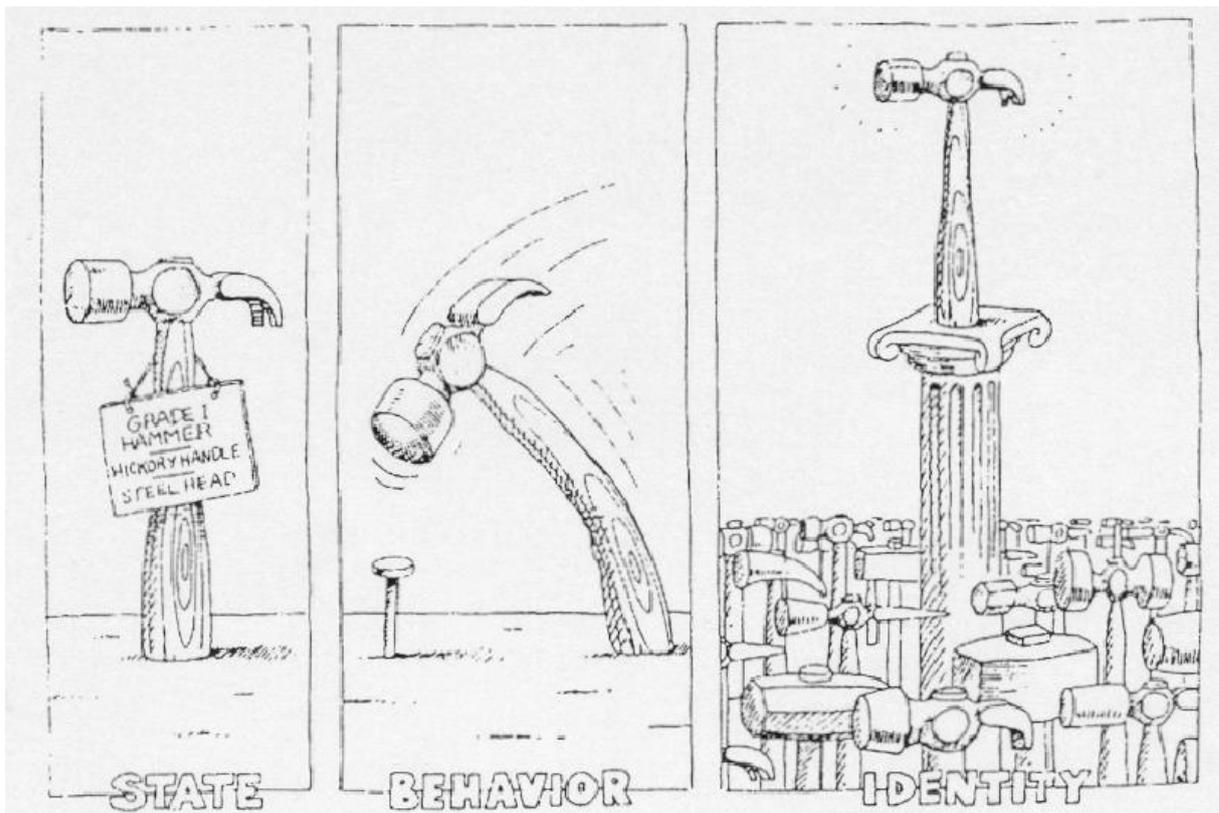
## **Klasse**

Die genannten Eigenschaften hat nun nicht nur mein eigenes Auto, sondern auch das Auto von anderen Leuten. Alle Autos haben diese Eigenschaften. Das gibt uns die Möglichkeit für diese Tatsache einen Begriff zu finden. Dieser Begriff heisst „Klasse“.

Wir können damit sagen, die Klasse der Autos hat diese genannten gemeinsamen Eigenschaften.

Was hat das jetzt mit Programmierung zu tun?

Objekte sind Grundelemente der Wahrnehmung des Menschen. Schon ab etwa dem ersten Lebensjahr kann der Mensch Dinge der Realen Welt durch ihren Zustand ( State ), ihr Verhalten ( Behavior )und ihre Identität ( Identity ) auseinanderhalten.



Genau diese Idee wird übernommen: Die Eigenschaften und das Verhalten von gleichartigen Programm-Objekten wird zusammengefasst.

Das ist die Grundidee der objektorientierten Programmierung.

Beim Programmieren sind die Objekte dann eher aber Zahlen, Zeichenketten, Fenster einer graphischen Benutzeroberfläche, oder aber die Abstraktion einer Bestellung, eines Auftrags und der Komponenten der Lagerhaltung in einem Unternehmen.

Eine Klasse ist dabei eine Art Bauplan für die Eigenschaften und das Verhalten gleichartiger Objekte.

Beispiel: Zum Bauplan eines jeden Autos gehört, dass ein Lenkrad eingebaut wird, dass man es starten kann und dass das Fahrzeug Reifen und Türen hat.

## Die Weltsicht der objektorientierten Programmierung

- Die Welt besteht aus agierenden **Objekten**.
- Die Objekte haben
  - eine **Identität**,
  - **Eigenschaften** (Zustand)
  - und ein **Verhalten**.
- Die Objekte können in **Klassen** eingeteilt werden.

Damit können wir eine Art "Weltsicht" der objektorientierten Programmierung festhalten.

In dieser Welt gibt es Objekte wie Autos oder Fenster in einer graphischen Benutzeroberfläche, die eine Identität haben - mein Auto - , die Eigenschaften haben und die sich irgendwie "verhalten" - mein Auto ist schwarz und hat 4 Türen - die Tür des Autos kann geöffnet werden, das Auto kann starten und es kann beschleunigt werden.

Von meinem Auto lässt sich mühelos erkennen, dass es zur Klasse der Personenkraftfahrzeuge gehört.

Wie funktioniert nun das „objektorientierte Autofahren?“

Dazu legen wir uns ein ganz einfaches Modell eines Autos zurecht.

## Objektorientiertes Autofahren

- Die **Klasse *Auto*** hat die **Eigenschaften *Marke*, *Farbe*, *Eigentümer* und *Geschwindigkeit*.**
- Es gibt **Objekte: *meinAuto*, *schumachersAuto*, ...**
- Es werden die **Eigenschaften festgelegt:**
  - (Fiat, schwarz, JG, 120)
  - (Ferrari, rot, MS, 250)
- So werden aus Objekten ***Instanzen*.**

Wir nehmen an, dass ein Auto durch seine Marke, seine Farbe, seinen Eigentümer und seine Geschwindigkeit allein bestimmt ist und legen eine Klasse Auto an.

Klassen, wie hier die Klasse Auto, sind allgemeine Beschreibungen - gleichsam Schablonen oder generische Darstellungen für "Dinge", für Objekte mit ähnlichen Eigenschaften.

In einer Klasse werden die möglichen Eigenschaften und Verhaltensweisen ihrer Objekte festgelegt. Hier z.B., dass jedes Objekt der Klasse Auto eine Farbe hat.

Die Objekte sind also die realisierten Mitglieder einer Klasse, sie haben Eigenschaften - auch Attribute genannt, die in der objektorientierten Programmierung durch Daten festgelegt werden.

Weiter ist ihr Verhalten festgelegt. In der objektorientierten Programmierung wird dies durch Methoden (d.h. Funktionen) beschrieben. Z.B. kann ein Auto beschleunigen.

Hat man die konkreten Werte für die Eigenschaften eines konkreten Objektes festgelegt, so wird aus einem Objekt eine Instanz der Klasse.

Damit belegt eine Instanz Speicherplatz im Computer für ihre Werte. Diese Eigenschaften können natürlich geändert werden.

Beispielsweise kann ein Auto beschleunigt werden und hat dann eine andere Geschwindigkeit.

Verschiedene Eigenschaften müssen aber auch programmiert werden können. Wir haben es dabei mit den sogenannten Instanz-Variablen zu tun. Zuerst programmieren wir eine Klasse „Auto“ in Java, dann legen wir die Instanzen dieser Klassen fest und fahren dann „objektorientiert“ mit diesen Autos.

Zunächst halten wir fest, dass eine Klasse durch das Schlüsselwort `class` definiert wird.

```
class Auto1{

    String marke;

    String eigentuemer;

    String farbe;

    double geschwindigkeit;

}
```

Dann kommt der Name der Klasse - hier `Auto1` -, der als Konvention in Java immer mit einem Großbuchstaben beginnen sollte.

Beachten Sie bitte, dass die Datei, in der der Programmcode für eine Klasse festgehalten wird, denselben Namen wie die Klasse haben muss und zusätzlich noch die Namenserverweiterung `.java` bekommt.

Syntaktisch ist danach der Klassenrumpf als Block innerhalb geschweifeter Klammern anzulegen.

Dann werden in diesem Rumpf, zunächst durch Anweisungen, die Eigenschaften der Klasse festgelegt.

In unserem Beispiel sind das die Eigenschaften:

Marke, Farbe, Eigentümer als Zeichenketten (`String`) und die aktuelle Geschwindigkeit als Gleitkommazahl (`double`).

Damit können bereits Objekte dieser Klasse `Auto1` angelegt werden und mit Hilfe dieser Instanzvariablen können den Objekten konkrete Werte zugeordnet werden.

```
public class Autofahren1 {
    public static void main(String args[]) {
        Auto1 meinAuto, schumachersAuto;

        meinAuto = new Auto1();
    }
}
```

```

        schumachersAuto = new Auto1();

        meinAuto.eigentuemer="JG";

        meinAuto.farbe="schwarz";

        meinAuto.marke="Fiat";

        ...

        schumachersAuto.marke="Ferrari";

        schumachersAuto.geschwindigkeit
        ="350";

        ...
    }
}

```

Hier wird eine Klasse `Autofahren1` definiert. Da dieses Programm als Applikation ausführbar sein soll, ist in der zweiten Zeile die Angabe der Methode `main()` notwendig!

Dann werden im Programm zunächst die Objekte mit Namen „meinAuto“ und „schumachersAuto“ angelegt. Das passiert in den Zeilen 2 und 3.

Die Eigenschaften werden dann durch Anweisungen unter Benutzung der Instanzvariablen festgelegt.

Beispielsweise geschieht dies für die Instanzvariable `farbe` aus der Klasse `Auto1` für das Objekt `meinAuto`, also hier `meinAuto.farbe = "schwarz"`.

Jetzt haben wir die Grundlagen kennengelernt. Die Eigenschaften dieser Autos wurden festgelegt. Diese Eigenschaften sollen aber auch veränderbar sein. Wenn der Fahrer bremst oder beschleunigt, dann ändert sich doch die Geschwindigkeit. In Java wird dies durch die Bereitstellung von Methoden für die Instanzen einer Klasse bewerkstelligt. Mit diesen Methoden können die Eigenschaften einer Instanz geändert werden.

Als weiteres Beispiel wollen wir nun unsere Autos mit der Möglichkeit zum Beschleunigen und zum Lenken ausstatten.

```
class Auto2{
String marke;
...
int richtung;
public int lenken(int w) {
    richtung = richtung+w;
    return richtung;
}
public double beschleunigen(double a, double t){
    v = v + a*t;
return v;
}
}
```

Die Klasse Auto2 enthält am Anfang ihres Klassenrumpfs die selben Instanzvariablen wie die Klasse Auto1.

Dazu kommt neu noch die Richtung in die das Auto fährt. Diese Richtung fassen wir als ganzzahlige Gradangabe auf.

Weiter werden dann zwei Methoden definiert. Zum einen die Methode lenken, zum anderen die Methode beschleunigen.

Beachten Sie auch, dass ich diesmal die Geschwindigkeit wie in der Physik üblich mit  $v$  bezeichnet habe. Die Methode beschleunigen implementiert dann die aus der Physik bekannte Formel - Anfangsgeschwindigkeit  $v$  plus (konstante) Beschleunigung  $a$  mal Zeit  $t$ .

Die Methoden sind "öffentlich" zugänglich, also werden sie mit `public` vereinbart, sie haben einen Rückgabewert - im einen Fall die ganzzahlige Richtung, im andern Fall die Geschwindigkeit mit einem Wert aus dem Datentyp `double`.

Diese Methoden erwarten Parameter beim Aufruf – eine „ganzzahlige“ Richtungsangabe beim Lenken bzw. eine Beschleunigung und eine Zeitdauer als Gleitkommazahl beim Beschleunigen.

Wie das funktioniert, zeigt uns das folgende Programm

Aus der Datei Autofahren2.java

```

public class Autofahren2 {
public static void main(String args[]) {

    Auto2 schumachersAuto;

    schumachersAuto = new Auto2();

    schumachersAuto.eigentuemer = "MS";

    schumachersAuto.lenken(15);

    schumachersAuto.beschleunigen(10.2, 8.8);

    Auto2 meinAuto;

    meinAuto = new Auto2();

    meinAuto.eigentuemer = "JG";

    meinAuto.lenken(10);

    meinAuto.beschleunigen(4.3, 8.0);

    System.out.println("\n AutoRennen  \n");

    System.out.println(" Das Auto von " + schumachersAuto.eigentuemer + " faehrt mit " + schumachersAuto.geschwindigkeit*3.6 + " km/h " + " in Richtung " + schumachersAuto.richtung + " Grad \n");

    System.out.println(" Das Auto von " + meinAuto.eigentuemer
+

```

```

        " faehrt mit " + meinAuto.geschwindigkeit*3.6 + " km/h "
        +" in Richtung " + meinAuto.richtung + " Grad \n");
    }
}

```

## Übungsaufgaben:

### Aufgabe Motorrad1:

Erstellen Sie analog zu Auto2 eine Klasse Motorrad1 mit selbst gewählten Eigenschaften (mindestens 5) und Methoden (mindestens3).

In einer Klasse MotorradFahren1 sollen mindestens 2 Objekte angelegt werden, denen Sie verschiedene Eigenschaften zuordnen. Wenden Sie alle definierten Methoden an und geben Sie die Eigenschaften der Objekte vorher und nachher aus.

### Aufgabe Yacht1:

Erstellen Sie analog zu Auto2 eine Klasse Yacht1 mit selbst gewählten Eigenschaften (mindestens 5) und Methoden (mindestens3).

In einer Klasse MitYachtSegeln1 sollen mindestens 2 Objekte angelegt werden, denen Sie verschiedene Eigenschaften zuordnen. Wenden Sie alle definierten Methoden an und geben Sie die Eigenschaften der Objekte vorher und nach