

### Objektorientierte Programmierung:

Eine grundlegende Idee der objektorientierten Programmierung (auch OOP genannt) ist es, Daten und die zu diesen Daten gehörende Funktionalität zu einer Einheit zusammenzufassen. Diese Einheiten werden zur Laufzeit des Programms (oder des Applets) als Objekte bezeichnet. Im allgemeinen gelten folgende Eigenschaften objektorientierter Sprachen als Vorteil gegenüber strukturierten Programmiersprachen:

- **Datenkapselung**
- **Vererbung**
- **Polymorphie**

Die Summe der Vorteile von OOP-Sprachen wie Java bietet eine bessere Beherrschbarkeit der Komplexität von Softwareentwicklung und damit die Erstellung qualitativ hochwertiger Softwareprodukte. Die Eigenschaft der Vererbung soll durch die Wiederverwendung vorhandener Klassen eine Verringerung des Arbeitsaufwands mit sich bringen.

### Klassen, die Baupläne von Objekten

Jede Java-Anwendung, egal ob es sich um ein Applet oder eine Applikation handelt, besteht immer aus mindestens einer Klasse. Alle Variablen, Methoden und Anweisungszeilen müssen immer innerhalb einer Klasse stehen. Einzige Ausnahmen bilden die Anweisungen zum Einbinden von Klassen aus Klassenbibliotheken (`import`-Anweisung) sowie Anweisungen, die eine Klasse als zu einer Klassenbibliothek zugehörig kennzeichnen (`package`-Anweisung).

Den Quellcode einer Klasse kann man sich als Bauplan für Objekte vorstellen, die zur Laufzeit der Anwendung für die Ausführung zuständig sind. Eine Klasse besteht also aus Daten und Methoden, die irgendeinen sinngemäßen Zusammenhang haben.

Eine gültige Klassendefinition könnte z.B. folgendermaßen aussehen:

```
/* (Ein Kommentar, der ueber mehrere Zeilen geht)
 * MeineKlasse erweitert die allgemein gehaltene SchuelerKlasse, die wiederum
 * in der Klassenbibliothek schule definiert ist und mit der import-Anweisung
 * sichtbar gemacht wird. Eine Klasse erweitern (engl. extend) bedeutet, dass
 * die neue Klasse alle Eigenschaften (Variablen) und Funktionalität (Methoden)
 * von uebergeordneten Klasse erbt und je nach Bedarf mit eigenen Eigenschaften
 * und Funktionalität erweitert.
 */

import schule.SchuelerKlasse;

public class MeineSchuelerKlasse extends SchuelerKlasse {
    // (Ein Kommentar ueber eine Zeile)
    // Die Klassenvariable, die die Klassenbezeichnung enthaelt:
    public String klassenBezeichnung;
    public int schuelerAnzahl; // enthaelt die Anzahl der Schueler
    // private bedeutet: nur in dieser Klasse sichtbar!
    private String[] schueler; // ein Array fuer alle Schuelernamen

    // Die Konstruktor-Methode ist fuer die Erstellung der SchuelerKlasse
    // verantwortlich:
    public SchuelerKlasse(String klassenBezeichnung, int schuelerAnzahl) {
        // mit dem Schluesselwort this bezieht man sich auf die aktuelle Klasse:
        this.klassenBezeichnung = klassenBezeichnung;
        this.schuelerAnzahl = schuelerAnzahl;
        // jetzt wird das Array fuer die Schuelernamen initialisiert:
        this.schueler = new String[schuelerAnzahl];
    }
}
```

## Java-AG – Lektion 05 – Seite -2-

```
public void setSchuelerName(int nummer, String name) {
    /* Im normalen Leben fangen wir mit 1 an zu zaehlen, beim Array wird das
     * erste Element aber mit 0 angesprochen, also bauen wir uns eine Um-
     * leitung mit nummer - 1:
     */
    schueler[nummer - 1] = name;
}

public String getSchuelerName(int nummer) {
    // das gleich wie oben...
    return schuler[nummer - 1]
}

public void printSchueler(int nummer) {
    // und hier auch so wie oben...
    System.out.println("Schüler Nummer " + nummer + " heißt "
        + this.getSchuelerName(nummer));
}

// Die main-Methode zum Ausfuehren der Applikation:
public static void main(String[] args) {
    // ein Objekt von MeineSchuelerklasse erzeugen: (instancieren)
    MeineSchuelerKlasse klasse10a = new MeineSchuelerKlasse("10a", 4);
    klasse10a.setSchuelerName(1, "Fritz Müller");
    klasse10a.setSchuelerName(2, "Anton Schmidt");
    klasse10a.setSchuelerName(3, "Melanie C.");
    klasse10a.setSchuelerName(4, "Ellie Pirelli");
    // und so weiter...

    // noch ein Objekt von MeineSchuelerklasse erzeugen: (instancieren)
    SchuelerKlasse klasse10b = new SchuelerKlasse("10b", 3);
    klasse10b.setSchuelerName(1, "Nina Hagen");
    klasse10b.setSchuelerName(2, "Udo Lindenberg");
    klasse10b.setSchuelerName(3, "Borris Becker");

    // sämtliche Übereinstimmungen mit noch lebenden Personen sind rein
    // zufällig! =;-))

    // Jetzt zeigen wir die Schüler der Klasse 10a:
    // (mit \n wird eine zusätzliche Leerzeile erzeugt)
    System.out.println("Die Schüler der Klasse 10a:\n");
    for (int i = 0; i < klasse10a.schuelerAnzahl; i++) {
        klasse10a.printSchueler(i + 1);
    }

    // Jetzt zeigen wir noch die Schüler der Klasse 10b:
    System.out.println("Die Schüler der Klasse 10b:\n");
    for (int i = 0; i < klasse10b.schuelerAnzahl; i++) {
        klasse10b.printSchueler(i + 1);
    }
}
}
```