

## Übung 03: Klassen und Objekte

In diesem Übungsblatt tasten wir uns an die objektorientierte Programmierung heran. Sie sollen für einen Kunden eine (sehr einfache) Software entwickeln, mit der sich die Angestellten (engl. *Employee*) einer Firma und deren Löhne verwalten lassen. In diesem Rahmen wird die Klasse `Employee` entworfen und programmiert:

1. Jeder Angestellte hat einen *Namen* und ein *Monatsgehalt*. Für das Monatsgehalt soll zwingend der Datentyp `float` verwendet werden.
2. Jedem Objekt der Klasse `Employee` muss ab dem Zeitpunkt der Initialisierung ein Name zugewiesen sein. Optional kann mit der Initialisierung auch gleich das Monatsgehalt festgelegt werden. Überlegen Sie, wie man das erreichen kann!
3. Die Klasse `Employee` verfügt über eine Methode, mit der man das Gehalt auf einen Wert setzen kann. Ein Gehalt von mehr als 50000 Euro wird abgelehnt und das Gehalt bleibt dann unverändert. Das Gehalt darf ausschließlich über eine Methode modifiziert werden.
4. Die Klasse `Employee` verfügt über Methoden, um das Gehalt und den Namen auszulesen (Getter).
5. Die Klasse hat eine Methode, die eine Zeichenkette mit dem Namen und dem Gehalt zurückliefert.

### Aufgabe 1: Implementierung der Klasse `Employee`

Legen Sie ein neues IntelliJ-Projekt an und implementieren Sie die Klasse `Employee.java` gemäß den oben beschriebenen Vorgaben!

Versuchen Sie bereits die Attribute und Methoden mit den korrekten Sichtbarkeiten `private` und `public` zu ergänzen.

Sogenannte Getter-Methoden (Punkt 4) können Sie in IntelliJ mit „Code-Generate“ automatisch erzeugen lassen.

### Aufgabe 2: Implementierung der Testanwendung

Eine kleine Anwendung testet die Implementierung Ihrer Klasse.

- a) Erstellen Sie in Ihrem IntelliJ Projekt eine weitere Klasse `TestEmployee` mit einer `main`-Methode. Im Folgenden wird die `main`-Methode schrittweise um Anweisungen erweitert.
- b) Legen Sie zunächst 3 Objekte der Klasse `Employee` an. Das Gehalt soll in einem Fall noch nicht gesetzt werden.
- c) Tragen Sie alle 3 Objekte in ein neues Array ein.
- d) Fragen sie den Benutzer in der Konsole nach dem Gehalt des Angestellten, dessen Gehalt in b) noch nicht gesetzt wurde. Setzen Sie es über die dafür vorgesehene Methode.
- e) Berechnen Sie nun das Durchschnittsgehalt aller 3 Angestellten, indem Sie das Array durchlaufen. Das Ergebnis soll mit der Funktion `Math.round()` gerundet und dann ausgegeben werden.

(umblättern)

### Aufgabe 3: Die Klasse Random

Schreiben Sie ein kleines Programm TestRandom, das einen Münzwurf simuliert. Die Münze soll 100000mal geworfen werden und dann ausgegeben werden wie oft man „Kopf“ bzw. „Zahl“ als Ergebnis bekommt. Leider handelt es sich um eine „unfaire“ Münze und die Wahrscheinlichkeit für „Kopf“ ist 75% statt 50%.

Zur Erzeugung von Zufallszahlen wurde auf dem letzten Übungsblatt die Methode `Math.random()` verwendet. Dieses Mal sollen Sie „objektorientiert“ ein Objekt der Klasse `Random` des Java Frameworks erzeugen und damit eine Pseudozufallszahlenfolge generieren. Die API der Klasse `Random` des Java Frameworks ist hier dokumentiert:

<https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>

*Hinweis:* Das Paket `Random` muss man mit der Anweisung `java.util.Random` importieren. In IntelliJ geht das automatisch durch `Alt+Enter`, wenn der Cursor auf dem Klassennamen steht.