

# Abstrakte Klassen / Interfaces

(25 Punkte)

Eine Mitarbeiterdatenbank enthält verschiedene Arten von Mitarbeitern: Hilfsarbeiter, Angestellte, Verkäufer usw.

Die verschiedenen Arten von Mitarbeitern haben neben ihrer Verschiedenheit auch Gemeinsamkeiten. Als Basis für die Mitarbeiterdatenbank braucht es zunächst Klassen.

## Aufgabe:

- a) (14 Punkte) Schreiben Sie folgende Klassen, um einen **Hilfsarbeiter** "darzustellen":
- eine abstrakte Klasse `Mitarbeiter`, die als Basis für alle Arten von Mitarbeitern verwendet werden kann.
  - eine Klasse `Hilfsarbeiter`, die von der Klasse `Mitarbeiter` abgeleitet wird.
- b) (11 Punkte) In dieser Aufgabe entwickeln Sie die Klasse **Verkäufer**, diesmal aber mittels Verwendung eines Interfaces. Entwickeln Sie dabei den Code von Grund auf, ohne Verwendung der Klassen aus Aufgabe a). Schreiben Sie dazu folgendes:
- ein Interface `MitarbeiterInterface`, welches für alle Arten von Mitarbeitern verwendet werden kann.
  - eine Klasse `Verkäufer`, die das Interface `MitarbeiterInterface` implementiert.

## Spezifikation:

Allen Mitarbeitern gemeinsam sind folgende Attribute:

- `personalNr` (int)
- `name` (String)
- `datumEintritt` (String)

Allen Mitarbeitern gemeinsam sind folgende Methoden:

- `getName`
- `getPersonalNr`
- `getMonatsLohn`

Beim Hilfsarbeiter wird der MonatsLohn wie folgt berechnet:

$\text{stundenlohn} * \text{anzahlstunden} + \text{ueberstundenzuschlag} * \text{anzahlueberstunden}$ ;

Beim Verkäufer wird der MonatsLohn wie folgt berechnet:

$\text{fixgehalt} + \text{umsatz} * \text{provision} / 100$ ;

## Anmerkungen:

- Überlegen Sie, welche Methoden in der Basisklasse bereits implementiert werden können und welche erst in der abgeleiteten Klasse.
- Die benötigten Variablen und Methoden ergeben sich aus obiger Spezifikation.