

Aufgabe 02: Vererbung / *abstract* / *ArrayList*

Zur Lösung der nachfolgenden Aufgabe empfehle ich sich zusätzlich zur Theoriestunde auch im Eigenstudium noch das Thema Vererbung, Überschreiben der Methode *equals()* und die Klasse *ArrayList* anzusehen:

Kapitel 8.1 und 9 aus: www.javabuch.de

ev. auch: [http://de.wikipedia.org/wiki/Vererbung_\(Programmierung\)](http://de.wikipedia.org/wiki/Vererbung_(Programmierung))

sowie die JAVA-API-Dokumentation zur *ArrayList*

1. Teil: Zu erstellen ist folgende Klassenhierarchie (- steht für Attribut, + kennzeichnet eine Methode):

Mitarbeiter

- *name*
- *sozVersNr* ein einfacher *String* genügt
+ *berechneGehalt()* : float kann hier noch nicht berechnet werden, die Methode soll aber existieren
+ *toString()* : String soll auch die Ausgabe des Gehaltes beinhalten
+ *printlnInfo()*

Angestellter (ist ein *Mitarbeiter*)

- *gehalt* : float vereinbartes monatliches Grundgehalt
+ *berechneGehalt()* : float gibt das Grundgehalt zurück
+ ...

TeilzeitAngestellter (ist ein *Angestellter*)

- *stundenProWoche* : float zu arbeitende Wochenstundenanzahl (soll ≥ 10 und ≤ 30 sein)
+ *berechneGehalt()* : float berechnet sich aus dem Angestelltengehalt * *stundenProWoche* / 38.5
+ ...

Abteilungsleiter (ist ein *Angestellter*)

- *anzahlMitarbeiter* Anzahl der Mitarbeiter der geleiteten Abteilung
+ *berechneGehalt()* : float berechnet sich aus dem Gehalt eines *Angestellten* + *anzahlMitarbeiter* * 100
+ ...

WerksvertragsAngestellter (ist ein *Mitarbeiter*)

- *stunden* : float pro Monat zu arbeitende Stunden
- *stundenlohn* : float
+ *berechneGehalt()* : float berechnet sich aus *stunden* * *stundenlohn*
+ ...

Für diese Klassen gilt:

- **Konstruktoren:** Der Konstruktor einer abgeleiteten Klasse ruft zur Initialisierung der geerbten Attribute den Konstruktor der Elternklasse auf.
- **Übergabewerte** sind auf **Plausibilität zu prüfen**, Bei einem ungültigen Wert ist, sofern kein Default-Wert sinnvoll erscheint, eine **Exception** zu werfen.
- **Bitte selbst überlegen:** welche Methoden wo am besten zu implementieren sind und ob bzw. wo sie „*abstract*“ deklariert werden müssen. *set...()*- und *get...()*-Methoden sind nur soweit erforderlich zu realisieren.
- ***toString()*:** diese Methode gibt die Eigenschaften des betreffenden Objektes als String zurück. Eine abgeleitete Klasse soll zum Aufbau des Rückgabestrings die *toString()*-Methode der Elternklasse verwenden.
- ***printlnInfo()*:** gibt den mittels *toString()* erzeugten Text zu Testzwecken auf die Konsole aus.

2. Teil: Zusätzlich ist eine Klasse Firma zu schreiben:

Firma (ist eine Klasse zur Verwaltung der Mitarbeiter)

- *name* : String (Name der Firma)

- *mitarbeiterliste* : Liste der Mitarbeiter (ArrayList, generisch, Typ: *Mitarbeiter*)

- ...

+ ***einstellen(neuerMitarbeiter : Mitarbeiter)***, fügt einen neuen Mitarbeiter hinzu. Dabei ist sicherzustellen, dass kein Mitarbeiterobjekt mehrmals eingefügt wird. Die *add()*-Methode der *ArrayList* prüft das nicht automatisch, aber es gibt eine genau dafür geeignete Methode – bitte in der JavaDoc nachsehen!

+ ***entlassen(szvnr : String) : boolean***, entfernt den Mitarbeiter mit der angegebenen Sozialversicherungsnummer aus der Firma. Rückgabewert: *true*...falls gefunden, *false*...wenn es keinen Mitarbeiter mit der angegebenen Nummer gibt

+ ***printMitarbeiter(gehalt : float)***, gibt mit Hilfe der *printlnInfo()*-Methoden eine Liste aller Angestellten aus, deren Gehalt größer-gleich dem angegebenen Wert ist. Bei 0 werden alle Mitarbeiter ausgegeben.

+ ***sucheMitarbeiter(name : String) : Mitarbeiter***, diese Methode sucht den Mitarbeiter mit dem angegebenen Namen und gibt das erste gefundene Objekt zurück.

Wie sieht der Rückgabewert sinnvollerweise aus, wenn kein passender Angestellter gefunden wird?

Bitte nicht vergessen: Testklasse / Testmethode(n) in der zumindest 5 Mitarbeiter „angestellt“ werden und alle Methoden getestet werden – inklusive Fehlerfälle!