

Übung 3 zur OOS SS16 Bearbeitung bis 07.04.2016**Aufgabe 11: (Implementierung einer Klasse)**

Gegeben sei folgende Klassenbeschreibung in UML-Notation (siehe hierzu Übungs- Aufgabe 6):

Point
x:int y:int
Point() Point(p:Point) Point(x:int,y:int) getLocation():Point setLocation(p:Point):void setLocation(x:int,y:int):void move(dx:int,dy:int):void equals(p:Point):boolean toString():String

- a) Programmieren Sie eine zu den Beschreibungen in Übungs-Aufgabe 6 analoge Klasse in Java. Programmieren Sie dabei die Methoden, wie sie in Übungs-Aufgabe 6 informell spezifiziert wurden.
- b) Erzeugen Sie Objekte dieser Klasse, indem jeder Konstruktor genau einmal aufgerufen wird und geben Sie jeweils die erzeugten Objekte aus.
- c) Initialisieren Sie die Objekte aus b) mit sinnvollen Werten.
- d) Geben Sie die Werte der Attribute dieser Objekte aus.
- e) Rufen Sie alle Methoden eines der von Ihnen erzeugten Objekte auf.
- f) Testen Sie Ihre Implementierungen. Geben Sie jeweils sinnvolle Testfälle an.

Aufgabe 12: (Vererbung & Aggregation)

- a) Implementieren Sie die Klasse `Quadrat` aus Übungs-Aufgabe 9 in Java durch Vererbung von der Klasse `Point`.
- b) Implementieren Sie die Klasse `Quadrat` aus Übungs-Aufgabe 9 in Java, indem Sie sie diesmal nicht durch Vererbung von der Klasse `Point` ableiten, sondern durch Aggregation unter Verwendung der Klasse `Point` implementieren.
- c) Zeichnen Sie zu Ihrer Lösung aus b) ein UML-Klassendiagramm.

Übung 3 zur OOS SS16 Bearbeitung bis 07.04.2016**Aufgabe 13: (Polymorphismus und dynamisches Binden)**

Erweitern Sie die Vererbungshierarchie mit der Klasse `Mitarbeiter` als Wurzelklasse um eine Klasse `Arbeiter`, die in UML wie folgt angegeben ist und die in der Vererbungshierarchie auf gleicher Ebene wie die Klassen `Angestellter` und `Manager` stehen soll.

Arbeiter
stundenLohn: double anzahlStunden: double ueberStundenZuschlag: double anzahlUeberStunden: double schichtZulage: double
Arbeiter() Arbeiter(stundenLohn: double, anzahlStunden: double, ueberStundenZuschlag: double anzahlUeberStunden: double, schichtZulage: double) monatsBrutto(): double equals(arbeiter: Arbeiter): boolean toString():String

- a) Geben Sie die Vererbungshierarchie in UML an, die sich ergibt, wenn Sie die Klasse hinzufügen.
- b) Definieren Sie in den Klassen `Angestellter` und `Manager` einen Konstruktor, der alle zur Berechnung der Methode `monatsBrutto` benötigten Attribute initialisiert, und die Methoden `equals` und `toString`.
- c) Implementieren Sie die Klasse `Arbeiter` in Java, wobei die Methode `monatsBrutto` sich wie folgt berechnet:


```
stundenLohn * anzahlStunden
+ ueberStundenZuschlag*anzahlUeberStunden
+ schichtZulage
```
- d) Erzeugen Sie in einem `main`-Programm ein Feld, das einen `Manager`, zwei `Angestellte` und zwei `Arbeiter` enthält und berechnen Sie das Gesamtbrutto dieser fünf Mitarbeiter.