

Aufgabe 1:

In einer Universitäts-Software gibt es die Klassen: `Person`, `Dozent`, `Student` und `Promotionsstudent`.

Attribute: `vorname`, `nachname`, `matrikelNr`, `doktorVater`

Methoden: Getter und Setter für alle Attribute, die Methode `ausgeben()` gibt alle Eigenschaften eines Objektes aus.

- Um Redundanzen im Code zu vermeiden, soll das Konzept der Vererbung verwendet werden. Zeichnen Sie ein UML – Diagramm der Vererbungshierarchie. Ordnen Sie die genannten Attribute und Methoden den Klassen zu.
- Welche der folgenden Code-Zeilen erzeugen keinen Fehler? Begründen Sie Ihre Antworten.

```
01      Person p1 = new Person();
02      Person p2 = new Promotionsstudent();
03      Promotionsstudent prol = new Student();
04      Dozent d1 = new Person();
05      Student s1 = new Promotionsstudent();
```

- Betrachten Sie folgende Deklarationen und Zuweisungen. Welche Code-Zeilen ab Zeile 07 sind zulässig und erzeugen keinen Fehler? Begründen Sie Ihre Antworten.

```
01      Person p1 = new Person();
02      Person p2 = new Person();
03      Promotionsstudent prol = new Promotionsstudent();
04      Dozent d1 = new Dozent();
05      Student s1 = new Student();
06
07      s1 = p1;
08      p1 = s1;
09      d1 = s1;
10      s1 = prol;
11      prol = s1;
```

Aufgabe 2

Implementieren Sie die Klassen-Hierarchie aus Aufgabe 1 und überprüfen Sie, ob Ihre Antworten stimmen.

Aufgabe 3

Auf der Folgeseite finden Sie einen JAVA – Code-Ausschnitt.

- Wo startet der Code?
- Welche Aufgabe hat das Schlüsselwort `super`?
- Was wird durch `super (...)` aufgerufen?
- Welche Aufgaben hat das Schlüsselwort `this`?
- Was wird durch `this (...)` aufgerufen?
- Was wird auf der Konsole exakt ausgegeben?
- Wie viele Methoden-Überladungen gibt es?
- Wie viele Methoden-Überschreibungen gibt es?

Code zu Aufgabe 3

```

01 public class Tier
02 {
03     protected String name;
04
05     public Tier(String name){
06         this.name = name;
07         System.out.println("Tier " + this.name + " erzeugt");
08     }
09
10     public String toString(){
11         return "Animal with name " + this.name;
12     }
13 }
14
15 public class Insekt extends Tier
16 {
17     public Insekt(String name){
18         super(name) ;
19         System.out.println(super.toString());
20     }
21
22     public String toString(){
23         return "Insekt " + name;
24     }
25 }
26
27 public class Biene extends Insekt
28 {
29     public Biene(){
30         this("Maja");
31         System.out.println("Insekt");
32         System.out.println(this.toString());
33     }
34
35     public Biene(String s){
36         super(s) ;
37         System.out.println(s);
38     }
39
40     public static void main(String args[]){
41         new Biene();
42     }
43 }
44

```

Aufgabe 4

Implementieren Sie den Code aus Aufgabe 3 und starten Sie das Programm. Überprüfen Sie, ob die tatsächliche Ausgabe auf der Konsole Ihren Erwartungen entspricht.

Hinweis: Bitte den Code nicht in eine einzige Klasse kopieren. Erstellen Sie für jede Klasse eine Datei.