

Übung zur Vorlesung *Einführung in die Informatik 2 für Ingenieure (MSE)*

Alexander van Renen (renen@in.tum.de)

<http://db.in.tum.de/teaching/ss16/ei2/>

Blatt Nr. 2

Dieses Blatt wird am Montag, den 25. April 2016 besprochen.

Aufgabe 1: Typisierung

Typisieren Sie die nachfolgenden Pfadausdrücke wie in dem Beispiel in Abbildung 1, das in der Vorlesung besprochen wurde. Geben Sie zusätzlich an, welchen Wert der gesamte Ausdruck hat.

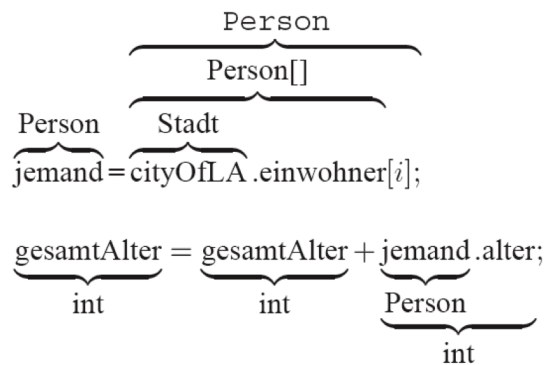


Abbildung 1: Typisierung von Pfadausdrücken

Achten Sie auf Konsistenz mit der Beispielimplementierung des Universitätsmodells.¹ Die Objekte seien wie folgt instanziiert:

```
1 Professor sokrates = new Professor(2125, "Sokrates",  
2 Professor.Rang.C4);  
3 Vorlesung ethik = new Vorlesung(5041, "Ethik", 4, sokrates);  
4 Assistent aristoteles = new Assistent(3003, "Aristoteles",  
5 "Syllogistik", sokrates);  
6 Student fichte = new Student(26120, "Fichte", 10);  
7 Calendar termin = Calendar.getInstance();  
8 Pruefung pruefung = new Pruefung(fichte, ethik, sokrates, termin);  
9 int netto, jahre, nr, stunden;
```

a) `fichte.name`

b) `netto = sokrates.gehalt() - aristoteles.boss.steuern()`

c) `jahre = pruefung.pruefling.semester / 2`

¹Die Implementierung finden Sie auf der Vorlesungswebseite: <http://www-db.in.tum.de/teaching/ss16/ei2/>

- d) `nr = ethik.dozent.persNr`
- e) `aristoteles.boss.rang`
- f) `stunden = pruefung.pruefungsstoff.dozent.lehrstunden`

Aufgabe 2: Werte vs. Objekte

- a) Nennen Sie mindestens drei Unterschiede zwischen Werten und Objekten.
- b) Wann verwendet man Komposition und wann Aggregation?
- c) Was ist der Unterschied zwischen Klassenattributen und Objektattributen?
- d) Initialisieren Sie ein Array auf zwei verschiedene Arten mit den Zahlen 2, 3, 5, 7, 11, 13, 17, 19 und 42. Welche würden Sie bevorzugen?

Aufgabe 3: Assoziationen in UML

Beschriften Sie die Assoziationen und Aggregationen im Klassendiagramm in Abbildung 2 (letzte Seite) mit den korrekten Funktionalitäten. Dabei sollte ein Student bis zu fünf Bücher ausleihen können, ein Mitarbeiter unbegrenzt viele.

Aufgabe 4: Java Garbage Collection

Der folgende Java-Programmcode generiert mehrere Objekte. Zwei der erzeugten Objekte werden am Ende des Programms nicht mehr referenziert und können daher von der *Garbage Collection* bereinigt werden. Welche sind dies? Geben Sie auch jeweils die Zeile an, ab der das Objekt nicht mehr referenziert wird.

```

1 Assistent wittgenstein = new Assistent(3004, "Wittgenstein",
2                                     "Sprachtheorie", null);
3 wittgenstein.boss = new Professor(2137, "Kant", Professor.Rang.C4);
4 Vorlesung ethik = new Vorlesung(5043, "Ethik", 3, wittgenstein.boss);
5 ethik.dozent = new Professor(2126, "Russel", Professor.Rang.C4);
6 wittgenstein.boss = new Professor(2133, "Popper", Professor.Rang.C3);
7 Student jonas = new Student(25403, "Jonas", 12);
8 Pruefung pruefung = new Pruefung(jonas, ethik,
9                                 wittgenstein.boss, termin);
10 pruefung.student = new Student(28106, "Carnap", 3);
11 pruefung.student = jonas;
12 wittgenstein.boss = new Professor(2136, "Curie", Professor.Rang.C4);

```

Aufgabe 5: Java

Laden Sie die Implementierung des Universitätsmodells von der Vorlesungswebseite herunter.² Kompilieren Sie die Dateien und führen Sie *Test.java* aus.

²Wie war die URL doch gleich? Ach genau: <http://www-db.in.tum.de/teaching/ss16/ei2/>

- a) Wir haben in Aufgabe 1 das Nettogehalt eines Professors bestimmt. Wir wollen dies nun als Methode implementieren. Zu welcher Klasse sollte diese am besten hinzugefügt werden? Implementieren Sie die Methode in dieser Klasse. Testen Sie die Methode anschließend in *Test.java*.
- b) Erweitern Sie die Klasse *Student.java* um eine Methode, die alle Vorlesungen zurückgibt, bei denen der Student durchgefallen ist. Fügen Sie zu Testzwecken in *Test.java* einen Aufruf der Methode hinzu (und noch ein paar weitere Prüfungen). Überzeugen Sie sich, dass ihre Implementierung funktioniert.

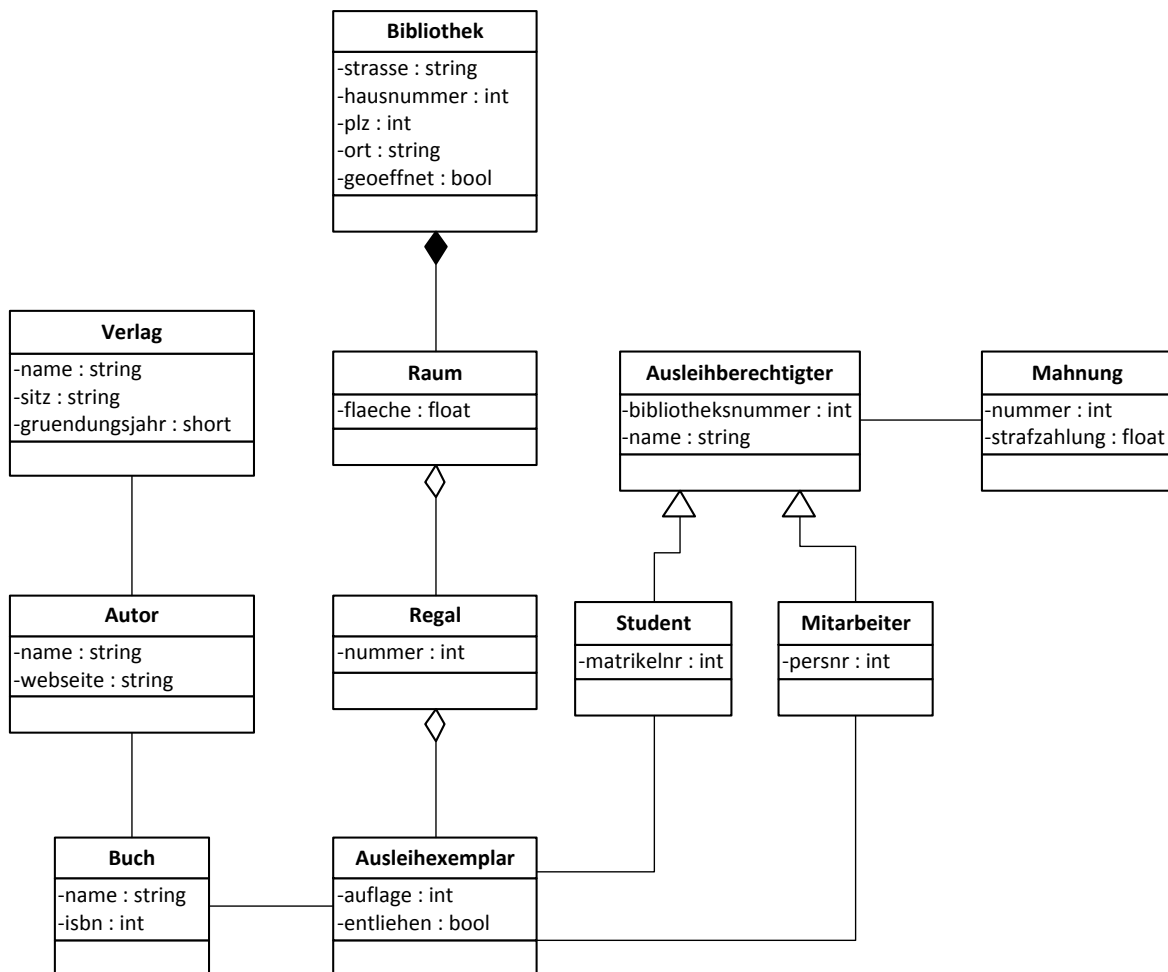


Abbildung 2: Modellierung einer Bibliothek