

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

Simon Ellmann (ellmann@in.tum.de), Florian Drescher (florian.drescher@tum.de)

<http://db.in.tum.de/teaching/ss25/ei2/>

Blatt Nr. 2

Dieses Blatt wird am Dienstag, den 13.05.2025 besprochen.

Zugehörige Vorlesungsaufzeichnung: [Vorlesung 3](#)

Aufgabe 1: Typisierung

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

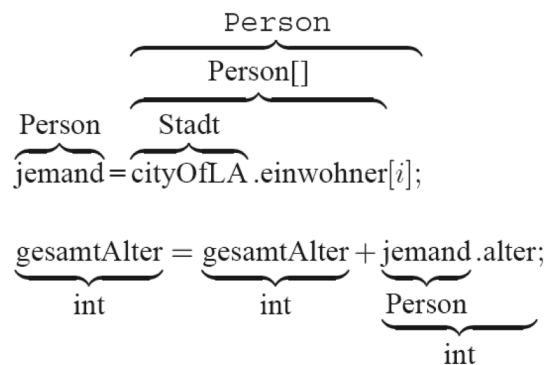


Abbildung 1: Typisierung von Pfadausdrücken

Achten Sie auf Konsistenz mit der Beispielimplementierung des Universitätsmodells. 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`

- 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

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

- 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.

Aufgabe 5: Bonus: UML Model

[[Dies ist ein Beispiel wie eine Aufgabe in der man ein UML Diagramm zeichnen soll in einer Klausur aussehen könnte]]

Erstellen Sie anhand der folgenden Anforderungen ein UML Klassendiagramm mit Multiplizitäten für einen online Streaming Service:

- Es gibt Filme. Diese haben einen Titel und ein Erscheinungsdatum.
- Ein Film hat beliebig viele Sequels, aber nur maximal ein Prequel.

- Schauspieler haben einen Namen und ein Geschlecht.
- Schauspieler spielen in Filmen mit.
- Jeder Film hat zwischen 1 und 5 Genres.
- Genres haben einen Namen.

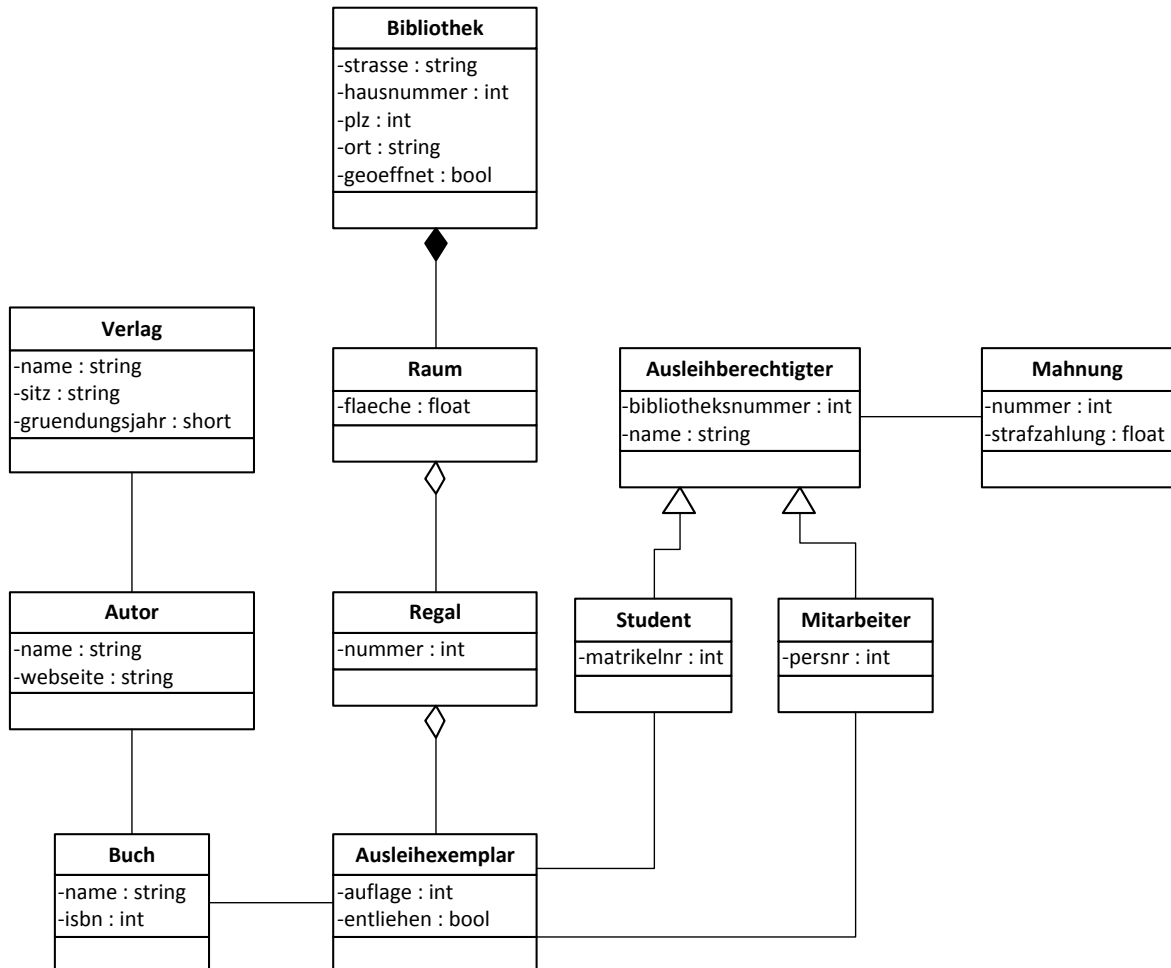


Abbildung 2: Modellierung einer Bibliothek