

Übungen zur Vorlesung Softwaretechnologie

- Wintersemester 2019/20 -
Dr. Günter Kniesel

Übungsblatt 4

Zu bearbeiten bis: 01.11.2019, 16 Uhr

Bitte fangen Sie **frühzeitig** mit der Bearbeitung an, damit wir Ihnen bei Bedarf helfen können. Checken Sie die Lösungen zu den Aufgaben bitte in Ihr Repository ein, „Erklärungen“ bitte als Textdatei. Fragen zu Übungsaufgaben respektive zur Vorlesung können Sie auf der Mailingliste swt-tutoren@lists.iai.uni-bonn.de, bzw. swt-vorlesung@lists.iai.uni-bonn.de stellen.

Praktische Aufgaben

Aufgabe 1. Klassendiagramm

18,5 Punkte

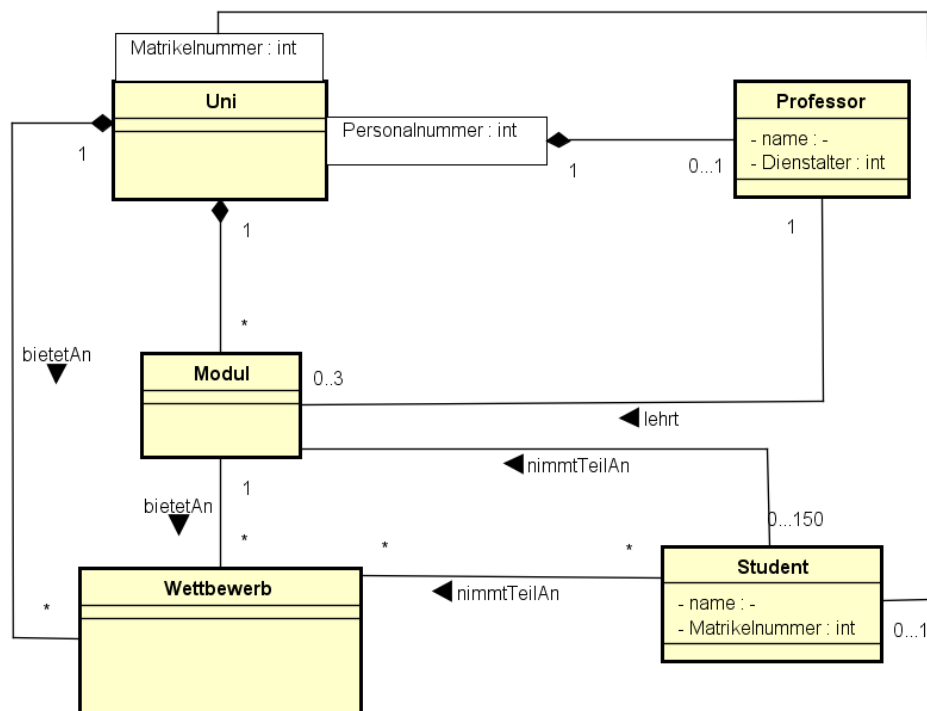
Modellieren Sie nachfolgenden Sachverhalt als Klassendiagramm. Lassen Sie dabei Details weg, die nicht in einem Klassendiagramm modellierbar sind:

- a. Das Institut für Informatik (IfI) gliedert sich in verschiedene Abteilungen, es muss allerdings mindestens eine Abteilung vorhanden sein, damit das IfI bestehen kann.
- b. Jede dieser Abteilungen unterteilt sich wiederum in Arbeitsgruppen.
- c. Jede Abteilung und jede Arbeitsgruppe hat einen Namen und einen Leiter, der ein Professor ist.
- d. Zum IfI gehört außerdem der Prüfungsausschuss.
- e. Der Prüfungsausschuss wird von einem Vorsitzenden geleitet, der ein Professor am IfI ist.
- f. Studenten haben einen Namen, eine Matrikelnummer und ein Fachsemester.
- g. Alle Studenten sind dem Institut über ihre Matrikelnummer bekannt. Jede Matrikelnummer ist dabei höchstens einem Studenten zugeordnet.
- h. Auch Studenten können Teil des Prüfungsausschusses sein.
- i. Mitarbeiter haben einen Namen, eine Personalnummer und ein Dienstalter.
- j. Alle Mitarbeiter sind dem Institut über ihre Personalnummer bekannt. Jede Personalnummer ist dabei höchstens einem Mitarbeiter zugeordnet.
- k. Alle Mitarbeiter arbeiten bei einer Arbeitsgruppe
- l. Das Institut für Informatik liegt in der Endenicher Allee 19a

Aufgabe 2. OCL-Constraints und Kompositionsstrukturdiagramm

11,5 Punkte

- a) **(2 Punkte)** Modellieren Sie als Ergänzung zu Aufgabe 1, dass nur Professoren, die eine Arbeitsgruppe in einer Abteilung haben, diese Abteilung leiten können. Setzen Sie diese Bedingung in OCL als Constraint um. Gegebenenfalls ergänzen Sie Ihre Lösung zu Aufgabe 1 um explizite Namen für Rollen oder für Instanzvariablen, die für die Formulierung des Constraints erforderlich sind.
- b) **(3 Punkte)** Betrachten Sie das folgende Klassendiagramm. Drücken Sie mit Hilfe eines Kompositionsstrukturdiagrammes aus, dass Module und dazugehörige Wettbewerbe stets zur selben Uni gehören müssen.



- c) **(3,5 Punkte)** Können Sie Ihr Kompositionsstrukturdiagramm so erweitern, dass es auch ausdrückt, dass Studenten nur an Wettbewerben teilnehmen können, die von der Uni angeboten werden, an der sie Module belegen?
- d) **(1 Punkt)** Geben Sie eine legale Instanziierung Ihres Kompositionsstrukturdiagramms an, also ein Kompositionsobjektdiagramm.

Aufgabe 3. Sequenzdiagramm

15 Punkte

In dieser Aufgabe sollen Sie den Ablauf einer Klausurkorrektur modellieren. Als Parameter werden übergeben:

- Der Stapel Klausuren, den wir hier der Einfachheit halber als ein Objekt zusammenfassen das intern die zugehörigen Klausuren verwaltet

- Das Prüfungsamt
- Das System, in welchem die Noten verwaltet werden

Der Ablauf soll nun folgendes umfassen:

1. Zunächst wird jede der n Klausuren im Stapel vom Prüfer korrigiert.
2. Danach liest der Prüfer von jeder Klausur die Note ab und trägt sie in BASIS ein.
3. Nach Eintragung der Noten findet eine Einsicht statt. Nehmen Sie vereinfachend an, der Professor hat eine Methode `klausureinsicht():boolean`.
4. Falls sich bei der Einsicht bei mindestens einer Klausur eine Änderung ergeben hat, müssen Noten aktualisiert werden. Dies kann er jedoch in einem Schritt durchführen
5. Zum Schluss bringt er den Stapel zum Prüfungsamt. In unserem Kontext existiert der Stapel ab diesem Zeitpunkt nicht mehr

Hinweis: Sie dürfen

- in Schleifen die Schleifenvariable i nutzen
- davon ausgehen, dass die i -te Klausur auch dem i -ten Eintrag in Basis entspricht

Notieren Sie den Datenfluss so explizit und kompakt wie möglich.

Theoretische Aufgaben

Aufgabe 4. *UML-Elemente in Java*

8 Punkte

Ordnen Sie den folgenden UML-Elementen die entsprechenden Code-Umsetzungen in Java zu. Geben Sie jeweils ein minimales Beispiel-Diagramm und entsprechendes Codestück an.

- a) **(1 Punkt)** Generalisierung und Implementierung
- b) **(1 Punkte)** Aktive Klasse bzw. aktives Objekt
- c) **(1 Punkt)** 1:1 und 1:n Assoziation
- d) **(3 Punkte)** Bidirektionale Assoziation
- e) **(2 Punkte)** Paketdiagramm mit Zugriff auf Klassen eines anderen Pakets

Aufgabe 5. *Frequently asked questions*

6 Punkte

Beantworten Sie die folgenden Fragen möglichst präzise:

- a) **(2 Punkte)** Was ist der Unterschied zwischen Knoten eines Zustandsdiagramms und Knoten eines Aktivitätsdiagramms?
- b) **(2 Punkte)** Was steht an den Kanten eines Aktivitätsdiagramms?
- c) **(2 Punkte)** Welche Komponenten stehen an den Kanten eines Zustandsdiagramms?

Σ 59 Punkte