

Übungen zur Vorlesung Softwaretechnologie

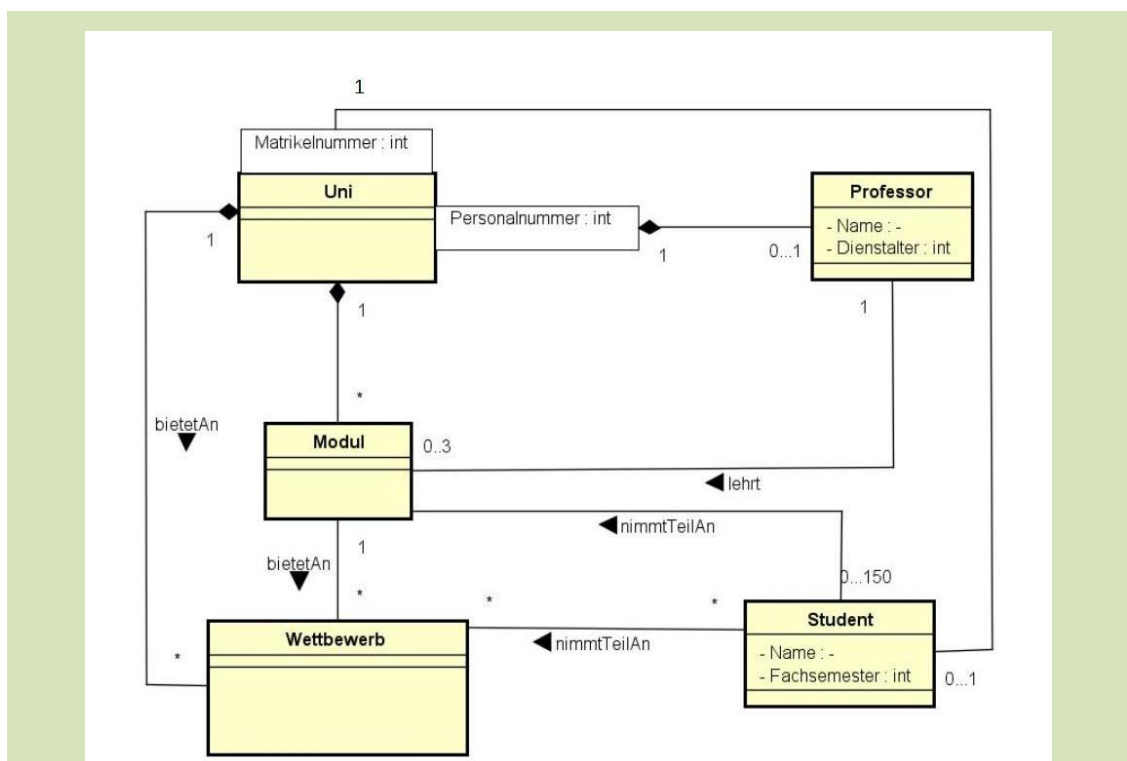
- Wintersemester 2017/18 -
Dr. Günter Kiesel

Übungsblatt 3 – Lösungen

Aufgabe 1. Klassendiagramm (15 Punkte)

a) (11 Punkte, s.u.) Modellieren Sie nachfolgende Sachverhalte als Klassendiagramm.

- I. (2,0) An einer Uni können beliebig viele Professoren lehren.
- II. (0,5) Ein Professor kann nur an einer Uni unterrichten.
- III. (1,0) Ein Professor hat einen Namen, ein Dienstalter und eine Personalnummer.
- IV. (1,0) Er kann maximal 3 Module lehren.
- V. (0,5) Ein Modul kann maximal von einem Professor gelehrt werden.
- VI. (1,0) Ein Modul kann maximal 150 Studenten fassen.
- VII. (1,0) Ein Student hat einen Namen, ein Fachsemester und eine Matrikelnummer.
- VIII. (1,0) Die Uni bietet Wettbewerbe an, an denen Studenten teilnehmen können.
- IX. (1,0) Jeder Wettbewerb ist einem Modul zugeordnet.
- X. (2,0) Im Rahmen der Uni ist jeder Professor anhand der Personalnummer eindeutig identifizierbar (d.h. zu einer bestimmten Personalnummer gibt es keinen oder genau einen Angestellten). Genauso sind Studenten eindeutig über ihre Matrikelnummer identifizierbar.



Ob ein Professor oder ein Student eine Komponente der Uni ist, mag einem Kopfzerbrechen bereiten. In der realen Welt würde man das spontan verneinen, da man an die Person denkt, die sicher nicht Teil der Uni ist, schon gar nicht exklusiv und existenzabhängig.

Wenn man allerdings an das Objekt in der modellierten Anwendung denkt und sich fragt, was dort einen Studenten ausmacht (z.B. die Matrikelnummer), könnte man schon der Meinung sein, dass ein Student der Uni Aachen schon exklusiv zur Uni Aachen gehört und z.B. dadurch, dass er

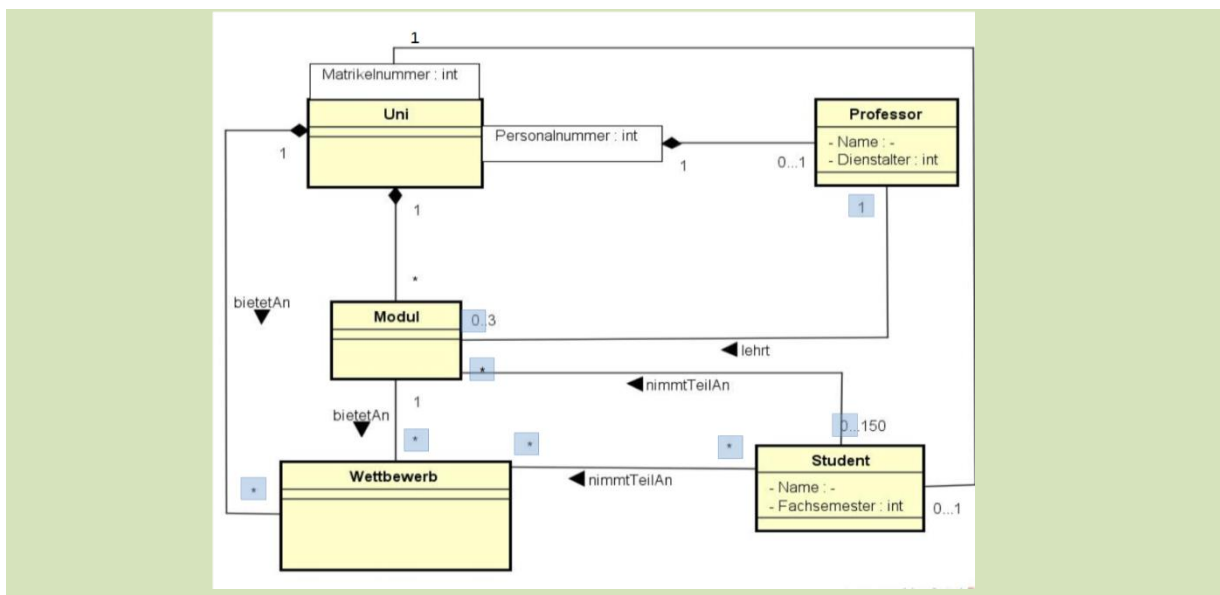
Veranstaltungen in Bonn hört, kein Student der Uni Bonn wird. Er ist und bleibt in Aachen immatrikuliert und behält z.B. seine Matrikelnummer dort. Und wenn es die Uni Aachen nicht mehr gibt, ist er plötzlich auch kein Student mehr. Die Person existiert zwar noch, aber das was sie als Student ausmacht (Matrikelnummer, Prüfungsgeschichte und Ergebnisse, etc.) wäre futsch. Die Person könnte versuchen, sich erneut woanders zu immatrikulieren, aber das würde einen neuen Studenten erzeugen mit neuer Nummer, neuer Prüfungshistorie, etc.

b) (2 Punkte) Schreiben Sie auf, welche Angaben in Teil a) unvollständig / mehrdeutig sind (erklären Sie welche Auslassungen / Mehrdeutigkeiten Sie entdeckt haben).

- I. „Beliebig viele“ klingt nach „Null bis unendlich“. Ist Null in der zu modellierenden Anwendung wirklich sinnvoll?
- II. ---
- III. ---
- IV. Wie viele Module muss ein Professor mindestens lehren?
- V. Kann ein Modul auch von 0 Professoren gelehrt werden? Macht das Sinn?
- VI. Wie viele Studenten sind mindestens nötig?
- VII. ---
- VIII. Wie viele? Sind auch 0 möglich?
- IX. Wie viele Wettbewerbe kann es zu einem Modul geben?
- X. ---

c) (2 Punkte) Ergänzen Sie vor der Abgabe die Angaben so wie es Ihnen sinnvoll erscheint, aber markieren Sie die „gerateten“ Teile des Diagramms (z.B. farblich).

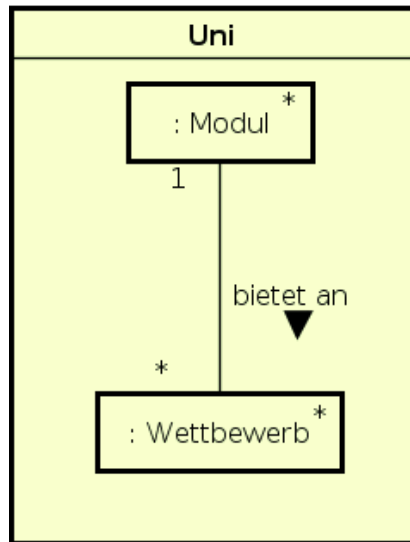
Fragen Sie im Tutorium den Kunden (Stellvertretend: Ihren Tutor) nach den fehlenden / unklaren Angaben und seien Sie bereit, Ihr Modell entsprechend der Antworten des Tutors anzupassen.



Aufgabe 2. Kompositionsstrukturdiagramm und OCL-Constraints (10 Punkte)

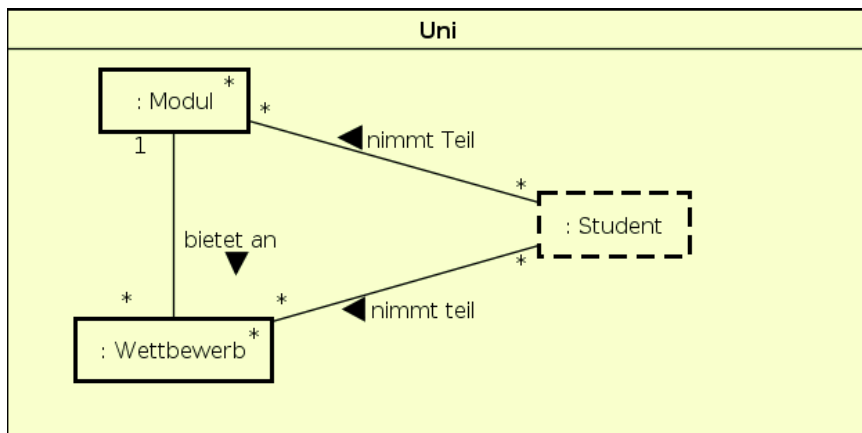
Gehen Sie im Folgenden von Ihrer Lösung zu Aufgabe 1 c) aus.

a) (4 Punkte) Drücken Sie als Ergänzung des Modells aus *Aufgabe 1* mit Hilfe eines Kompositionsstrukturdiagrammes aus, dass Module und dazugehörige Wettbewerbe stets zur selben Uni gehören müssen.



b) (4 Punkte) Können Sie das Diagramm so erweitern, dass es auch ausdrückt, dass Studenten nur an einem Wettbewerb teilnehmen können, der von der Uni angeboten wird, an der sie ein (oder mehrere) Module belegen.

Lösung: Das folgende Kompositionsstrukturdiagramm drückt aus, dass das Modul und der Wettbewerb zu der gleichen Uni gehören. Es lässt aber offen, ob unter den Modulen und Wettbewerben, an denen ein Student teilnehmen kann, immer gilt, dass jeder Wettbewerb, an dem der Student teilnimmt, von einem Modul angeboten wird, an dem der Student ebenfalls teilnimmt.



c) (2 Punkte) Wie können Sie Ihre Lösung zu b) mit Hilfe von OCL so erweitern, dass auch festgelegt wird, dass ein Student nur an dem Wettbewerb teilnehmen kann, zu dem er auch das passende Modul besucht?

Lösung: Die eigentlich in Aufgabe 1 gemeinte Beziehung zwischen Student, Modul, Wettbewerb und Uni lässt sich immer noch nicht vollständig mit a) und b) ausdrücken. Hierfür ist ein expliziter OCL-Constraint (egal ob im Klassen- oder Kompositionsstrukturdiagramm) erforderlich:

{ Student.nimmtTeilAnWettbewerb == Student.nimmtTeilAnModul.bietetAn }

Hinweis: Ob der Student gestrichelt oder mit einem durchgängigen Rand dargestellt wird hängt davon ab, ob man ihn als komponierten Teil der Uni betrachtet oder nicht. Egal wie Sie sich entscheiden: Wichtig ist, dass ihr Klassendiagramm mit dem Kompositionsstrukturdiagramm konsistent ist. Wenn

Sie im Klassendiagramm eine Komposition modelliert haben muss der Student hier auch komponierter Teil der Uni sein, also mit einem durchgezogenen Kasten dargestellt sein und umgekehrt.

Aufgabe 3. Aktivitätsdiagramme (8 Punkte)

Zeichnen Sie das Aktivitätsdiagramm für die Organisation einer Klausur. Enthalten sein sollten mindestens:

- Die Teilnahme an den Übungen (erfolgreich / nicht erfolgreich)
- Die erfolgreiche Teilnahme führt zur Klausuranmeldung, dem Erfragen des Termins und sorgfältiger Vorbereitung in beliebiger Reihenfolge.
- Die Klausur selbst mit anschließender Abfrage der Ergebnisse im Internet.
- Eine einmalige Zulassung zur Nachklausur, wenn die Klausur im ersten Versuch nicht bestanden wurde.
- Die Nachholklausur erfordert eine erneute Terminabfrage und Vorbereitung.
- Ein Krankheitsfall kann ebenfalls eintreten. Mit einem Attest und einer anschließenden Genesung ist eine Wiederholung der jeweiligen Klausur möglich. Dies ist beliebig oft möglich.

Beachten Sie, dass Aktivitäten auch parallel oder alternativ ablaufen können. Jeder der obigen Punkte kann dabei mehrere Aktionen umfassen, wenn es Ihnen angebracht erscheint.

