

inoffizielle Musterlösung zu

Objektorientierte Modellierung
(TU/Kappel) SS 2008

Übungsblatt 6

Anmerkung: Die Beispiele in diesem Übungsblatt sind sehr frei definiert. Entsprechend unwahrscheinlich ist es, dass zwei Studierende auf die exakt selbe Lösung kommen. Daher wird eine Präsentation einer Lösung, die bereits in einer zeitlich vorgelagerten Gruppe vorgestellt wurde, negativ bewertet. Verwenden Sie auch bei den Theoriefragen andere Beispiele als jene aus der Vorlesung.

Aufgabe 1: Anwendungsfallmodellierung – Theoriefragen I

Wiederholen Sie das Kapitel aus der Vorlesung, das sich mit dem UML2-Anwendungsfalldiagramm beschäftigt.

- a) Welche Idee steckt hinter der Verwendung von Use Cases? Was ist ein Use Case?
- b) Beschreiben Sie die Bestandteile eines Use-Case-Diagramms an Hand eines Beispiels.
- c) Auf welche Bestandteile des Anwendungsfalldiagramms kann das Konzept der Generalisierung angewendet werden? Geben Sie entsprechende Beispiele.
- d) Erklären Sie anhand eines Beispiels wieso Akteure keine Objekte sondern Rollen repräsentieren.

Aufgabe 2: Anwendungsfallmodellierung – Theoriefragen II

- a) Wozu dient eine Anwendungsfallbeschreibung? Wie könnte man eine Anwendungsfallbeschreibung strukturieren? Was gehört in eine solche Beschreibung?
- b) Diskutieren Sie die Erweiterungsbeziehung “extend” in Anwendungsfalldiagrammen. Finden Sie ein Beispiel
- c) Diskutieren Sie die Erweiterungsbeziehung “include” in Anwendungsfalldiagrammen. Finden Sie ein Beispiel.
- d) Was sind Erweiterungsstellen (extension points)? Finden Sie ein Beispiel.

Aufgabe 3: Anwendungsfallmodellierung – OOM VU

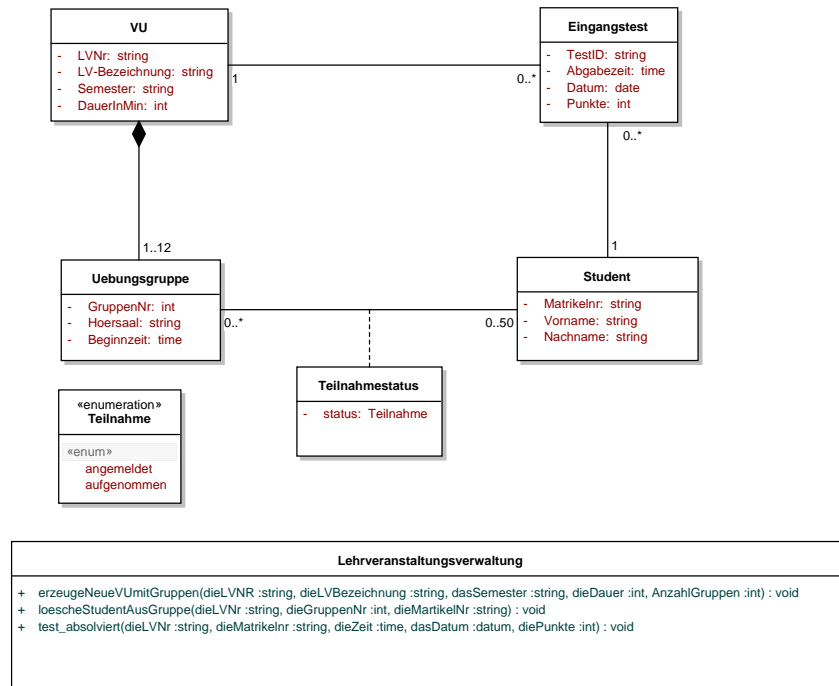
Nehmen Sie an es gibt kein TUWEL. Sie sind damit beauftragt ein Informationssystem für die Abwicklung von VUs gemäß dem Modus von OOM zu entwickeln. In einem ersten Schritt entwerfen Sie ein entsprechendes Anwendungsfalldiagramm. Beschreiben Sie mindestens einen Anwendungsfall in Prosa an Hand der in der Vorlesung vorgestellten Punkte (Seite 29 und 30). Nehmen Sie die Beschreibung auf Folie mit in die Übung - kann auch handgeschrieben sein.

Aufgabe 4: Prozessmodellierung – OOM Eingangstest

Entwerfen Sie mit Hilfe eines Aktivitätsdiagramms den Ablauf bei der Durchführung des OOM Eingangstests. Modellieren Sie welche Aktivitäten von der Studentin und welche vom System durchgeführt werden.

Aufgabe 5: Operationen und Sequenzdiagramme – OOM Eingangstest

Gebgeben sei der folgende Ausschnitt aus einem Klassendiagramm des LV-Verwaltungssystems:



Dieses Klassendiagramm beschreibt statische Beziehungen zwischen Klassen. Dabei seien folgende Annahmen gültig: Einer VU werden exklusiv ein bis zwölf Übungsgruppen zugeordnet. In eine Übungsgruppe können bis zu 50 Studenten zugeordnet werden. Ein Student kann mehreren Übungsgruppen zugeordnet sein, jedoch immer nur einer Übungsgruppe einer entsprechenden VU (letzteres ist im Klassendiagramm nicht ersichtlich). Ein Student befindet sich in einer Übungsgruppe entweder im Zustand *angemeldet* oder *aufgenommen*. Sollte er nicht aufgenommen werden, so wird die Zuordnung gelöscht.

Für eine VU kann ein Student einen Eingangstest absolvieren. Dabei wird für jeden Studenten für jede VU ein eigenes Eingangstest-Objekt erzeugt, in dem das jeweils beste Ergebnis des Eingangstests gespeichert wird. Ein Student kann natürlich wieder mehreren Eingangstests zugeordnet sein, aber für unterschiedliche VUs.

Weiters sei die Klasse Lehrveranstaltungsverwaltung gegeben, welche Operationen enthält, die von der Benutzerschnittstelle aufgerufen werden.

Erweitern Sie das Klassendiagramm um entsprechende Methoden und eventuell zusätzlich benötigte Klassen um

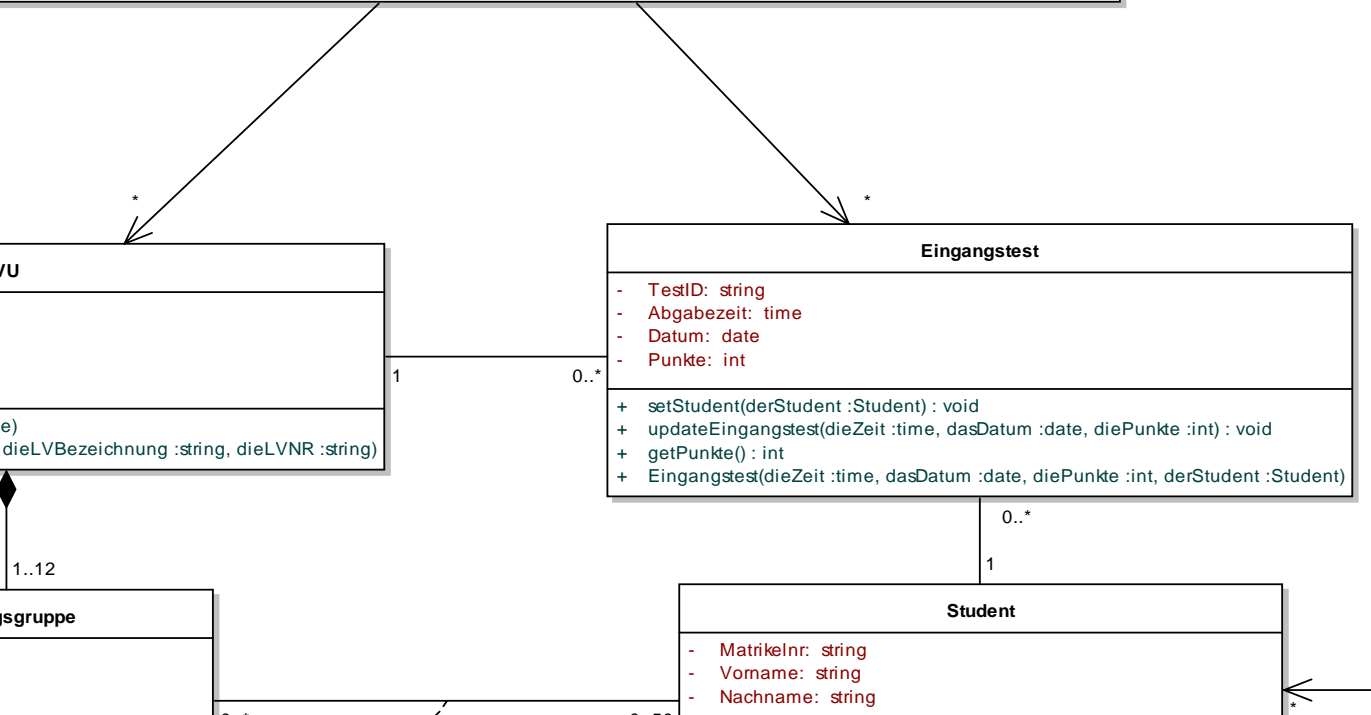
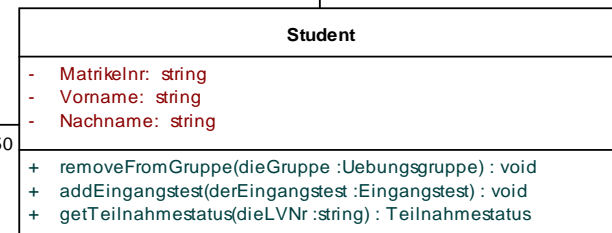
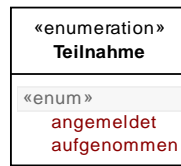
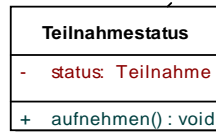
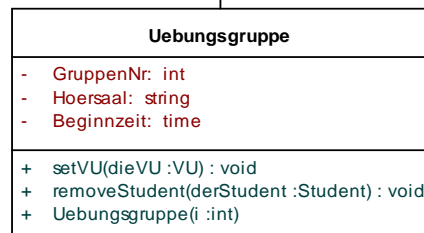
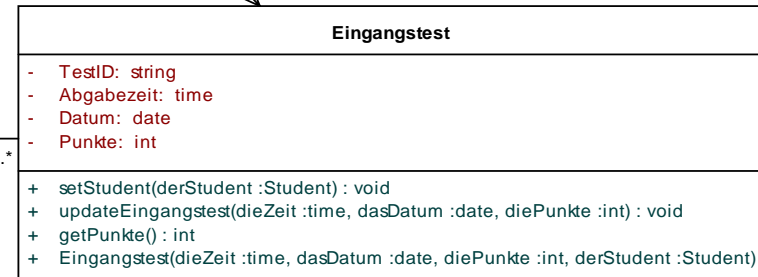
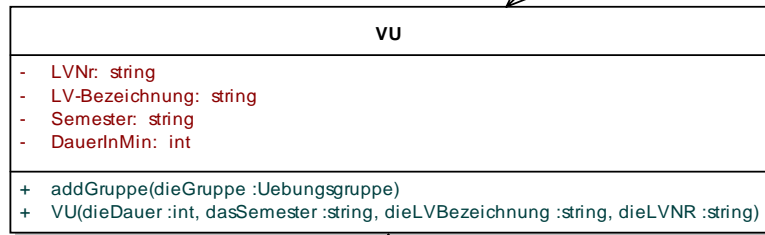
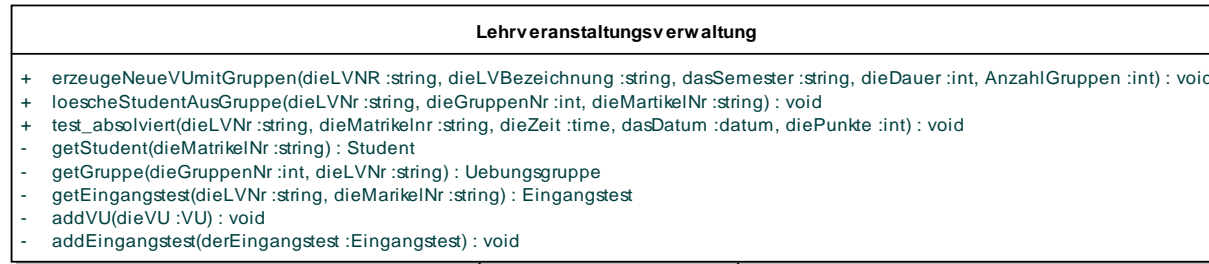
- das Erzeugen einer neuen VU mit Gruppen
- das Löschen eines Studenten aus einer Gruppe

wie in der Klasse Lehrveranstaltungsverwaltung vorgesehen, durchzuführen. Zeigen Sie die entsprechenden Methodenaufrufe in einem zugehörigen Sequenzdiagramm.

Aufgabe 6: Operationen und Sequenzdiagramme – OOM Eingangstest II

Fortsetzung des vorhergehenden Beispiels. Erweitern Sie das Klassendiagramm um entsprechende Methoden und eventuell zusätzlich benötigte Klassen, um den Aufruf eines absolvierten Tests entsprechend zu behandeln. Zeigen Sie wieder die entsprechenden Methodenaufrufe in einem zugehörigen Sequenzdiagramm.

class Domain Objects



:LV-System

:Lehrveranstaltungsverwaltung

:VU

:Uebungsgruppe

erzeugeNeueVUmitGruppen(dieLVNR,dieLVBezeichnung,dasSemester,dieDauer,AnzahlGruppen)

VU(dieDauer,dasSemester,dieLVBezeichnung,dieLVNR)

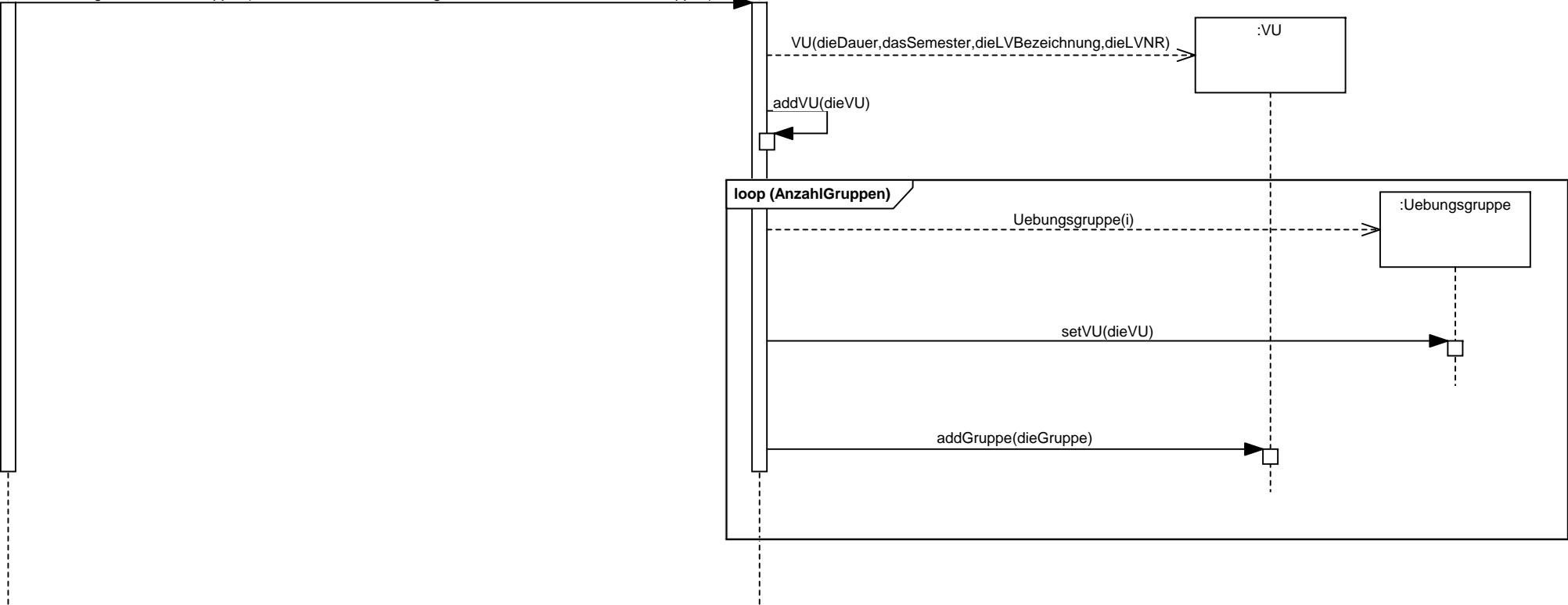
addVU(dieVU)

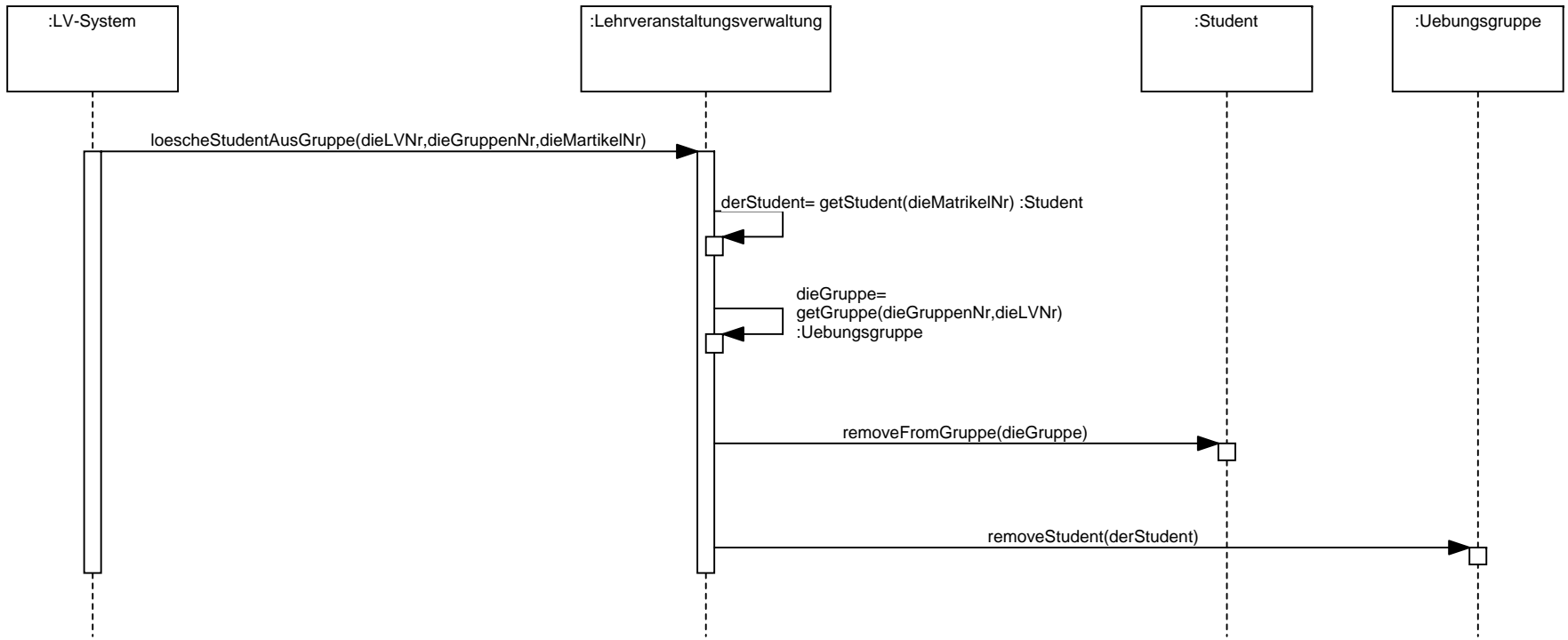
loop (AnzahlGruppen)

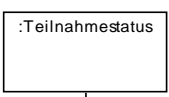
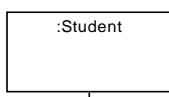
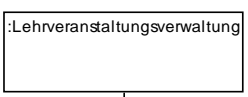
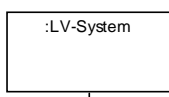
Uebungsgruppe(i)

setVU(dieVU)

addGruppe(dieGruppe)







test_absolviert(dieLVNr,dieMatrikelNr,dieZeit,dasDatum,diePunkte)

derStudent= getStudent(dieMatrikelNr)

derEingangstest= getEingangstest(dieLVNr,dieMatrikelNr) :
Eingangstest

alt
[derEingangstest = null]

Eingangstest(dieZeit,dasDatum,diePunkte,derStudent) :Eingangstest

addEingangstest(derEingangstest)

addEingangstest(derEingangstest)

[else]

vorherPunkte= getPunkte() :int

opt
[diePunkte > vorherPunkte]

updateEingangstest(dieZeit,dasDatum,diePunkte)

opt
[vorher Punkte < 50 AND die Punkte >=50]

derTeilnahmestatus= getTeilnahmestatus(dieLVNr) :Teilnahmestatus

aufnehmen()

