

Übungen zu Grundlagen der Systementwicklung
(Prof. Dr. M. Wirsing, Dr. P. Kosiuczenko, A. Rauschmayer)

Aufgabe 20 (Präsenzaufgabe, Modellierung und Verfeinerung)

Das Uni-Beispiel aus Aufgabe 16 kann man so erweitern, daß nicht mehr nur ein Kurs, sondern eine Liste von Kursen verwaltet werden können. Eine Uni besteht demnach aus einer Liste von Kursen. Ein Kurs umfaßt wie bisher eine Menge von Studenten und eine maximale Teilnehmeranzahl und ist als Schema *Class* spezifiziert. Aus Effizienzgründen will man in einer Implementierung die Teilnehmer nicht als Menge verwalten, sondern als Liste.

a) Erweitern Sie das Uni-Beispiel:

- Erstellen Sie ein Schema *Uni* für die Universität
- Spezifizieren Sie die Operation *Register_Student*, die neue Studenten in einem vorgegebenem Kurs einschreibt und die Operation *Register_Class*, die neue Klassen einführt.
- Spezifizieren Sie eine globale Operation *count*, die die Gesamtzahl aller an der Uni eingeschriebenen Studenten zurückgibt.

b) Verfeinern Sie das Uni-Beispiel:

- Verfeinern Sie das Schema *Class*, indem Sie Studenten statt in Mengen in Listen verwalten.
- Verfeinern Sie außerdem die Operationen (*Enrol_Ok*, *Already_Enrolled*, *Fully_Booked*, ...).
- Definieren Sie eine geeignete Kopplungsinvariante *Rep* und beweisen Sie, daß Verfeinerung bezüglich *Rep* vorliegt.

Aufgabe 21 (**Hausaufgabe**, Jaza)

Jaza ist ein Werkzeug zum Ausführen und Testen von Z-Spezifikationen. Es ist im CIP-Pool bereits installiert, Hinweise zur Installation auf dem PC finden Sie hier:

<http://www.pst.informatik.uni-muenchen.de/lehre/WS0203/gse/z-notation/>

Starten Sie Jaza und laden Sie die Ihre Lösung der letzten Hausaufgaben, die Spezifikation des Iterators. Sie können auch die Musterlösung verwenden:

```
r/rauschma> jaza
Welcome to Jaza, version 0.90 (beta).
Copyright(C) 1999-2002 Mark Utting (marku@cs.waikato.ac.nz).
Jaza comes with ABSOLUTELY NO WARRANTY (see file COPYING).
This is free software, and you are welcome to redistribute
it under certain conditions (see file COPYING).
```

Type 'help' to see the available commands.

```
JAZA> load haus_lsg_07.tex
load haus_lsg_07.tex
Loading 'haus_lsg_07.tex' ...
Added 6 definitions.
JAZA>
```

- Starten Sie mit der Sequenz $\langle 1, 2, 3 \rangle$ und führen Sie vier mal `Next` aus.
- Achtung! Damit der Initial-Zustand mit den Folge-Zuständen komponiert werden kann, muss er als `InitIterator` angegeben werden.
- Rufen Sie zwischendurch und vor dem letzten `Next` die Operation `HasNext` auf.
- Die Jaza-Dokumentation erwähnt auch, wie man Testing mit Jaza-Durchführt: Ein- *und* Ausgabewerte werden angegeben. Wenn sie bezüglich einer Operation korrekt sind, wird ein leeres `\lbracket \rbracket` ausgegeben. Starten Sie wieder mit $\langle 1, 2, 3 \rangle$ und überprüfen sie, ob der erste `Next`-Aufruf korrekt funktioniert.
- Geben Sie bitte das ausgedruckte Protokoll der Interaktion mit Jaza ab.

Frohe Weihnachten und ein gutes neues Jahr!

Abgabe: Bis Freitag, 10. Januar zu Beginn der Übung.