

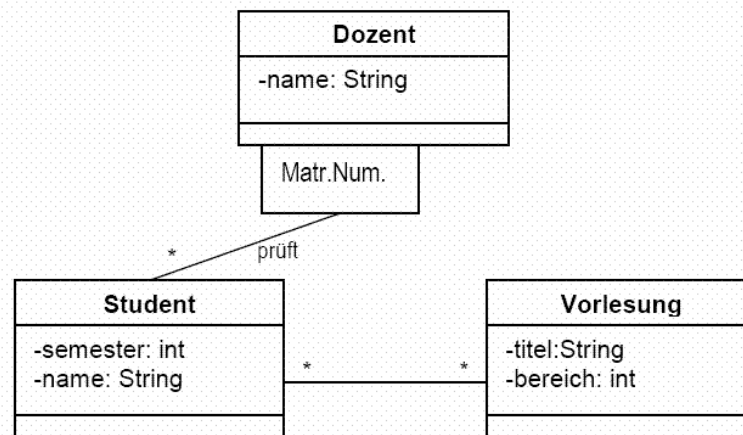
Quelle:

[http://www4.in.tum.de/lehre/vorlesungen/sw\\_engineering/WS0203/uebung/blatt14.pdf](http://www4.in.tum.de/lehre/vorlesungen/sw_engineering/WS0203/uebung/blatt14.pdf)

[http://www4.in.tum.de/lehre/vorlesungen/sw\\_engineering/WS0203/uebung/loesung14.pdf](http://www4.in.tum.de/lehre/vorlesungen/sw_engineering/WS0203/uebung/loesung14.pdf)

### Aufgabe 31 : OCL (T)

Gegeben sei folgendes Klassendiagramm. Die Studenten sollen sich zu einer Prüfung anmelden. Dabei sollen einige Bedingungen erfüllt sein. Drücken Sie mit der OCL die unten gegebenen Bedingungen aus.



- Studenten können maximal im 16ten Semester sein.
- Es gibt die Vorlesungsbereiche 0, 1 und 2.
- Die Studenten mit den Matrikelnummern 1000, 500 und 123 haben disjunkte Vorlesungen.
- Ein Student muss aus jedem Bereich genau eine Vorlesung angeben.
- Jeder Vorlesungstitel muss eindeutig sein.
- Zu jeder Vorlesung muss mindestens ein Dozent eine Prüfung halten.

## Lösung

### a) Wertebeschränkung(1)

```
context Student inv: self.semester <= 16
```

### b) Wertbeschränkung (2)

```
context Vorlesung inv:  
(self.bereich >= 0) && (self.bereich <= 2)
```

### c) Auflösen von qualifizierten Assoziationen

```
context Dozent inv:  
(forall v in self.prüft[1000].vorlesung:  
    !(exists w in self.prüft[500].vorlesung: v == w))  
&& (forall v in self.prüft[1000].vorlesung:  
    !(exists w in self.prüft[123].vorlesung: v == w))  
&& (forall v in self.prüft[500].vorlesung:  
    !(exists w in self.prüft[123].vorlesung: v == w))
```

Es gibt auch andere effizientere Möglichkeiten, diese Bedingung auszudrücken. Die hier gegebene Lösung ist nur ein Vorschlag.

### d) Einteilung

```
context Student inv:  
(exists v in self.vorlesung: v.bereich == 0)  
&& (exists v in self.vorlesung: v.bereich == 1)  
&& (exists v in self.vorlesung: v.bereich == 2)  
&& (self.vorlesung->size() == 3)
```

Auch diese Bedingung lässt sich auf verschiedenste und evtl. einfachere Weisen ausdrücken.

### e) Eindeutigkeit

```
context Vorlesung inv:  
forall Vorlesung v: forall Vorlesung w:  
(( v == w) || ((v != w) && (v.titel != w.titel)))
```

Alternatives Constraint mit gleicher Aussage:

```
context Vorlesung inv:  
forall Vorlesung v: forall Vorlesung w: ((w.titel == v.titel)  
implies (w == v))
```

### f) Mindesangabe

```
context Vorlesung inv:  
self.student.prüft->size() > 0
```