

## Formale Systeme, WS 2009/2010

### Übungsblatt 11

Dieses Übungsblatt wird in der Übung am 05.02.2010 besprochen.

Dieses Übungsblatt beschäftigt sich mit dem folgenden Klassendiagramm:

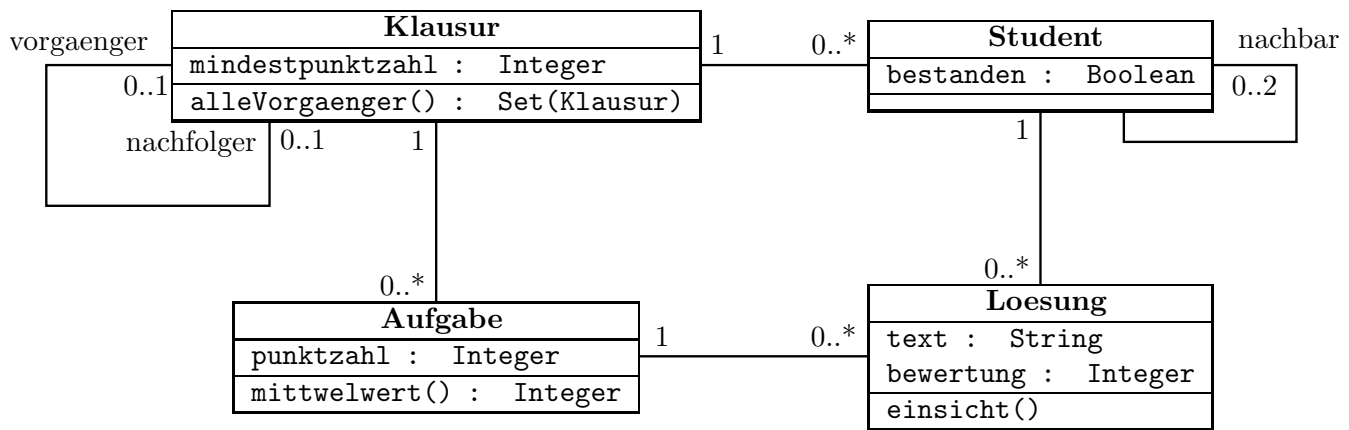


Abbildung 1: UML-Klassendiagramm (*nachfolger* und *vorgaenger* bezeichnen die einer Klausur direkt vorausgehende bzw. nachfolgende Klausur zur selben Vorlesung).

#### Aufgabe 1

Geben Sie OCL-Bedingungen (Invarianten) an, die die folgenden Sachverhalte modellieren:

- Eine Klausur und die ihre nachfolgende Klausur haben keine Aufgaben gemeinsam.
- Wenn ein Student bestanden hat, dann ist die Summe der Bewertungen seiner Lösungen größer oder gleich der Mindestpunktzahl der Klausur. (Verwenden Sie das **sum**-Konstrukt.)
- Wer abschreibt, besteht die Klausur nicht, wer abschreiben läßt – ebenso. Sie können davon ausgehen, dass es sich um Abschreiben handelt, wenn zwei Nachbarn die gleiche Lösung haben.

#### Aufgabe 2

Spezifizieren Sie die folgenden Operationen mittels Vor-/Nachbedingungs-paaren.

- `mittelwert()` berechnet die Durchschnittsbewertung für die Lösungen einer Aufgabe.
- Nach `einsicht()` – was für die Klausureinsicht steht – ist die Bewertung einer Lösung nicht schlechter als zuvor.<sup>1</sup>

<sup>1</sup>Es gibt keine Garantie, dass das in der Realität so ist.

### Aufgabe 3

Spezifizieren Sie, dass die **vorgaenger/nachfolger**-Relation azyklisch ist. Spezifizieren Sie dafür zuerst die Operation **alleVorgaenger()**. Beachten Sie dabei die folgende Aussage des OCL-Standards:

The right-hand-side of this [postcondition] definition may refer to the operation being defined (i.e., the definition may be recursive) as long as the recursion is not infinite.

Was bedeutet diese Einschränkung aus dem Standard für die Syntaxdefinition von OCL?

### Aufgabe 4

Laut OCL Standard liefert die Operation **reject** mit der Signatur

```
source->reject(iterator | body)
```

die Gesamtheit, die genau die Elemente in **source** enthält, für die **body** zu falsch ausgewertet. Der Typ des Ergebnisses entspricht dabei dem Typ von **source**.

Geben Sie eine Definition des **reject**-Konstrukts mittels des **iterate**-Konstrukts an. Nehmen Sie dabei an, dass **source** den Typ **Set(T)** hat.