

Formale Systeme, WS 2008/2009

Übungsblatt 10

Dieses Übungsblatt wird in der Übung am 13.02.2009 besprochen.

Dieses Übungsblatt beschäftigt sich mit dem folgenden Klassendiagramm:

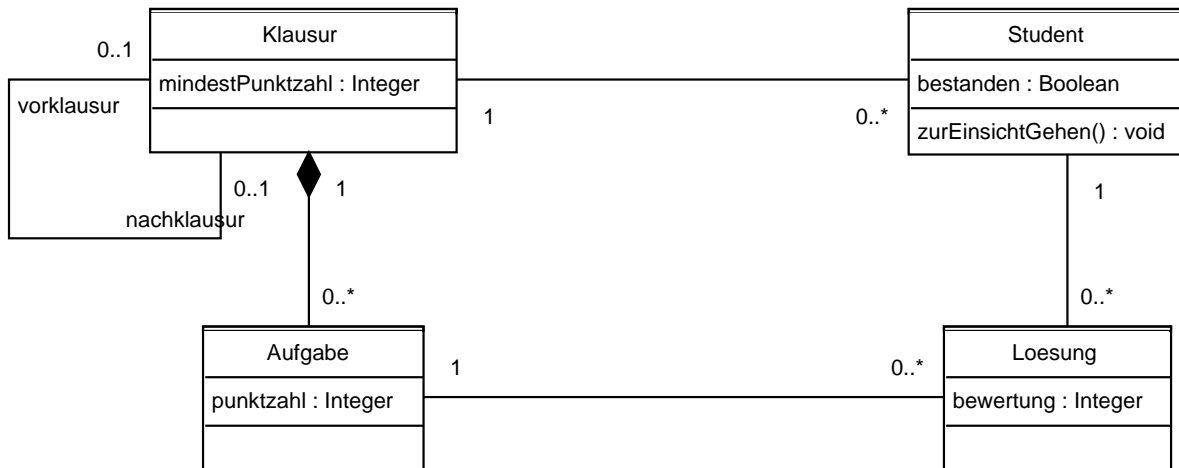


Abbildung 1: UML-Klassendiagramm

Aufgabe 1

Geben Sie natürlichsprachliche Übersetzungen der folgenden OCL-Bedingungen an:

- (a) context Loesung
inv: bewertung >= 0 and bewertung <= aufgabe.punktzahl
- (b) context Student
inv: self.loesung->size() <= klausur.aufgabe->size()
- (c) context Klausur
inv: aufgabe->includesAll(student.loesung.aufgabe)

Aufgabe 2

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

- (a) Eine Klausur und ihre Nachklausur haben keine gemeinsamen Aufgaben.
- (b) Jede Klausur ist entweder Vorklausur einer anderen Klausur oder Nachklausur einer anderen Klausur, aber nie beides.

- (c) Wenn ein Student bestanden hat, dann ist die Summe der Bewertungen der bearbeiteten Aufgaben größer oder gleich der Mindestpunktzahl der Klausur.

Hinweis: Verwenden Sie für diese Modellierung das **sum**-Konstrukt.

Aufgabe 3

Die Funktion **union** operiert auf einer Menge, beschreibt die Vereinigung von Mengen und ist folgendermaßen spezifiziert:

```
-- union(s:Set(T)) : Set(T)
-- The union of self:Set(T) and s:Set(T)
pre:  -- none
post: result -> forAll(elem | self->includes(elem) or s->includes(elem))
post: self -> forAll(elem | result->includes(elem))
post: s -> forAll(elem | result->includes(elem))
```

Geben Sie eine Definition des **union**-Konstrukts mittels des **iterate**-Konstrukts an.

`Set(T)` als Typ bedeutet dabei, dass es sich um eine Menge handelt, deren Elemente vom Typ `T` sind.

Aufgabe 4

Die Klasse `Student` besitzt eine Methode “zurEinsichtGehen”. Formulieren Sie einen Methodenvertrag in OCL, der besagt:

Ein Student wird nur zur Einsicht zugelassen, wenn er mindestens eine Aufgabe bearbeitet hat. Dann ist die Bewertung jeder Aufgabe nach der Einsicht aber nicht kleiner als davor.