

Achtung: Die Abgabefrist für diesen und folgende Übungszettel liegt wieder auf Montag 10:00 Uhr und wird sich nicht mehr ändern.

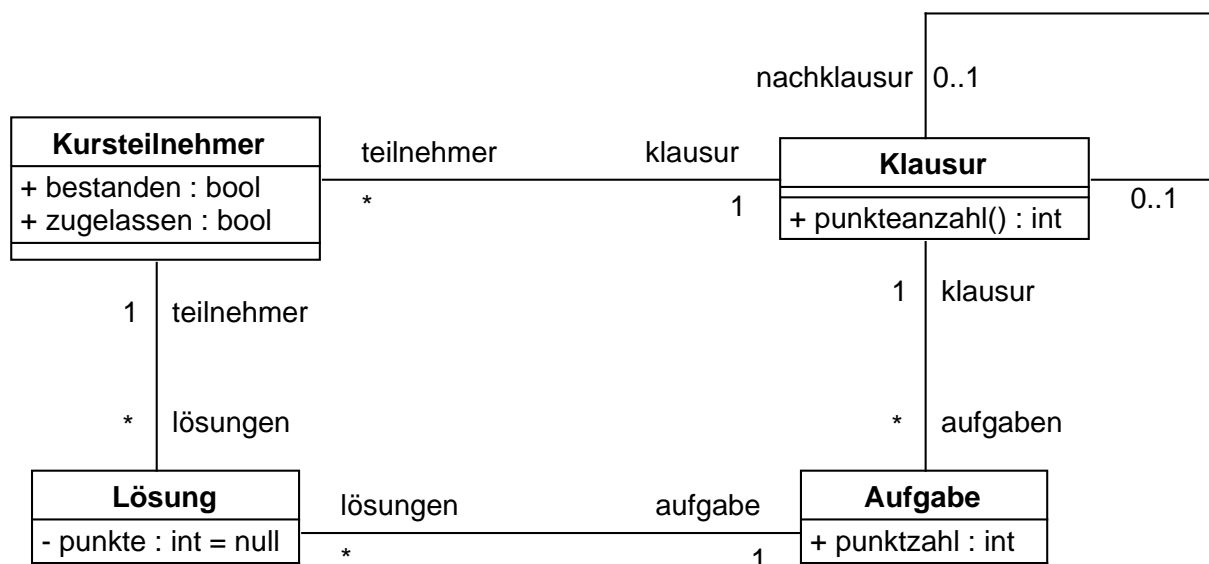
Aufgabe 9-1: Begriffe

- a) Was versteht man im Kontext der Softwareentwicklung unter dem Begriff „*Information Hiding*“?
Erläutern Sie das Prinzip und erklären Sie, in welchem Zusammenhang es zum „*Need to Know*“-Prinzip steht.
- b) Warum ist das Prinzip des Information Hiding sinnvoll? Erläutern Sie dieses an einem Beispiel.
- c) Was versteht man unter dem „*Design by Contract*“-Prinzip? In welchem Kontext wird es eingesetzt und warum?
- d) Erläutern Sie, in welchem Zusammenhang die Begriffe *OCL*, *Invariante*, *Design by Contract*, *precondition*, *constraints*, *postcondition* stehen.

Aufgabe 9-2: OCL lesen und schreiben ★

In einem Prüfungsverwaltungssystem sollen die Klausurergebnisse und die Punktzahlen der Studierenden bei den Lösungen der einzelnen Aufgaben verwaltet werden.

Das folgende UML-Klassendiagramm modelliert einen Teil der benötigten Daten.



- a) Drücken Sie auf Basis des Diagramms die OCL-Constraints c1 bis c3 natürlichsprachlich aus, also in einer Form, die Sie in einem Gespräch mit einer Person ohne Informatik-Hintergrund verwenden würden.

```
context Aufgabe inv c1: punktzahl > 0

context Kursteilnehmer inv c2: lösungen->size() = klausur.aufgaben->size()

context Klausur inv c3:
  teilnehmer->forAll (t |
    t.bestanden implies t.lösungen->exists (l |
      l.punkte > 0 and l.aufgabe.klausur = self
    )
  )
```

- b) Geben Sie OCL-Constraints an, die die folgenden Sachverhalte formalisieren. Achten Sie darauf, syntaktisch einwandfreie OCL-Ausdrücke zu formulieren. Das schließt auch die Beachtung der genauen Schreibweisen von Klassen, Attributen, Operationen und Assoziationen aus dem UML-Klassendiagramm ein.

Schauen Sie auch in der OCL 2.4 Spezifikation (<http://www.omg.org/spec/OCL/2.4/PDF>) nach, falls Ihnen Ausdrucksmittel fehlen.

1. In jeder Klausur gibt es mindestens eine Aufgabe mit genau einem Punkt.
2. Eine Nachklausur kann keine Nachklausur haben.
3. Ist ein/e Kursteilnehmer/in zugelassen, gibt es für jede Klausuraufgabe auch eine Lösung von ihm/ihr.

Aufgabe 9-3: OCL-Modellerweiterungen

Diese Aufgabe baut auf dem gleichen UML-Modell auf wie Aufgabe 9-2. Nun soll zusätzlich auch die Klausurkorrektur modelliert werden: Bei der Korrektur geht der/die Dozent/in alle Lösungen einzeln durch, bewertet sie jeweils und errechnet die Gesamtpunktzahl für jede/n Teilnehmer/in.

- a) Erweitern Sie das Modell nun um eine Operation `korrigieren` mit passender Signatur sowie ein neues Attribut; darüber hinaus darf das Klassendiagramm in keiner Weise verändert werden (also keine neuen Klassen, veränderte Sichtbarkeiten, etc.). Die Operation `korrigieren` wird aufgerufen, sobald der/die Dozent/in eine Lösung korrigiert hat.

Es soll damit insbesondere möglich sein, dass

- das Attribut `punkte` in `Lösung` gefüllt wird, und
- die erreichte Gesamtpunktzahl des Teilnehmers aktualisiert wird.

Beschreiben Sie zunächst verbal, was `korrigieren` genau leisten soll. Finden Sie geeignete Stellen (und für das neue Attribut: auch einen geeigneten Namen) für die neuen Member im Klassendiagramm.

- b) Gehen Sie zunächst in einer ersten Version davon aus, dass `korrigieren` nur einmal pro Exemplar der Klasse `Lösung` aufgerufen werden darf; die einmal gesetzte Punktzahl ist danach unveränderlich.

Spezifizieren Sie sowohl möglichst strenge Vorbedingungen für Ihre Operation `korrigieren`, als auch alle Effekte (*postcondition*) der Operation komplett in OCL.

- c) Gehen Sie nun von der realistischeren Anforderung aus, dass `korrigieren` mehrfach aufgerufen werden kann, um z.B. die Punktzahl einer Lösung bei einer Klausureinsicht zu korrigieren.

Spezifizieren Sie wiederum möglichst strenge Vorbedingungen und alle Effekte der Operation in OCL.

- d) Spezifizieren Sie den Effekt der Operation `Klausur.punkteanzahl` unter Verwendung des OCL-Schlüsselwortes `result`.

Hinweis: Der resultierende Ausdruck passt bequem auf eine Zeile. Recherchieren Sie bei Bedarf nach OCL-Collections und ihren Operationen.