



Formale Methoden

SS 2012 – Übungsblatt 12 (klausurähnliche Aufgaben)

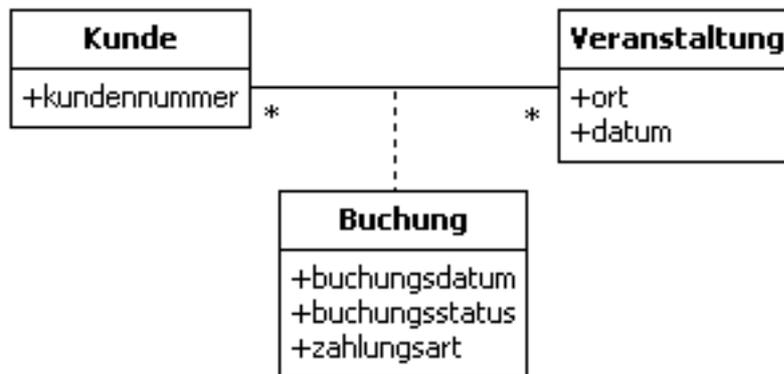
4. Juli 2012

Ausgabe: 5. Juli 2012

ohne Abgabe

Aufgabe 1. *Assoziationsklassen*

Was ist eine Assoziationsklasse wie zum Beispiel:



Welche Vorteile hat sie?

Schreiben Sie die OCL-Constraints:

- Das Buchungsdatum darf nicht länger als zwei Jahre in der Vergangenheit liegen.
- Veranstaltungsdaten dürfen nicht in der Vergangenheit liegen.
- Für Veranstaltungen in den nächsten drei Tagen muss die Zahlungsart **bar** gewählt werden.
- Kunden mit Kundennummern größer als 10000 dürfen die Zahlungsart **Kreditkarte** wählen.

Schreiben Sie ein analoges UML-Diagramm für den Fall, dass Assoziationsklassen von Ihrem UML-Tool nicht unterstützt werden oder beschreiben Sie den in der Vorlesung benutzten Workaround für das aktuelle Papyrus.

Aufgabe 2. *OclHelper*

Was sind OclHelper? Definieren Sie (in OCL-Syntax) ein Helper-Attribut und eine Helper-Methode für das Modell „Kunde/Buchung/Veranstaltung“ von Aufgabe 1.

Wo überall dürfen sie benutzt werden?

Aufgabe 3. *OCL-Constraints*

Entwerfen Sie in UML-Form ein Softwaresystem zu:

- Rathaus
- Etage
- Pförtnerloge
- Büro
- Bürger
- Waschraum
- Cafeteria

Zeichnen Sie das zugehörige UML-Klassendiagramm!

Schreiben Sie fünf sinnvolle OCL-Constraints der folgenden Form:

- Jedes Rathaus enthält genau eine Pförtnerloge.
- Jede Etage enthält mindestens einen Waschraum.
- ...

Aufgabe 4. *(umgangssprachliche) Bedeutung von OCL-Constraints*

Erläutern Sie in eigenen Worten (umgangssprachlich) die Bedeutung der folgenden OCL-Constraints/Ausdrücke beziehungsweise ergänzen Sie Spezifikationen:

- context Pruefungsergebnisvermerk
student.kursnote[pruefungsnummer] = self
- context Pruefungsergebnisvermerk
inv : Pruefungsergebnisvermerk.allInstances()
->isUnique(pruefungsnummer)
- context Universitaet
inv : student->size() > 0 implies
student.kursnote->size() > 0 implies
student.kursnote->asSet()->isUnique(pruefungsnummer)
- context Universitaet
inv : name <> ''
inv : student->size() >= 0
inv : immatrikulation->isUnique(matrikelnummer)
inv : student.universitaet->includes(self)
inv : student[03123456].familienname = 'Bauer'

```
- context Pruefungsergebnisvermerk
  inv : kurs <> ''
  inv : 1 <= note and note <= 5
  inv : matrikelNr > 0
  inv : student.immatrikulation.matrikelNummer->includes(matrikelNr)
  inv : pruefungsnummer > 0
```