

Übungen zur Vorlesung Softwaretechnologie

- Wintersemester 2014/2015 -

Dr. Günter Kiesel

Übungsblatt 7

Zu bearbeiten bis: 05.12.2014

Bitte fangen Sie **frühzeitig** mit der Bearbeitung an, damit wir Ihnen bei Bedarf helfen können. Checken Sie die Lösungen zu den Aufgaben in Ihr Repository ein, „Erklärungen“ bitte als Textdatei.

Fragen zu Übungsaufgaben/Vorlesung können Sie auf der Mailingliste swt-tutoren@lists.iai.uni-bonn.de, bzw. swt-vorlesung@lists.iai.uni-bonn.de stellen.

Aufgabe 1. CRC-Karten (7 Punkte)

Es geht um die Modellierung der folgenden Bibliotheks-Anwendung:

Die Unibibliothek verwaltet Bücher, Zeitschriften und Seminararbeiten. Diese Gegenstände sollen rechnergestützt gefunden und ausgeliehen werden.

Zur Identifikation eines Benutzers bei Ausleihe und Rückgabe dient der Informatik-Login. Jeder Benutzer kann höchstens sieben Medien entleihen. Die Ausleihfrist beträgt bei Büchern 4 Wochen, bei Zeitschriften 2 Wochen und bei Seminararbeiten 1 Woche. Die Überziehungsgebühr beträgt bei Büchern 1 €, bei Zeitschriften 3 €, bei Seminararbeiten 5 € pro Tag der Überschreitung.

Benutzer dürfen nur dann weitere Medien ausleihen, wenn sie noch keine 42€ Überziehungsgebühren angesammelt haben und auch bei keinem entliehenen Medium die Frist überziehen.

- a) Erstellen Sie ein Glossar für die Bibliotheksanwendung:
 - a. Eine textuelle Aufzählung von Klassen mit aussagekräftigen Namen.
 - b. Je ein kurzer prägnanter Text (je < 50 Wörter) der die Aufgabe der Klasse beschreibt.
- b) Unter Einsatz von CRC-Cards¹ soll das Objektmodell um Verantwortlichkeiten und Kollaborationen erweitert werden, und zwar so, dass die Klassen anschließend alles beinhalten, was für folgende Szenarien erforderlich ist:
 - a. Suchen
 - b. Erfolgreiches Ausleihen
 - c. Verweigertes Ausleihen
 - d. Rückgabe ohne Überziehung
 - e. Rückgabe mit Überziehung

Wenden Sie die in der Vorlesung eingeführten Techniken an. Es kann durchaus sein, dass Sie auf neue sinnvolle Klassen stoßen. Beschreiben Sie diese gegebenenfalls auch durch je eine CRC-Karte.

¹ Abgabe in Tabellenform als PDF-Dokument

Aufgabe 2. *Design by Contract* (7 Punkte)

Verfeinern Sie ihren Entwurf zu Aufgabe 1 wie folgt:

- a) Erstellen Sie ein Klassendiagramm, das die Ergebnisse der CRC-Analyse widerspiegelt, aber komplett typisiert ist (Ergebnis- und Parametertypen zu jeder Operation). Achten Sie darauf, dass manche Beziehungen evtl. eigene Attribute und Operationen besitzen.
- b) Beschreiben Sie für drei Operationen mit unterschiedlichen Verantwortlichkeiten, jeweils ihre Vor- und Nachbedingungen, die sich aus Aufgabe 1 ableiten lassen. Versuchen sie diese so genau wie möglich zu formulieren, unter Verwendung der Object Constraint Language (OCL).

Aufgabe 3. *Design by Contract* (11 Punkte)

- a) Erläutern Sie die drei Kernkonzepte von „Design by Contract“.
- b) Wie können diese Konzepte in Java durch die Verwendung von `assert()`-Anweisungen umgesetzt werden (siehe <http://www.oracle.com/technetwork/articles/javase/assertions-139853.html>)? Beschreiben Sie zu jedem Konzept kurz die Umsetzung.
- c) Wie würde man mit einem `AssertionError` umgehen, der im Fehlerfall ausgelöst wird? Würde man ihn abfangen und wenn dann wo (direkt nach der Assertion oder an der Stelle die die Methode aufruft, in der sich die Assertion befindet)? Begründen Sie ihre Meinung.
- d) Implementieren Sie für den Entwurf auf der nächsten Seite des Übungsblatts die Methoden
 - a. `Veranstaltung.addStudent(Student s)`,
 - b. `Veranstaltung.removeStudent(Student s)` und
 - c. `Veranstaltungsangebot.eintragen(Veranstaltung v)`.

Garantieren Sie dabei durch die Verwendung geeigneter `assert`-Ausdrücke, dass:

- a. ein Student nicht mehrfach als Teilnehmer einer Veranstaltung eingetragen sein kann,
 - b. ein Student sich nur von einer Veranstaltung abmelden kann, für die er eingetragen ist,
 - c. der Titel einer Veranstaltung im System zur Raumreservierung eindeutig ist (d.h. es darf keine zwei Veranstaltungen mit demselben Titel geben).
- Wie würden Sie Vorbedingungen in Java ohne Assertions realisieren? Nennen Sie zwei unterschiedliche Herangehensweisen. Beschreiben Sie die Unterschiede mit Beispiel-Code unter Abwandlung eines der Codestücke aus der vorherigen Teilaufgabe.
 - In Teilaufgabe e) haben Sie festgestellt, dass es sehr einfach ist DBC in Java zu realisieren, selbst ohne spezielle Sprachunterstützung. Warum glauben Sie, haben sich die Entwickler von Java trotzdem dazu entschlossen, die Sprache um Assertions zu erweitern?

