

Übungen zur Vorlesung Softwaretechnologie

-Wintersemester 2008/2009-
Dr. Günter Kniesel, Daniel Morales

Übungsblatt 10

Abgabetermin am 11. Januar, 23:59 Uhr

Bitte fangen Sie **frühzeitig** mit der Bearbeitung des Zettels an, damit wir Ihnen bei Bedarf helfen können!!!! Lösen Sie die Aufgaben mit Together 2007 in UML2.0, wenn nichts anderes angegeben ist. Fragen zu Übungsaufgaben respektive zur Vorlesung können Sie auf der Mailingliste swt-tutoren@iai.uni-bonn.de, bzw. swt-vorlesung@iai.uni-bonn.de stellen.

Aufgabe 1. *Design by Contract* (17 Punkte)

- a) (4 Punkte) Erläutern Sie die drei Kernkonzepte von „Design by Contract“.
- b) (3 Punkte) Wie können diese Konzepte in Java durch die Verwendung von assert()-Anweisungen umgesetzt werden (siehe <http://java.sun.com/developer/technicalArticles/JavaLP/assertions/>)? Beschreiben Sie zu jedem kurz die allgemeine Vorgehensweise. Hier ein Auszug aus der Dokumentation von Assertions:

An assertion has a Boolean expression that, if evaluated as false, indicates a bug in the code. This mechanism provides a way to detect when a program starts falling into an inconsistent state. Assertions are excellent for documenting assumptions and invariants about a class. Here is a simple example of assertion:

```
BankAccount acct = null;  
  
// ... Get a BankAccount object ...  
  
// Check to ensure we have one  
assert acct != null;
```

This asserts that acct is not null. If acct is null, an AssertionError is thrown. Any line that executes after the assert statement can safely assume that acct is not null. Using assertions helps developers write code that is more correct, more readable, and easier to maintain. Thus, assertions improve the odds that the behavior of a class matches the expectations of its clients.

- c) (3 Punkte) 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) (3 Punkte) Implementieren Sie dementsprechend für den Entwurf auf Seite 4 Methoden, die sicherstellen, dass:
- ein Student nicht mehrfach als Teilnehmer einer Veranstaltung eingetragen sein kann,
 - ein Student sich nur von einer Veranstaltung abmelden kann, für die er eingetragen ist,
 - der Titel einer Veranstaltung im System zur Raumreservierung eindeutig ist (d.h. es darf keine zwei Veranstaltungen mit demselben Titel geben).
- e) (1+2 Punkte) 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.
- f) (2 Punkte) 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?

Aufgabe 2. *Split Objects* (9 Punkte)

Ein Mobilfunkanbieter hat ein Tarifmodell, das durch die Klassen in dem Archiv *PlanModel.zip* spezifiziert ist (zu beziehen über die Webseite der Vorlesung). Die Klassen enthalten Operationen zur Abfrage der Kosten und der Beschreibung eines Tarifs.

- a) (5 Punkte) Es soll nun möglich sein, zu jedem der obigen (Grund-)Tarife folgende Tarif-Optionen hinzu zu buchen:
- Weekend Option mit Kostenpunkt 5.00 €
 - FreeZone Option mit Kostenpunkt 5.00 €
 - BestFriends Option mit Kostenpunkt 5.00 €
 - DataFlat Option mit Kostenpunkt 10.00 €

Fügen Sie Klassen für die Tarif-Optionen dem existierenden Modell hinzu. Die Klassen sollen Operationen erhalten, die die Kosten und eine sinnvolle Beschreibung liefern. Erweitern Sie das Modell (mit Hilfe eines Entwurfsmusters) so, das folgende Anforderungen erfüllt sind:

- Jeder Tarif soll mit beliebig vielen der obigen Optionen kombinierbar sein.
- Ein (Grund-)Tarif und eine Option bilden selbst wieder einen Tarif.
- Die Berechnung der Kosten wird abhängig von dem gewählten Grundtarif und der Optionen durchgeführt. Gleiches gilt für die Beschreibung.

- Das spätere Hinzufügen von Tarifen und Tarif-Optionen soll möglich sein, ohne Anpassungen an bereits existierenden Code vornehmen zu müssen.
 - Es ist nicht notwendig auf bestimmte Komponenten des Tarifs zugreifen zu können, das Endergebnis (Gesamtbeschreibung aller kombinierten Tarife und Gesamtkosten laut Tarif) ist ausreichend. Z. B.:
Standard Plan, DataFlat Option, Weekend Option, BestFriends Option: 29.9€
 - Die korrekte Kombination eines Tarifs, d.h. genau ein Grundtarif kombiniert mit mehreren (nicht doppelt vorhandenen) Optionen, muss nicht von Ihnen gewährleistet werden.
- b) (2 Punkte) Schreiben Sie ein Programm, das einen Grundtarif mit einigen Optionen kombiniert und geben Sie dann die Kosten und die Beschreibung des resultierenden Tarifs auf die Konsole aus.
- c) (2 Punkte) Gehen Sie nun davon aus, dass jeder Grundtarif mit nur einer einzigen Tarif-Option kombiniert werden darf (ansonsten gelten die Anforderungen aus Teilaufgabe a). Würde sich dadurch ein anderes Entwurfsmuster anbieten, um die Aufgabenstellung zu lösen? Begründen Sie stichpunktartig Ihre Antwort.

