

## Übungen zur Vorlesung

# Softwaretechnologie

- Wintersemester 2010/11 -

Dr. Günter Kniessel

## Übungsblatt 7

Zu bearbeiten bis: 05.12.2010

Bitte fangen Sie frühzeitig mit der Bearbeitung an, damit wir Ihnen bei Bedarf helfen können. Checken Sie die Lösungen zu den Aufgaben in Ihr SVN-Repository ein, Diagramme als VP-Dateien, Texte als Textdatei. Fragen zu Übungsaufgaben/Vorlesung können Sie auf der Mailingliste [swt-tutoren@lists.iai.uni-bonn.de](mailto:swt-tutoren@lists.iai.uni-bonn.de), bzw. [swt-vorlesung@lists.iai.uni-bonn.de](mailto:swt-vorlesung@lists.iai.uni-bonn.de) stellen.

### Aufgabe 1. Entwurfsmuster (12 Punkte)

Gegeben sei das Bibliotheksverwaltungs-Projekt, das bereits im letzten Übungsblatt vorgestellt wurde. Der Programmcode steht im SVN-Repository, im Ordner „share“, als „Bibliothek.zip“ zur Verfügung.

Das existierende System sieht für Ausleihgegenstände nur eine Klasse vor. Diese hat ein Feld, welches den Typ des Mediums angibt. Diese Entscheidung ist sinnvoll, wenn man davon ausgeht, dass sich die Medien vom Verhalten nicht sehr unterscheiden.

Unser System sollen aber in Zukunft an Dritte verkauft werden und ihnen die Möglichkeit bieten, in ihre Variante der Bibliothek neue Medienarten mit eigenen Verhalten zu integrieren, ohne die von uns erstellten Klassen ändern zu müssen. Beispielsweise sollen Videos innerhalb einer Ausleihe dem Benutzer online abgespielt werden können.

Das bisherige Design soll daher nun modifiziert werden.

- Welches Entwurfsmuster bietet sich an, um beliebige Medien-Objekte mit dem statischen Typ *Ausleihgegenstand* erzeugen, ohne dass der konkrete Typ der möglichen Objekte schon bei der Entwicklung der Bibliothek bekannt sind?
- Welche Rollen gibt es in diesem Entwurfsmuster? Wie können Sie diese Rolle bestehenden oder noch zu entwickelnden Klassen des Projektes zuordnen?
- Implementieren Sie das Entwurfsmuster im Projekt. Entwerfen Sie dabei Klassen für *Video*, *Buch* und *Zeitschrift*, die von der Klasse *Ausleihgegenstand* erben und integrieren Sie diese Klassen in Ihr Projekt (ohne Änderungen an *Ausleihgegenstand* und nur minimale Änderung<sup>1</sup> an *Bibliothek*).
- Erstellen Sie ein neues Java-Projekt „Unibibliothek“ und fügen Sie das Projekt „Bibliothek“ in den Build-Path des neuen Projektes ein. Entwickeln Sie hier nun mit minimalem Aufwand auf Basis von Aufgabenteil (c) eine *Unibibliothek*, die zusätzlich die Typen *Seminararbeit* und *Uralter Wälzer* kennt.

---

<sup>1</sup> Z.B. ein neues Feld und einen modifizierten Methodenaufruf.

## Aufgabe 2. Entwurfsmuster (12 Punkte)

Passend zur Vorweihnachtszeit hatte ein großer Discounter in der vergangenen Woche verschiedene Funksteckdosen unterschiedlicher Zulieferer im Angebot: Einfache Dosen, dimmbare Dosen und wetterfeste Dosen. Alle Steckdosen können mit einer Fernbedienung gesteuert werden.



Die Fernbedienung enthält unter Anderem vier frei belegbare Reihen von An-/Aus-Knöpfen, sowie eine Reihe „Master An / Master Aus“, mit der alle angeschlossenen Steckdosen ein- oder ausgeschaltet werden.

Leider konnte die Steuerungs-Entwicklung nicht rechtzeitig abgeschlossen werden. Den bisherigen Code finden Sie im SVN-Repository, im Ordner „share“ als „Fernbedienung.zip“.

- a) Mit welchem Entwurfsmuster ließe sich alle folgenden Funktionalitäten der Fernbedienung realisieren:
  - a. Die An/Aus Knöpfe sollen *paarweise* einem beliebigen Gerät zugeordnet werden können.
  - b. Alle Dosenarten sollen unterstützt werden, ohne die Klassen zu verändern, welche die einzelnen Gerätearten darstellen.
  - c. Neue Geräte sollen später hinzugefügt werden können, ohne existierende Klassen verändern zu müssen.
- b) Implementieren Sie die benötigten Klassen und realisieren sie das Entwurfsmuster in der gegebenen Steuerungs-Software der Fernbedienung.
- c) Schreiben Sie einen Programm, das jeder Tasten-Reihe ein Gerät zuordnet, anschließend alle Steckdosen nacheinander einmal ein und ausschaltet und dann die Master-Funktion zum Ein- und Ausschalten verwendet. Kompletieren Sie dazu die Klasse im Paket *client*.
- d) Kopieren Sie das Programm aus Aufgabenteil (c) und vereinfachen Sie es, indem Sie das *Abstract Factory* Pattern bei der Belegung der Knöpfe einsetzen. Sie dürfen dazu auch die Geräte-Klassen erweitern, da üblicherweise Zulieferer bestimmte Spezifikationen des Auftraggebers einhalten müssen.

### **Aufgabe 3.** *Entwurfsmuster* (6 Punkte)

Für die Verwaltung der Termine, die im Rahmen des Bildungsstreiks anstehen, hat die Bonner Bildungsstreikbewegung mit der Entwicklung einer „Appointment-Anwendung“ begonnen. Das Programm soll Appointments (Termine), die aus einer Beschreibung und einem Datum bestehen, verwalten. Eine erste Version dieses Programms finden Sie im SVN-Repository im Ordner „share“ als „AppointmentManager.zip“.

Inzwischen haben sich so viele Termine angesammelt, dass auch eine Gruppierung der Termine unterstützt werden soll.

- a) Überarbeiten Sie das Programm mit Hilfe eines Entwurfsmusters. Es soll
  - a. möglichst wenig an den bestehenden Klassen geändert werden. Vor allem soll die Funktionalität der Klasse *AppointmentItem* erhalten bleiben.
  - b. die Möglichkeit bestehen, Gruppen, die wiederum (Unter-)Gruppen oder natürlich auch *Appointment*-Einträge enthalten, einzufügen.
  - c. möglich sein, überall Gruppen anzutreffen, wo *Appointment*-Einträge erwartet werden und umgekehrt. Insbesondere muss eine Gruppe dieselben Methoden unterstützen, die auch ein einfacher Eintrag bietet.
  
- b) In der Klasse *AppointmentManagerTest* findet sich ein erstes Programm, das ein paar Beispieleinträge erzeugt und dann auf die Konsole ausgibt. Stellen Sie sicher, dass nach Ihrer Anpassung des Modells die Ausgabe noch funktioniert. Kopieren Sie die Klasse und passen Sie die Kopie so an, dass die Einträge in Gruppen geordnet werden. Es sollen zusätzlich zu den einfachen Einträgen auch die Gruppeneinträge und ihre Unterelemente ausgegeben werden, ohne dass der entsprechende *println*-Aufruf oder die *AppointmentManager*-Klasse geändert werden.