

Übungen zur Vorlesung Softwaretechnologie

- Wintersemester 2012/13 -

Dr. Günter Kniesel

Übungsblatt 6

Zu bearbeiten bis: 25.11.2012

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, „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. *Proxy-Pattern* (2+2+1 Punkte)

- Erläutern Sie kurz die Motivation und die Funktionsweise des Proxy-Entwurfsmusters.
- Welches sind die teilnehmenden Rollen (Klassen) des Proxy-Entwurfsmusters? Welche Aufgaben übernehmen diese? Welche Operationen sind notwendig?
- Nennen Sie mindestens drei Beispiele für die Anwendung eines Proxy-Entwurfsmusters und erläutern Sie kurz die Motivation für den jeweiligen Pattern-Einsatz.

Aufgabe 2. *Umsetzung von Modellen* (2+8+2+3 Punkte)

In dieser Aufgabe stellen wir Ihnen ein Klassendiagramm und Sequenzdiagramm zur Verfügung, das einen Taschenrechner mit Plugin-Fähigkeiten modelliert. Ihre Aufgabe liegt darin das gegebene Modell in Java umzusetzen.

Die benötigten Diagramme und Java-Dateien finden Sie unter:

https://svn.iai.uni-bonn.de/repos/IAI_Software/se/swt2012/gruppe/readonly

Die blau eingefärbten Elemente des Klassendiagrammes geben wir Ihnen vor. Sie finden die Dateien ebenfalls im readonly-Ordner.

- Analysieren Sie das gegebene Klassendiagramm. Identifizieren Sie welche Entwurfsmuster in diesem Modell verwendet wurden.
- Setzen Sie das Modell in Java um. Verwenden Sie das Klassendiagramm um die Struktur, das Sequenzdiagramm um den Ablauf einer Rechenaktion zu verstehen. Beachten Sie zusätzlich die folgenden Informationen:
 - Alle Operationen (Add, Sqrt,...) sollen als .class Dateien vom OperatorLoader geladen werden, und sind nicht Teil des Rechners.

- Verwenden Sie die gegebenen *Add.class* und *PrimeDecomposition.class* Dateien, um ihre Implementierung zu testen. (Token: Add (+), PrimeDecomposition (*prime*))
 - Der Einfachheit halber können sie die Präfix Notation für Eingaben verwenden (z.B.: „+ 2 3“ oder „prime 18“).
 - Ergänzen Sie eigene Methoden, wenn es ihnen notwendig erscheint. Achten Sie in diesem Fall darauf, dass Sie entsprechend das Modell konsistent halten.
- c) Ergänzen Sie eine neue Operation, die ebenfalls als Class-Datei vom OperatorLoader geladen werden kann.
- d) Erklären Sie wie Sie die verschiedenen Elemente des Klassendiagramms in Code umgesetzt haben.

Aufgabe 3. *Entwurfsmuster* (13 Punkte)

Gegeben sei der Kern einer Bibliotheksverwaltung, deren Java-Programme Sie ebenfalls im SVN-Repository im Ordner `readonly` finden:

https://svn.iai.uni-bonn.de/repos/IAI_Software/se/swt2012/gruppe/readonly

Ziel ist es nun, dieses Minimal-System an geeigneter Stelle mit Hilfe von Entwurfsmustern zu erweitern.

Hinweis: Der mitgelieferte Programmcode enthält im Paket `client` die Klasse `BibliotheksTest`. Benutzen Sie diese als Hauptprogramm, um Ihre Implementierung zu testen.

- a) In unserem System führt jedes Objekt der Klasse `Ausleihgegenstand` eine Warteliste, auf der Kunden sich bei Interesse für ein aktuell entliehenes Medium eintragen können. Realisieren Sie mittels eines geeigneten Musters, dass der erste Kunde aus der Warteliste eines Mediums informiert wird, sobald dieses Medium zurückgegeben wurde. Als Reaktion auf die Benachrichtigung soll das Kunden-Objekt eine E-Mail an den entsprechenden Kunden schicken (symbolisiert durch die Text-Ausgabe „Schicke E-Mail an ...Name...: ... Inhalt...“).
- b) In einem nächsten Schritt soll das E-Mail-System der Bibliothek erweitert werden. Erstellen Sie dazu eine Klasse `EmailSystem`, die das E-Mail-System repräsentiert und eine Methode `sendeMail(String name, String nachricht)` unterstützt (die eine Zeichenkette auf `System.out` ausgibt). Garantieren Sie durch die korrekte Anwendung eines geeigneten Entwurfsmusters, das maximal eine Instanz von `EmailSystem` erzeugt und verwendet wird. Binden Sie zuletzt die neue Klasse und ihre Funktionalität in das Gesamtsystem ein.