



Universität  
Zürich<sup>UZH</sup>

Institut für Informatik



# Software Engineering Übung 4

Architektur, Modulentwurf

## 1 Informationen

### 1.1 Daten

- Ausgabe Di 01.11.2011
- Abgabe So 13.11.2011 bis 23:59 Uhr
- Besprechung am Di 22.11.2011 um 11:45 Uhr

### 1.2 Formales

Die Lösungen sollen als PDF Datei mit dem Namen **Ex[n]\_[NameA\_NameB\_NameC].pdf** abgegeben werden, wobei [n] die Nummer der Übung ist und [NameA\_NameB\_NameC] die Nachnamen der Gruppenmitglieder sind. Die PDF Datei sollte ausserdem ebenfalls Ihre Namen und Matrikelnummern beinhalten.

Mailen sie Ihre Lösungen vor dem Abgabetermin an [todoran@ifi.uzh.ch](mailto:todoran@ifi.uzh.ch) und [nhoby@access.uzh.ch](mailto:nhoby@access.uzh.ch). Der Betreff der E-mail sollte mit **[SE EX HS11]** beginnen. Falls Sie zusätzliche Abgabematerialien (z.B. Source Code) haben, mailen Sie bitte ein Archiv (.zip-File), welches alle Dateien, einschliesslich dem PDF, enthält. Benennen sie das Archiv anhand der oben erwähnten Konventionen. Als Abgabedateien sind nur PDF-Dokumente und Source Code Dateien erlaubt, keine Bilder.

Die Übungen sollen in 3er Gruppen gelöst werden. Jedes Gruppenmitglied muss über alle Teile der Lösungen Auskunft geben können. Verspätete Abgaben werden korrigiert, aber nicht bewertet.

## 2 Aufgabenstellung

Die folgende Übung bezieht sich auf die Fallstudie, welche mit Übung 2 ausgegeben wurde. Sie finden sie auch separat im File *casestudy.pdf*.

Möglicherweise verfügen Sie noch nicht über genügend Informationen und müssen an manchen Stellen vorläufige Annahmen treffen, da die Spezifikation der *iCommute'n'Smile* einige Fragen offen lässt (welche Sie in einem realen Projekt später mit den beteiligten Interesseneignern diskutieren und klären würden). Damit Fakten und Annahmen klar unterscheidbar sind, setzen Sie alle Texte oder Modellteile, die auf Annahmen basieren, in eckige Klammern.

## 2.1 Architekturstile (6 Punkte)

Die folgenden Abschnitte beschreiben wesentliche Anforderungen des *iCommute'n'Smile* Ticket-systems. Ihre Aufgabe ist es, zu jedem der zwei Szenarien einen passenden Architekturstil zu wählen. Begründen Sie Ihre Auswahl, indem Sie erklären, wie der Architekturstil die genannten Anforderungen adressiert. Falls etwas unklar ist, treffen Sie Annahmen und markieren diese. Antworten ohne Begründung werden nicht bewertet.

### 2.1.A Ausgabe eines Monatsabos an Angestellte (3 Punkte)

Arbeitgeber können für ihre Angestellten Rabattgutscheine kaufen. Wenn ein Angestellter mit einem gültigen Rabattcode ein Monatsabo kaufen will, benutzt er/sie dazu entweder einen Selbstbedienungsautomaten oder das Webinterface von *iCommute'n'Smile*. Zuerst gibt er/sie alle relevanten Daten ein: Den Namen, die AHV-Nummer und den Rabattcode. Die Preisberechnungskomponente berechnet den Preis aus den im Code gespeicherten Daten (Rabatt und Gültigkeitsdauer des Gutscheins). Eine weitere Komponente verzeichnet den Namen und weitere Daten auf dem Ticket. Das Ticket wird auf Papier gedruckt, sollte der Vorgang an einem Automaten vollzogen werden, oder als PDF-Datei zugestellt, wenn man es via Webinterface bestellt.

### 2.1.B Abschliessen der Matura und Immatrikulation an einer Hochschule (3 Punkte)

Wenn jemand die Matura abschliesst, speichert das Gymnasium seine/ihre Daten, macht den Schülerausweis ungültig und aktualisiert den Status der betroffenen Person in der Datenbank (z.B. *nicht mehr rabattberechtigt*). Sobald er/sie sich an einer Universität immatrikuliert, löst dies die Ausstellung eines neuen Studentenausweises sowie einen Statuswechsel in der Datenbank aus. Ein allfälliger Adresswechsel wird ebenfalls notiert.

## 2.2 Entwurfsmuster (Design Patterns) (9 Punkte)

Um diese Aufgabe zu lösen, benötigen Sie eventuell zusätzliches Material über Entwurfsmuster. Einige Kopien von [1] sind zur Konsultation in unserer Forschungsgruppe erhältlich (schauen Sie einfach bei Irina Todoran, BIN 2.B.20, oder Nicolas Hoby, BIN 2.B.17, vorbei). Sie können auch auf dem Web viele Informationen zu Entwurfsmustern finden.

Die folgenden Teilaufgaben behandeln einige der 23 Entwurfsmuster von [1]. Das Dokument *design\_patterns.pdf* gibt einen Überblick über diese Entwurfsmuster.

### 2.2.A Strukturelle Muster (2 Punkte)

Das geplante System soll mit Datenbanken verschiedener Universitäten kommunizieren können. Die dabei ausgetauschten Daten sind grundsätzlich immer die gleichen, aber die Schnittstelle zur jeweiligen Datenbank kann variieren. Das *iCommute'n'Smile* Systems soll aber mit beliebigen fremden Datenbanken kommunizieren können. Welches Strukturpattern kommt hier zum Einsatz? Weshalb wird es verwendet? Begründen Sie Ihre Antwort.

### 2.2.B Erzeugungsmuster (4 Punkte)

Bei der Ausgabe von Monatsabos werden je nach Abotyp unterschiedliche Daten benötigt, z.B. die AHV-Nummer bei ermässigten Abos für Angestellte. Jeder Abotyp besitze einen internen, eindeutigen Identifier (z.B. ein `enum`), der bei der Erzeugung übergeben wird. Welches Erzeugungsmuster schlagen Sie für diese Komponente vor? Begründen Sie Ihre Antwort. Zeichnen Sie ein Klassendiagramm Ihrer Lösung und implementieren Sie einen kleinen Prototypen davon. Erstellen Sie die beteiligten Klassen und definieren Sie die wichtigsten Methoden, Parameter und Relationen. Nur ein Protoyp ist verlangt, Sie brauchen keine Funktionalität zu implementieren.

### 2.2.C Verhaltensmuster (3 Punkte)

Die Erstellung eines Tickets folgt einem vorgegebenen Ablauf. Zuerst werden die jeweils benötigten Daten der gewünschten Ticketart abgefragt. Anschliessend werden diese Daten validiert. Als dritter Schritt, und dieser ist bei allen Tickets gleich, wird das Ticket ausgedruckt (auf Papier oder als PDF-Dokument). Welches Design Pattern würden Sie vorschlagen, um sicherzugehen, dass dieser Ablauf für alle möglichen Ticketarten eingehalten wird? Begründen Sie Ihre Antwort. Erstellen Sie die beteiligten Klassen und definieren Sie die wichtigsten Methoden, Parameter und Relationen. Nur ein Protoyp ist verlangt, Sie brauchen keine Funktionalität zu implementieren.

### 2.3 Entwurfsmuster in Java API (5 Punkte)

Durchsuchen Sie die Java API und finden Sie drei Stellen, an denen Entwurfsmuster verwendet werden. Für jede Ihrer Entdeckungen, nennen Sie:

- Den Namen des Musters und seine Kategorie.
- Die Liste der Beteiligten in diesem Muster.
- Die Java Elemente (Klassen, Interfaces, Methoden), welche an der Kollaboration beteiligt sind (und deren Rollen in dem Muster).

In den von Ihnen genannten Mustern sollte jede der drei Kategorien - Erzeugungsmuster, Verhaltensmuster und Strukturmuster - vertreten sein.

*Tipp:* Die Packages `javax.swing` und `java.util` benutzen viele Entwurfsmuster!

## Literatur

- [1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.