Universität Zürich
Institut für Informatik

s.e.a.l. software evolution & architecture lab
RERG

Software Engineering Übung 4

Architektur, Modulentwurf

1 Informationen

1.1 Daten

- Ausgabe Di 27.10.2009
- Abgabe So 08.11.2009 bis 23:59 Uhr
- Besprechung am Di 17.11.2009

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 `wueest@ifi.uzh.ch`. Der Betreff der E-mail sollte mit **[SE EX HS09]** 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.

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

Diese Übung bezieht sich auf die Fallstudie *Ambidramas*, welche mit der Übung 2 ausgegeben wurde.

2.1 Architekturstile (7.5 Punkte)

Im folgenden beschreibt die Theaterleitung einige Anforderungen an das neue System. Ihre Aufgabe ist es, für jeden der drei Fälle jeweils einen adäquaten Architekturstil auszuwählen. Begründen Sie Ihre Auswahl, indem Sie erklären, wie der Architekturstil die genannten Anforderungen adressiert. Beschreiben und illustrieren Sie mit einem UML Diagramm, was die Komponenten, die Konnektoren und die Einschränkungen der von Ihnen gewählten, spezifischen Konfiguration sind. Nennen Sie ausserdem einen Nachteil von Ihrer Wahl.

2.1.A Infrastruktur (2.5 Punkte)

Wie in der Übung 2 beschrieben, wird das neue System an das alte Platzreservationssystem angebunden. Die Kunden werden in Zukunft über die Homepage des Theaters mit dem neuen System kommunizieren und Sitzplätze reservieren oder kaufen. Das neue System beinhaltet die Anwendungslogik (es verarbeitet zum Beispiel die Kundenanfragen). Das alte Platzreservationssystem läuft auf einem Server im Theatergebäude und beinhaltet die Datenbank, in der die Zustände der Sitzplätze gespeichert werden.

2.1.B Theater-News (2.5 Punkte)

Kunden können sich auf der Homepage für einen persönlichen Newsletter registrieren. Dabei können sie aus einer Liste diejenigen Kategorien von Nachrichten auswählen, für die sie sich Interessieren. Wenn Neuigkeiten aus einer bestimmten Kategorie veröffentlicht werden, sollen nun alle Kunden, welche sich für diese Kategorie angemeldet haben, automatisch eine E-mail erhalten, welche die News enthält.

2.1.C Vorstellungsinformationen (2.5 Punkte)

Die Informationen, welche zu den einzelnen Vorstellungen von der Theaterleitung und den Tournee-Veranstaltern eingegeben werden, erscheinen an mehreren Orten.

Unter anderem wird eine Liste aller eingetragenen Vorstellungen benutzt, um automatisch die persönlichen Newsletter zu erstellen (die Definition des persönlichen Newsletters finden Sie in Aufgabe 2.1.B). Die Liste soll entweder nach Datum oder Namen der Vorstellungen sortiert werden können. Für jeden persönlichen Newsletter werden diejenigen Veranstaltungen herausgefiltert, welche für den jeweiligen Kunde von Interesse sind. Die Informationen zu diesen Veranstaltungen werden dann in einem PDF oder HTML Dokument gespeichert und in die Newsletter-E-mail eingefügt.

2.2 Entwurfsmuster (Design Patterns) (12.5 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 im Assistentenbüro, 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 Verschiedene Arten von Entwurfsmustern (1 Punkt)

In den folgenden Beispielen werden Entwurfsmuster aus den Kategorien Erzeugungsmuster, Strukturmuster und Verhaltensmuster angesprochen. Erklären Sie, was der Unterschied zwischen diesen Kategorien ist, bzw. welchem Zweck jede Kategorie von Entwurfsmustern jeweils dient.

2.2.B Mehrere Typen von Tickets (4 Punkte)

Ein Kino in der Nähe hat von dem geplanten Reservationssystem gehört und möchte dieses ebenfalls benutzen. Die Kinotickets unterscheiden sich jedoch von den Theatertickets. Aus diesem Grund wurden mehrere Subklassen von *Ticket* erstellt.

Ticket-Objekte werden an vielen Stellen in der Applikation instanziert und verwendet. Abbildung 1 zeigt eine Klasse, welche mit Theatertickets arbeitet.

```java
/**
 * Test application, creates and prints tickets for the theatre and cinema.
 *
 * @author      Dustin Wueest
 * @history     21.10.09 DW first version
 * @version     21.10.09 1.0
 * @responsibilities creates and prints tickets
 */
public class TicketApp1 {

    public static void main(String[] args) {
        TicketApp1 app = new TicketApp1();
        app.methodOne();
    }

    public void methodOne() {
        Ticket ti = new TheatreTicket();
        ti.printMe();
    }

    public void methodTwo( String name ) {
        Ticket ti = new TheatreTicket();
        ti.setPerson( name );
        ti.save();
    }

    public void methodThree() {
        Ticket ti = new TheatreTicket();
        ti.save();
    }
}
```

Abbildung 1: TicketApp1.java instanziert und verwendet Tickets

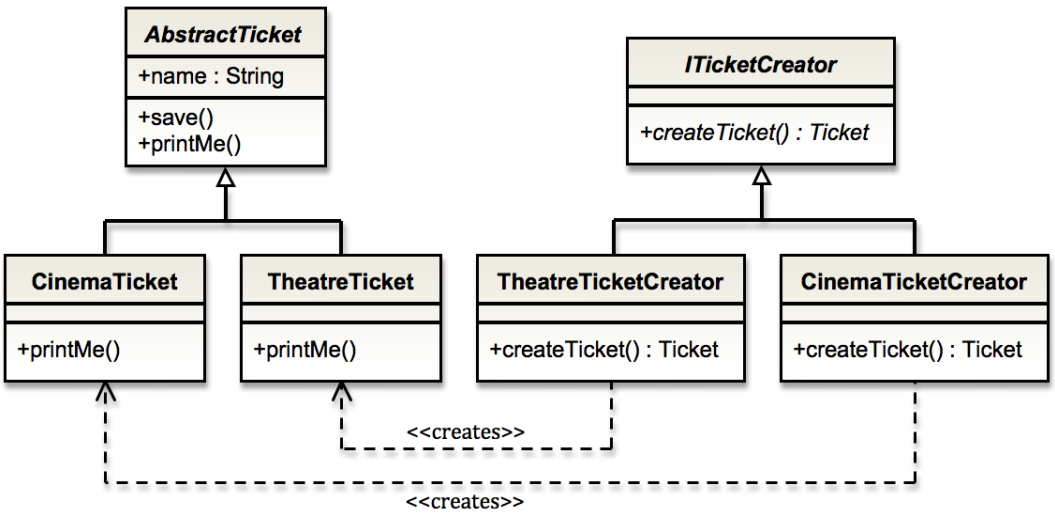

Abbildung 2: Klassendiagramm für die Ticketerstellung

Die Klasse *TicketApp1* soll nun aber nicht nur Tickets vom Typ *TheatreTicket* bearbeiten können, sondern auch Tickets vom Typ *CinemaTicket*. Später könnten noch weitere Ticketarten dazukommen. Dies ist momentan nur schwer zu realisieren, weil die Klasse *TicketApp1* mit konkreten Typen von Tickets arbeitet. Abhilfe schafft hier das in Abbildung 2 dargestellte Entwurfsmuster.

a) Welches Entwurfsmuster wird hier verwendet?
b) Welche Rollen spielen die jeweiligen Klassen, welche am Entwurfsmuster beteiligt sind?
c) Implementieren Sie das gewählte Entwurfsmuster. Benutzen Sie dazu das Projekt *TicketSystem*, welches sich im zip-file *Code4U.zip* befindet. Im bisherigen Code müssten drei Zuweisungen geändert werden, damit die Klasse *TicketApp1* mit einem anderen Ticket-Typ arbeitet. Ihre Implementierung soll es dagegen ermöglichen, dass maximal eine Zuweisung geändert werden muss, wenn ein anderer Ticket-Typ verwendet werden soll.

2.2.C Der einsame ObjectCreator (2.5 Punkte)

Sehen Sie Sich die Java-Klasse in Abbildung 3 an.

```
public class ObjectCreator {

    private static ObjectCreator instance;

    //The only Constructor of this class.
    private ObjectCreator() {
    }

    public static ObjectCreator getInstance() {
        if( ObjectCreator.instance == null ) {
            ObjectCreator.instance = new ObjectCreator();
        }
        return ObjectCreator.instance;
    }
}
```

Abbildung 3: ObjectCreator Klasse

a) Welches Entwurfsmuster wurde hier implementiert?
b) Welche Folgen hat diese Implementierung von *ObjectCreator* auf die Instanzierung von Objekten dieser Klasse?
c) Sehen Sie eine Möglichkeit, dieses Entwurfsmuster auf eine Klasse aus dem UML Klassendiagrammen anzuwenden, welches in der Aufgabe 2.2.B skizziert ist (Abbildung 2)? Wenn ja: in welcher Klasse macht es Sinn (begründen Sie kurz in 1-2 Sätzen)? Wenn nein, warum macht es keinen Sinn (begründen Sie kurz in 1-2 Sätzen)?

2.2.D Inkompatible Sitzplätze (2 Punkte)

Im Gegensatz zu den normalen Theaterplätzen sind die Plätze in der Lounge nicht nummeriert, sondern besitzen Buchstaben. Sie wurden angewiesen, die Klasse *LoungeChair* aus dem alten System wiederzuverwenden. Diese Klasse ist Teil eines Frameworks und kann nicht verändert werden. Ihr eigenes Sitzplatz-Interface hält fest, dass jede Sitzplatz-Subklasse eine Methode *getId()* besitzen muss, welche die Nummer oder den Buchstaben des Sitzplatzes als String zurück liefert.

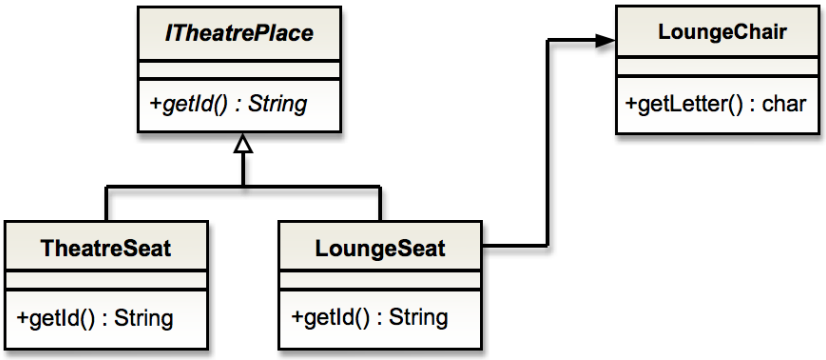

Abbildung 4: Sitzplatz Klassendiagramm

Die Klasse *LoungeChair* hat jedoch nur eine Methode *getLetter()* und liefert den Buchstaben des Platzes als Character zurück. Sie wollen nun die verschiedenen Interfaces miteinander kompatibel machen, indem Sie sie über eine neue Klasse namens *LoungeSeat* verbinden (wie in Abbildung 4 dargestellt). Diese Klasse enthält ein Objekt der Klasse *LoungeChair* als Attribut.

a) Welches Entwurfsmuster verwenden Sie hier?
b) Beschreiben Sie die Hauptaufgabe der Klasse *LoungeSeat*, indem Sie angeben, was im Konstruktor oder in der Methode *getId()* passiert.

2.2.E Automatische Platzzuordnung (3 Punkte)

In einem alternativen Szenario können die Theaterbesucher lediglich die Sitzplatz-Kategorie wählen, aber nicht den konkreten Sitzplatz. Ein Algorithmus entscheidet, in welcher Reihenfolge die Sitzplätze einer Kategorie den Besuchern zugeordnet werden. Der Theaterleitung fallen hierzu mehrere mögliche Algorithmen ein, welche untereinander austauschbar sein sollen.

a) Welches Entwurfsmuster ermöglicht die Austauschbarkeit der Algorithmen?
b) Beschreiben Sie in 2-3 Sätzen, wie es dank dem gewählten Entwurfsmuster möglich ist, verschiedene Algorithmen zu verwenden.
c) Zeichnen Sie ein UML Klassendiagramm, welches das von Ihnen gewählte Entwurfsmuster im konkreten Beispiel beschreibt (d.h. benutzen Sie Klassennamen, so wie Sie die betroffenen Klassen in der Theater-Software nennen würden).

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.