

## Hauptaufgabe

Entwickeln Sie in Gruppenarbeit einen Prototyp einer Software zur Verwaltung eines Kinos. Eine erste Beschreibung des Anwendungsgebietes finden Sie unten. Wenden Sie dabei den in der Vorlesung vorgestellten, objektorientierten Softwarekonstruktionsprozess an. Liefern Sie Ihre Zwischen- und Endergebnisse in dem in der Vorlesung bekannt gegebenen Umfang rechtzeitig zu den festgelegten Terminen bei Ihren Betreuern ab. Stellen Sie die von Ihnen entwickelte Software und ihr Design in einer Präsentation (ca. 25 Minuten) vor.

### Beschreibung des Anwendungsgebietes

Der *Betreiber* eines Kinos richtet die Software für sein Kino ein, indem er die vorhandenen *Säle* mit ihren Kapazitäten und die von ihm gewünschten *Preisklassen* eingibt.

Er kann nun *Filme* in sein Angebot aufnehmen und ein *Programm* festlegen. Im Programm wird aufgeführt, welche *Filme* zu welchen *Terminen* und *Zeiten* in welchen Sälen des Kinos in einer *Vorstellung* vorgeführt werden.

*Besucher* können sich dieses Programm anschauen und erhalten zu jedem Film noch eine *Kurzinformation*.

Bis zu einer gewissen *Reservierungsfrist* vor Beginn, können nun *Karten* zu Vorstellungen *reserviert* werden. Dabei steht für eine Vorstellung, bedingt durch die Gegebenheiten des Saals und die Art der Vorstellung, nur ein gewisses Kontingent von Karten für eine Preisklasse zur Reservierung zur Verfügung. Reservierungen verfallen nach Ablauf der Frist. Bei erfolgreicher Reservierung erhält der Besucher eine Kennung, um sich an der Kasse zu identifizieren.

Der endgültige *Verkauf* von Karten und die Zuweisung von *Plätzen* erfolgt durch einen *Kassierer*. Dabei kann es sich um reservierte Karten, nach erfolgreicher Identifikation, oder noch freie Karten aus dem Direktverkauf handeln. Der Verkauf von Karten ist bis zum Beginn des Films möglich.

### Anforderungen

#### Allgemeine Anforderungen

- Ihre Software soll in Java geschrieben sein. Verwenden Sie mindestens die Version 1.5 (empfohlen 1.6).
- Ihre Software soll eine Swing-GUI haben. Falls Sie ein anderes GUI-Framework verwenden wollen, halten Sie bitte Rücksprache mit Ihrem Betreuer.
- Die wichtigen Daten sollen zwischen dem Beenden und einem erneuten Start Ihrer Applikation persistent gehalten werden.

## Anwendungsspezifische Anforderungen

Ihre Software muss mindestens die folgende Funktionalität unterstützen:

- Grundeinstellungen für ein Kino eintragen, z.B. vorhandene Säle und Kapazitäten, Reservierungsfrist, Preise, etc. Es muss nur eine Preiskategorie unterstützt werden.
- Filme und zugehörige Kurzinformation, z.B. Inhalt, Länge, etc., erstellen.
- Programm für Vorstellungen von Filmen in Sälen einrichten.
- Anzeige der geplanten Vorstellungen für den Besucher.
- Abgabe von Reservierungen durch Besucher und Vergabe von Identifikationsnummern durch das System, solange noch Kapazitäten zur Verfügung stehen und die Reservierungsfrist noch nicht abgelaufen ist. Eine gezielte Auswahl von Sitzplätzen durch den Benutzer muss nicht unterstützt werden.
- Freigabe von Reservierungen nach Ablauf der Reservierungsfrist.
- Verwaltung der Kassierer.
- Verkauf von Karten durch Kassierer und Zuweisung von Sitzplätzen bis zum Beginn einer Vorstellung. Die Vergabe der Sitzplätze kann automatisch, z.B. in aufsteigender Nummerierung, erfolgen. Der Verkauf von reservierten Karten erfolgt nur nach erfolgreicher Identifikation des Besuchers.

Zusätzlich wäre es schön, wenn sie möglichst viele der folgenden Funktionalitäten unterstützt:

- Unterstützung mehrerer Betreiber, die sich die administrativen Aufgaben teilen.
- Interaktive multimediale Informationen zu Filmen, z.B. Poster, Trailer, Foren, Bewertungssystem, etc.
- Verschiedene Preiskategorien, die für Vorstellungen/Säle möglich sind, z.B. Spartag, Parkett und Loge, Überlängeaufschlag, 3D-Aufschlag, etc.
- Sitzplatzlayout für Säle und gezielte Platzwahl durch Besucher und Kassierer.
- Verschiedene Statistiken, z.B. für Einnahmen, Besucherzahlen, etc.
- Automatische Unterstützung bei Vorstellungsplanung, Saalvergabe nach Angabe erwarteter Besucher und/oder basierend auf Statistiken.
- Weitere Funktionen, die Ihnen sinnvoll erscheinen.

## Einschränkungen

Im Rahmen eines vierwöchigen Softwarepraktikums kann man keine vollständig ausgereifte Software mit “allen Schikanen” entwickeln. Folgende Einschränkungen, die den Entwicklungsaufwand verringern, sollten für den zu entwickelnden Prototyp der Software gemacht werden. Falls Sie eine dieser Einschränkungen nicht machen wollen, halten Sie vorher bitte Rücksprache mit Ihrem Betreuer.

- Die GUI soll ausschließlich mit Java-Swing entworfen werden. Ein zusätzliches Web-Interface, z.B. für die Reservierung von Karten, ist nicht notwendig.
- Die Anwendung soll als Einzelplatzanwendung entworfen werden, d.h. der Zugriff erfolgt von höchstens einem Terminal gleichzeitig.

## Warm-Up Aufgabe

Diese Aufgabe dient dem (wieder-) hineinfinden in die Java-Programmierung. Sie soll in Zweiergruppen bearbeitet werden.

Implementieren Sie ein Adressbuch als Java-Anwendung. Die Anwendung soll Informationen zu Personen verwalten. Ein Eintrag besteht aus Vor- und Nachname, Emailadresse, Telefonnummer, Straße, Hausnummer, PLZ, Ort und Land einer Person. Felder können dabei ggf. leer gelassen werden. Die GUI soll alle Einträge in einer Liste (Tabelle) darstellen, und das Hinzufügen, Entfernen und Ändern von Einträgen erlauben. Die Einträge sollen beim Beenden der Anwendung natürlich nicht verloren gehen. Benutzen Sie einen Serialisierungsmechanismus, um die Daten zu speichern. Dabei genügt es, wenn die Daten beim Starten der Anwendung aus einer festen Datei geladen, und beim Beenden der Anwendung in diese Datei zurückgeschrieben werden. Die GUI könnte zum Beispiel wie in Abbildung ?? dargestellt aussehen.

## Hinweise

- Bei der Erstellung dieser Anwendung darf aus didaktischen Gründen **kein** GUI-Editor verwendet werden.
- Vergessen Sie nicht, dem Fenster einen sinnvollen Titel und eine sinnvolle Anfangsgröße zu geben.

Wenn Sie nicht genau wissen, wie Sie bei der Realisierung der GUI vorgehen sollen, beachten Sie die folgenden Anregungen:

- Leiten Sie die Klasse für Ihr Fenster von *JFrame* ab.

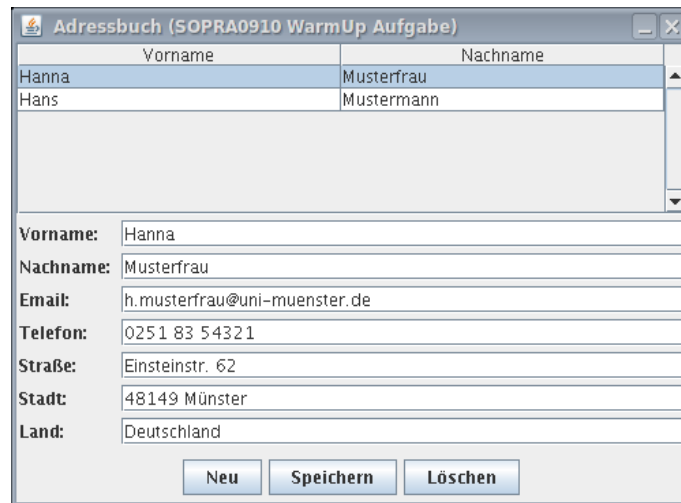


Abbildung 1: Beispiel für Lösung der Warm-Up Aufgabe

- Verwenden Sie *BorderLayout* als Layout-Manager für die *ContentPane* des Fensters.
- Ordnen Sie die Knöpfe in einem *JPanel* an, das Sie mit dem *BorderLayout.SOUTH*-Constraint zur *ContentPane* hinzufügen. Dieses *JPanel* verwendet am besten ein *FlowLayout*.
- Erzeugen Sie auf ähnliche Art und Weise ein *JPanel* mit geeignetem Layout für die Eingabefelder zum Ändern der Daten in der Mitte der *ContentPane* (*BorderLayout.CENTER*).
- Platzieren Sie die Tabelle in einer *JScrollPane* im oberen Bereich der *ContentPane* (*BorderLayout.NORTH*).
- Verwenden Sie *DefaultTableModel* als Basis für das Datenmodell Ihrer Tabelle.
- Um das Speichern beim Beenden der Anwendung zu realisieren, müssen Sie einen *WindowListener* implementieren. Über dessen *WindowClosed* bzw. *WindowClosing* Methoden bekommen Sie mit, wenn das Fenster geschlossen wird. Achten Sie darauf, dass Sie wirklich nur die Adressdaten serialisieren und nicht (ungewollt) auch Teile der Komponentenstruktur der Anwendung. Dies kann passieren, wenn Sie das Datenmodell der Tabelle serialisieren, und die Serialisierung rekursiv auch alle registrierten Listener und alles was daran hängt mitserialisiert.

## Material

Das folgende Material kann bei der Bearbeitung der Warm-Up Aufgabe hilfreich sein:

- Ein gutes Buch zur Java Programmierung, z.B.:
  - *Handbuch der Java-Programmierung*. Guido Krüger, Thomas Stark. Eine HTML-Version kann unter <http://javabuch.de/download.html> heruntergeladen werden. Teil 6 (Kap. 35-38) beschäftigt sich mit Swing.

- *Java ist auch eine Insel. Ullenboom.* Eine Online-Version ist verfügbar unter: <http://www.galileocomputing.de/openbook/javainsel8/>. Kapitel 16 beschäftigt sich mit der GUI-Programmierung mit Swing.
- Java-Swing-Tutorial: <http://java.sun.com/docs/books/tutorial/uiswing/TOC.html>. Hieraus insbesondere die Abschnitte: *Using Swing-Components*, *Laying Out Components Within a Container* und *Writing Event Listener*.
- XStream Two Minute Tutorial: <http://xstream.codehaus.org/tutorial.html>
- Java-API Spezifikation: <http://java.sun.com/javase/6/docs/api>

## Zeitplan

|                           |   |
|---------------------------|---|
| Mo, 01.03.2010, 18:00 Uhr | Endgültige Abgabe des Analysedokuments (Pflichtenheft)  |
| Di, 02.03.2010, 18:00 Uhr | Endgültige Abgabe der Warm-Up Aufgabe   |
| Mi, 10.03.2010, 18:00 Uhr | Erste Abgabe des Entwurfsdokuments und der Implementierung  |
| Mo, 15.03.2010, 18:00 Uhr | Endgültige Abgabe des überarbeiteten Entwurfsdokuments und der Implementierung, zudem endgültige Abgabe des Handbuchs |
| Di, 16.03.2010            | Vorbereitung der Abschlusspräsentation und Einzelvorführung in den Gruppen  |
| Mi, 17.03.2010            | Abschlusspräsentation vor allen Gruppen   |