

# Informationsinfrastrukturen

---

## Einschub: Projektarbeit

Prof. Dr. Jil Klünder

# Formalitäten

- Ihr bearbeitet in Kleingruppen (insgesamt 12 Gruppen, d.h., 4-6 Studierende) ein Maxiprojekt. In den letzten zwei Wochen des Quartals präsentiert ihr eure Ergebnisse.
- Bis Weihnachten gebt ihr eine schriftliche Ausarbeitung ab, in der ihr eure Ergebnisse und die Erkenntnisse zusammenfasst.
- In den 4 Wochen Bearbeitungszeit präsentiert ihr mir einmal eure Zwischenergebnisse. Die Termine könnt ihr dafür frei auswählen (in den Vorlesungsslots). Eine Excel-Datei zum Eintragen werde ich zur Verfügung stellen.
- Die Note für die Präsentation und die Dokumentation ergeben die Note für das Modul (jeweils 50%).
- Beide Teile müssen bestanden sein, um das Modul zu bestehen.
- Die Aufgabenstellung setzt sich zusammen aus einer Basisaufgabe, die pro Gruppe durch Zusatzanforderungen erweitert wird.

# Beschreibung der Situation

Ihr seid Besitzer eines kleinen Kinos. Ihr wollt den elektronischen Verkauf von Eintrittskarten realisieren. In einer Aufbauphase nehmt ihr die Bestuhlung von zwei Kinosälen vor: Jeder Kinosaal besteht aus Sitz-Reihen. Jede Reihe hat eine bestimmte Anzahl an Plätzen, ein Saal muss nicht "rechteckig" sein. Jeder Reihe wird eine der drei Kategorien "Parkett" (niedrigster Sitzplatz-Preis), "Loge" und "Loge mit Service" (höchster Preis) zugeordnet. Ist die konfigurierte Bestuhlung komplett, wird ein Saal für den Verkauf von Eintrittskarten freigegeben. Erst dann dürfen Reservierungen und Buchungen auf den Plätzen vorgenommen werden.

Im laufenden Betrieb des Kinos finden Filmaufführungen statt, denen jeweils ein für den Verkauf freigegebener Kinosaal zugeordnet ist. Ein Film kann in mehreren Aufführungen in verschiedenen Sälen gezeigt werden.

Ein Kunde sucht sich eine Filmaufführung aus, für die Karten in einer von ihm gewünschten Kategorie reserviert werden können. Natürlich kann nicht reserviert werden, wenn keine Plätze mehr vorhanden sind. Reservierungen können wieder storniert werden. In einer späteren Aktion kann ein Kunde dann eine Reservierung in eine Buchung umwandeln. Buchungen können auch direkt ohne vorige Reservierung vorgenommen werden, können aber nicht mehr storniert werden. Aufführungen finden in gegebenen Zeitintervallen statt und können danach natürlich nicht mehr gebucht werden. Als Kinobesitzer möchte man dann wissen, wie viel Einnahmen eine einzelne Aufführung bzw. ein einzelner Film auf Basis der aktuell durchgeführten Buchungen einbringt / eingebracht hat.

Realisiert dieses kleine Reservierungs- und Buchungssystem mit Hilfe von EMF.

# Detaillierte Aufgabenstellung – Teil 1

1. Erstellt Anwendungsfälle mit einer natürlichsprachlichen Beschreibung des Ablaufs.
2. Extrahiert daraus Klassen, Assoziationen, Operationen und Exceptions und modelliert diese in einem UML-(Fach-)Klassen-Modell.
3. Erstellt dieses Modell als Ecore-Datei in EMF, lässt euch daraus Java-Code generieren und reichert den Code so an, dass die Anforderungen korrekt umgesetzt werden.
4. Wählt eine beliebige, aber sinnvolle Technologie, um eine Benutzer-Schnittstelle zu implementieren. In Frage kommen u.a. Java Swing, Java FX, Angular, Vaadin, React und viele weitere. Achtet auf die richtige Entkopplung (MVC) und auf eine plausible Darstellung der Objekte.
5. Bindet eine relationale Datenbank an, um die Objekte zu persistieren. Das soll in einer separaten Komponente geschehen, die keine Anwendungslogik, sondern nur die Entitätsklassen enthält.

# Detaillierte Aufgabenstellung – Teil 2

6. Die Statistiken zu den Einnahmen einer Aufführung bzw. eines Films haben Aggregat-Charakter. Sie sollen deswegen in einer MongoDB verwaltet werden. Die Integration mit MongoDB soll ebenso als separate Komponente realisiert werden.
7. Die Kommunikation zwischen der Server-Komponente und den Datenbank-Komponenten soll mit RabbitMQ mittels Messaging erfolgen, dabei sollen Kommandos serialisiert und deserialisiert werden.
8. Garantiert Isolation, d.h. keine unerwünschten Effekte, wenn mehrere Anwender konkurrierend auf dieselben Daten zugreifen.

# Spezielle Anforderungen

Unter den folgenden Anforderungen sucht sich jede Gruppe eine aus, die zusätzlich umgesetzt werden soll.

- Die Kommunikation vom Client zum Server ist hinsichtlich Authentifizierung, Vertraulichkeit und Autorisierung abzusichern.
- Die lauffähige Anwendung soll als Container in der Cloud z.B. mittels Docker bereitgestellt werden.
- Statt RabbitMQ soll ein anderer Message Broker verwendet werden, z.B. Apache Kafka.
- Statt MongoDB soll eine andere NoSQL-Datenbank verwendet werden. Auch Graph-Datenbanken sind erlaubt.

# Präsentation

- Pro Termin werden 3 Gruppen präsentieren.
- Pro Gruppe haben wir ca. 30 Minuten Zeit. Präsentiert innerhalb von 15-17 Minuten eure Ergebnisse. Danach sind ca. 12 Minuten für Fragen eingeplant.
- Achtet darauf, dass jede/jeder etwas sagt.
- Eine Live-Demo ist möglich (in den 15 Minuten Präsentationszeit).
- Geht insbesondere auch auf Schwierigkeiten ein und wie ihr diese gelöst habt.
- Zieht auch ein reflektierendes Fazit in Bezug auf die Verwendung der Tools und Frameworks (EMF, RabbitMQ, ...). Das Fazit ist ein Kernergebnis eurer Arbeit und sollte daher mindestens 5 Minuten der Präsentationszeit bekommen.

# Dokumentation

In eure Dokumentation gehören mindestens

- Die Anwendungsfälle
- Das Fachmodell
- Das Klassenmodell
- Die Isolationsstrategie
- Die Gesamtarchitektur
- Und die gewählte spezielle Technik
- Eure Reflektion / euer Fazit in Bezug auf die verwendeten Technologien

Achtet darauf, alles adäquat und aussagekräftig darzustellen. Die Dokumentation sollte 10-15 Seiten lang sein.



# Fazit

Euer Fazit sollte die folgenden Fragen beantworten:

- Wie gut ist die Usability der Modellierungswerkzeuge?
- Werden Service-Schnittstellen (API z.B. REST) klar identifiziert und bereitgestellt?
- Werden Architekturparadigmen automatisch berücksichtigt (MVC, Kommandos,...)?
- Wie ist das MDE-basierte Vorgehen mit verwendeten Frameworks (z.B. Spring, ...) verträglich bzw. kombinierbar?
- Verträglichkeit von (Re-)Generieren und Anreichern?
- Unterstützung für relationale Datenbank und OR-Mapping?
- (Wie) werden bei der Code Generierung abstrakte Klassen, Interfaces und entsprechend Vererbung und Spezialisierung unterstützt? Geht auch Mehrfachvererbung?
- (Wie gut) Werden spezielle Constraints aus dem Modell berücksichtigt (Multiplizitäten, Containments, bidirektionale Assoziationen, evtl. sogar OCL)?
- Wie ist die Integration mit Messaging Werkzeugen?

# Zeitplan

- 28.10. (evtl bis 24.11.: Projektarbeit zur Evaluation von MDE-Tools
  - In der Zeit individuelle Einzeltermine
- 24.11.: Deadline für das Hochladen der Präsentationen
- 25.11.: 4 Präsentationen
- 01.12.: 4 Präsentationen
- 02.12.: 4 Präsentationen
- 19.12.: Deadline für das Hochladen der Dokumentationen