

# Kurs 1793 "Software Engineering I"

Klausur am 08.08.2009

---

## Hinweise zur Bearbeitung der Klausur zum Kurs 1793 "Software Engineering I" im Sommersemester 2009

Wir begrüßen Sie zur Klausur "Software Engineering I". Bitte lesen Sie sich diese Hinweise vollständig und aufmerksam durch, bevor Sie mit der Bearbeitung der Aufgaben beginnen.

- 1) Prüfen Sie bitte die Vollständigkeit Ihrer Unterlagen. Ihr Klausurexemplar umfasst:
  - 2 Deckblätter,
  - 1 Formblatt für eine Bescheinigung für das Finanzamt,
  - diese Hinweise zur Bearbeitung,
  - 4 Aufgaben (Seite 3 - Seite 11)
- 2) Füllen Sie, **bevor** Sie mit der Bearbeitung der Aufgaben beginnen, folgende Seiten des Klausurexemplars aus:
  - a) **Beide** Deckblätter mit Name, Anschrift sowie Matrikelnummer. **Markieren Sie vor der Abgabe auf beiden Deckblättern die von Ihnen bearbeiteten Aufgaben.**
  - b) Falls Sie eine Teilnahmebescheinigung für das Finanzamt wünschen, füllen Sie bitte das entsprechende Formblatt aus.

**Nur wenn Sie beide Deckblätter vollständig ausgefüllt haben, können wir Ihre Klausur korrigieren!**
- 3) Streichen Sie ungültige Lösungen deutlich durch.
- 4) Schreiben Sie bitte auf jedes beschriebene Blatt oben links Ihren Namen und oben rechts Ihre Matrikelnummer. Wenn Sie weitere eigene Blätter benutzt haben, heften Sie auch diese, mit Namen und Matrikelnummer versehen, an Ihr Klausurexemplar.

**Lose Blätter, insbesondere ohne Name und Matrikelnummer, werden nicht bewertet!**
- 5) Neben Schreibgerät und unbeschriebenen Konzeptpapier sind **Kurseinheiten, Einsendeaufgaben** sowie deren **Musterlösungen** als Unterlagen zugelassen.
- 6) Mit **Bleistift** geschriebene oder gezeichnete Lösungen werden **nicht korrigiert**. Verwenden Sie bitte nur Füller oder Kugelschreiber.
- 7) Es sind maximal 100 Punkte erreichbar.

Wir wünschen Ihnen bei der Bearbeitung der Klausur viel Erfolg!



**Aufgabe 1** (28 Punkte)

Eine Direktbank plant, den Kunden eine Online-Plattform zur Verwaltung ihrer Wertpapier-Depots zur Verfügung zu stellen. Diese soll Informationen liefern (Anzeige des Depotbestands und des aktuellen Depotwerts, Ermittlung der aktuellen Kurse von Wertpapieren, ...), aber auch die Aufgabe von Orders (Kauf oder Verkauf) ermöglichen. In dieser Aufgabe geht es darum, wichtige Datenanforderungen an die geplante Software in einem **Domänenklassenmodell** festzuhalten. Diesbezügliche Gespräche mit einem Domänenexperten lassen sich wie folgt zusammenfassen:

*„Wertpapiere werden an zahlreichen Börsen auf der Welt gehandelt. Jede Börse besitzt ein eindeutiges, aus drei Buchstaben bestehendes Kürzel. Der Kunde orientiert sich jedoch eher am Ort und Namen der Börse (z.B. die NASDAQ in New York). Damit ein Wertpapier gehandelt werden kann, muss es an mindestens einer Börse gelistet sein. Ein Listing an mehreren Börsen ist möglich und für große Werte durchaus üblich. Wir planen, zunächst nur zwei Arten von börsenhandelbaren Wertpapieren zu betrachten: Aktien und Optionsscheine auf Aktien. Wertpapiere beider Arten besitzen weltweit eindeutige, 6-stellige Wertpapierkennnummern (WPKN). An jedem Handelstag einer Börse ist es möglich, die dort gelisteten Wertpapiere zu kaufen oder zu verkaufen. Wenn an einem Tag Umsätze zustande gekommen sind, so interessieren den Kunden (insbesondere dann, wenn er eine Order (s.u.) aufgeben möchte) der höchste und der letzte Kurs, zu denen ein Papier gehandelt wurde. Wichtig ist auch das Handelsvolumen, d.h. der Tagesumsatz eines Papiers in Stück.*

*Betrachten wir nun die beiden relevanten Arten von Wertpapieren etwas genauer: Eine Aktie bezeichnet einen Anteil an einem Unternehmen und besitzt einen festen, ganzzahligen Nennwert (z.B. 10 Euro). Ferner ist für den Anleger interessant, welche Dividende (in Emissionswährung) zuletzt pro Aktie gezahlt wurde. Der Besitz eines Optionsscheins auf eine Aktie berechtigt dazu, diese Aktie zu einem bestimmten Zeitpunkt (der Fälligkeit) und festen Preis (der Basis) in der Emissionswährung zu erwerben oder zu verkaufen. Wir unterscheiden entsprechend zwischen Optionsscheinen der Typen CALL (Kauf) und PUT (Verkauf). Wie viele Aktien auf einen Optionsschein erworben bzw. verkauft werden können, gibt das Bezugsverhältnis an. Möglich sind Bezugsverhältnisse von 1:1 und 1:10. Optionsscheine beziehen sich stets auf Aktien, die an mindestens einer Börse weltweit gelistet sind. Verschiedene Optionsscheine können sich auf dieselbe Aktie beziehen.*

*Ein Wertpapierdepot ist stets einem einzelnen Kunden zugeordnet, wobei ein Kunde auch Inhaber mehrerer Depots sein kann. Es setzt sich aus einzelnen Positionen zusammen, wobei sich eine Position auf ein bestimmtes Wertpapier bezieht und eine Stückzahl angibt (z.B. 50 Siemens-Aktien).*

*Wenn der Kunde eine Position eines Depots vergrößern oder neu anlegen möchte, muss er eine Kauforder aufgeben. Dabei muss er neben der Stückzahl und dem gewünschten Wertpapier angeben, wie viel er maximal zu zahlen bereit ist und welche Börse er mit der Abwicklung des Vorgangs beauftragen möchte. Bei einer Verkauforder ist der Ablauf der gleiche, nur dass der Preis angegeben wird, den der Verkäufer mindestens haben möchte. Eine Order verbleibt auch nach der Abwicklung oder Stornierung im System. Sie wird dann entsprechend gekennzeichnet und ggf. um die Information ergänzt, zu welchem Preis (pro Stück) der Handel abgeschlossen wurde.“*

**Aufgabenstellung:**

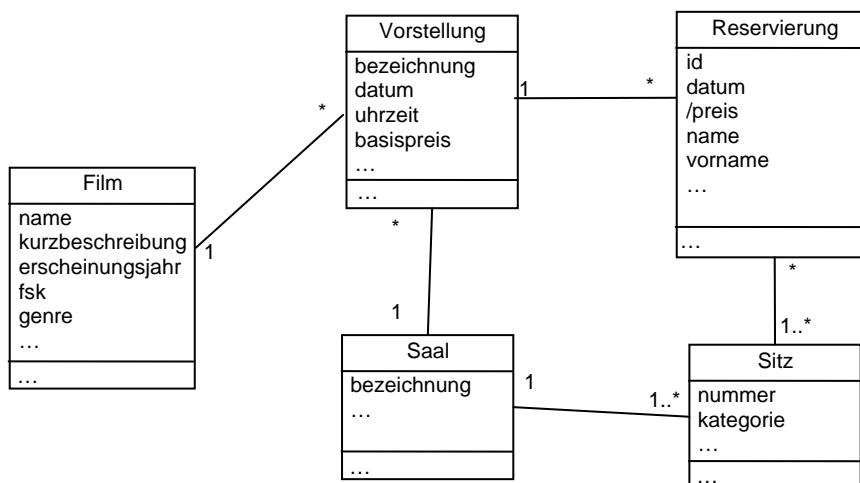
- Erstellen Sie ein redundanzfreies **Domänenklassenmodell** für die beschriebene Anwendungsdomäne. Definieren Sie dabei nicht unnötig viele Klassen.
- Geben Sie zu jeder Beziehung die Multiplizitäten an.
- Ergänzen Sie in Ihren Klassen die für das Verständnis Ihres Modells relevanten Attribute.
- Geben Sie den verwendeten Assoziationen aussagekräftige Bezeichnungen.

**Aufgabe 2** (25 Punkte)

Erstellen Sie, ausgehend vom Klassendiagramm in Abb. 1, ein *Feinentwurfs-Klassendiagramm* durch Anwendung der im Kurstext beschriebenen Transformationsregeln. Beachten Sie dabei die folgenden Punkte:

- Stellen Sie in Ihrem Diagramm alle Elemente dar, die durch die Transformation der Assoziationen und eventueller echter Ganzes-Klassen neu hinzugekommen sind.
- Geben Sie auch die hinzugekommenen Attribute und Standardoperationen (mit Sichtbarkeiten und Parametertypen) an. Sie müssen diese nicht in Ihr Diagramm einzeichnen, eine Auflistung auf einem separaten Blatt Papier reicht aus.
- Tragen Sie an den Assoziationsenden die Multiplizitäten ein.
- Von Ordner-, Relations- und Mengenklassen ausgehende Assoziationen müssen nicht weiter transformiert werden.

*Hinweis:* Assoziationen ohne Angabe von Pfeilspitzen sind als bidirektional anzusehen.



**Abb. 1:** Zu transformierendes Entitätsklassen-Diagramm zur Aufgabe 1

**Aufgabe 3** (27 Punkte)

Die Verwaltungssoftware eines Kinos soll erweitert werden. Im Rahmen dieser Maßnahmen soll es den Kino-Mitarbeitern ermöglicht werden, Reservierungen in das System einzutragen und zu verwalten. Gehen Sie im Folgenden davon aus, dass der Mitarbeiter mit dem Kunden telefoniert und dabei dessen Angaben in das System einträgt. Der Ablauf einer Reservierung ist den nachfolgenden Dokumenten zu entnehmen.

Im Rahmen dieser Aufgabe soll ein *Sequenzdiagramm* erstellt werden, welches das Szenario „Kunde reserviert einzelnen Sitzplatz - erfolgreich“ behandelt. Gehen Sie davon aus, dass der Kunde nur einen einzigen Sitzplatz reserviert.

Wir befinden uns in der *Analyse* und können auf die Dokumente der Anforderungsspezifikation zurückgreifen, s. dazu die nachfolgend beschriebenen Anwendungsfälle, das Anwendungsfalldiagramm (Abb. 4) und den Ausschnitt aus dem Analyseklassenmodell (Abb. 5). Letzteres enthält noch keine Nicht-Standardoperationen, diese müssen Sie bei Bedarf selbst ergänzen. Sie dürfen so viele Operationen einführen, wie Sie zur Lösung der Aufgabe benötigen.

*Hinweise:*

- Vervollständigen Sie das Sequenzdiagramm in Abb. 6. Die Abb. 6a ist mit Abb. 6 identisch und als Konzeptpapier gedacht, d.h. Sie können darauf zunächst Ihre Lösung skizzieren.
- Modellieren Sie bitte nicht die detaillierten Abläufe der Ereignisbehandlung! Wir berücksichtigen in der Analyse noch kein GUI-Rahmenwerk. Hier reicht die Angabe aus, dass der Kino-Mitarbeiter beispielsweise eine bestimmte Vorstellung oder einen bestimmten Sitz auswählt.
- Falls Sie Objekte von Entitätsklassen erzeugen, vergessen Sie die Attribute und Objektverbindungen nicht! In der Analyse wird das zwar normalerweise nicht im Detail modelliert, wir erwarten es hier aber trotzdem.

Film	FSK	Saal	So, 21.06.	Mo, 22.06	Di, 23.06
Star Trek	12	1	-	17:15 20:15	14:30 20:15
State of Play	12	1	14:30 20:15	14:30 20:15	-
Terminator – Salvation	16	3	20:00	20:15 22:15	20:00 23:00
X-Men Origins: Wolverine	16	4	20:15 23:00	20:15 22:15	17:00 -
Sneak Preview	18	4	-	-	20:15

**Abb. 2:** Im gewählten Kino verfügbare Vorstellungen (Auswahl durch Anklicken der Uhrzeit mit der Maus).

gewählte Sitzplätze: 4, Reihe 14  
Preis: 21,50 Euro

Name:

Vorname:

**Abb. 3:** Darstellung des Kinosaals; Auswahl der Sitzplätze per Mausclick

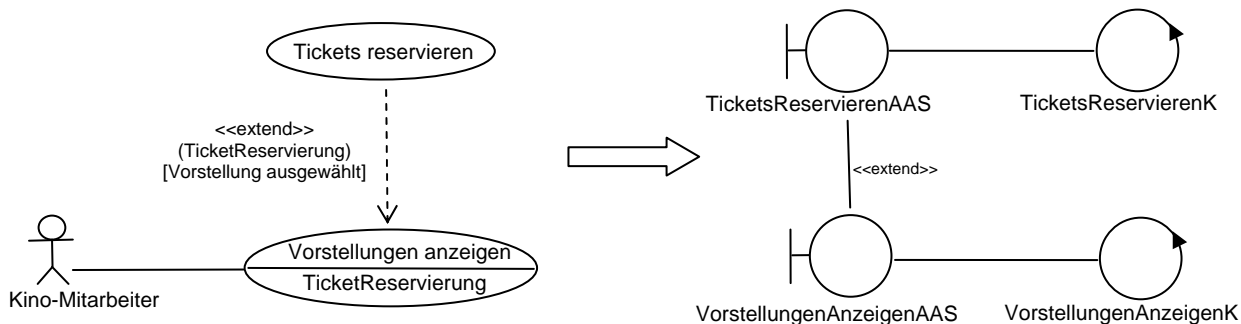


Abb. 4: Die beteiligten Anwendungsfälle und die daraus resultierenden Analyseklassen

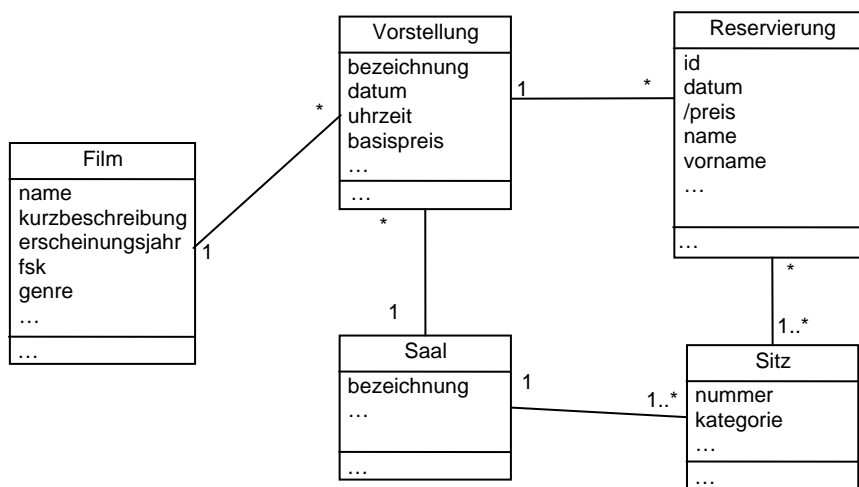


Abb. 5: Einige Entitätsklassen des Analyseklassenmodells

**use case** Vorstellungen anzeigen

**actors**

Kino-Mitarbeiter

**precondition**

-

**main flow**

Zunächst werden alle Vorstellungen ermittelt, die innerhalb der nächsten  $n$  Tage in diesem Kino laufen. Eine Möglichkeit, wie man diese Vorstellungen dem Kino-Mitarbeiter am Bildschirm präsentieren könnte, zeigt die GUI-Skizze in Abbildung 2. Dabei wird in Form einer Tabelle dargestellt, welcher Film wann und wo vorgeführt wird. Die Auswahl einer Vorstellung erfolgt per Mausklick auf eine Uhrzeit, ähnlich wie bei einem Link auf einer Webseite. Nach erfolgter Auswahl wird der Reservierungsvorgang gestartet (extension point: TicketReservierung).

**alternative flow**

Der Kunde startet keinen Reservierungsvorgang.

**postcondition**

...

**exceptional flow**

...

**postcondition**

...

**end** Vorstellungen anzeigen

**Kurs 1793 “Software Engineering I”**Klausur am 08.08.2009

---

**use case** Tickets reservieren**actors**

Kino-Mitarbeiter

**precondition**

Eine Vorstellung wurde ausgewählt.

**main flow**

Beim Start des Anwendungsfalls, also nach Auswahl einer Vorstellung, wird dem Kino-Mitarbeiter ein Fenster angezeigt, das etwa so wie die GUI-Skizze in Abbildung 3 aussehen soll. Dieses zeigt in einer interaktiven Skizze die aktuelle Sitzbelegung der Vorstellung (genauer: des zugehörigen Kino-Saals) an; bereits reservierte Plätze sind mit einem X gekennzeichnet. Der Mitarbeiter wählt in dieser Skizze per Mausklick den gewünschten Platz bzw. die gewünschten Plätze aus. Der Preis der aktuellen Auswahl wird im Fenster angezeigt (Abb. 3); er berechnet sich aus dem Basispreis der Vorstellung und der Kategorie des Sitzes bzw. der Sitze (Vordere Reihen, Parkett oder Loge). Bezahlt wird erst bei Abholung der Tickets, die Reservierung selbst kostet nichts. Danach gibt der Kino-Mitarbeiter den Namen und den Vornamen des Kunden ein und schließt den Eingabevorgang mit dem Drücken des Reservieren-Buttons ab. Danach wird die Zulässigkeit der Eingaben geprüft (Details für die Aufgabe nicht von Interesse). Nach erfolgreicher Prüfung wird die Reservierung im System registriert. Dabei wird jeder Reservierung eine eindeutige ID zugewiesen (deren Bestimmung ist für die Aufgabe nicht von Interesse). Diese wird zum Schluss zusammen mit einer Erfolgsmeldung am Bildschirm angezeigt.

**alternative flow**

...

**postcondition**

Ein oder mehrere Tickets sind für den Kunden reserviert.

**exceptional flow**

...

**postcondition**

...

**end** Tickets reservieren

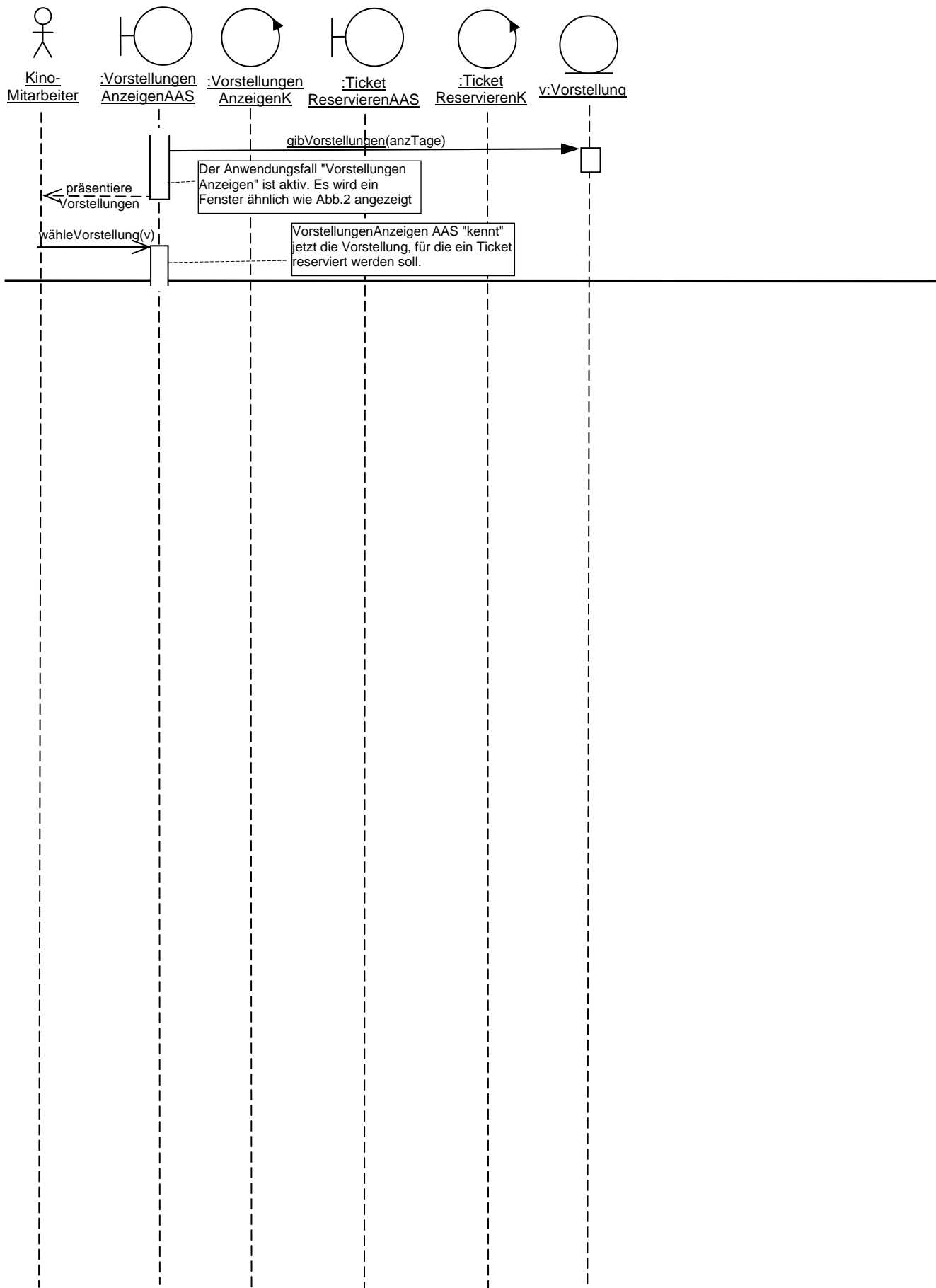
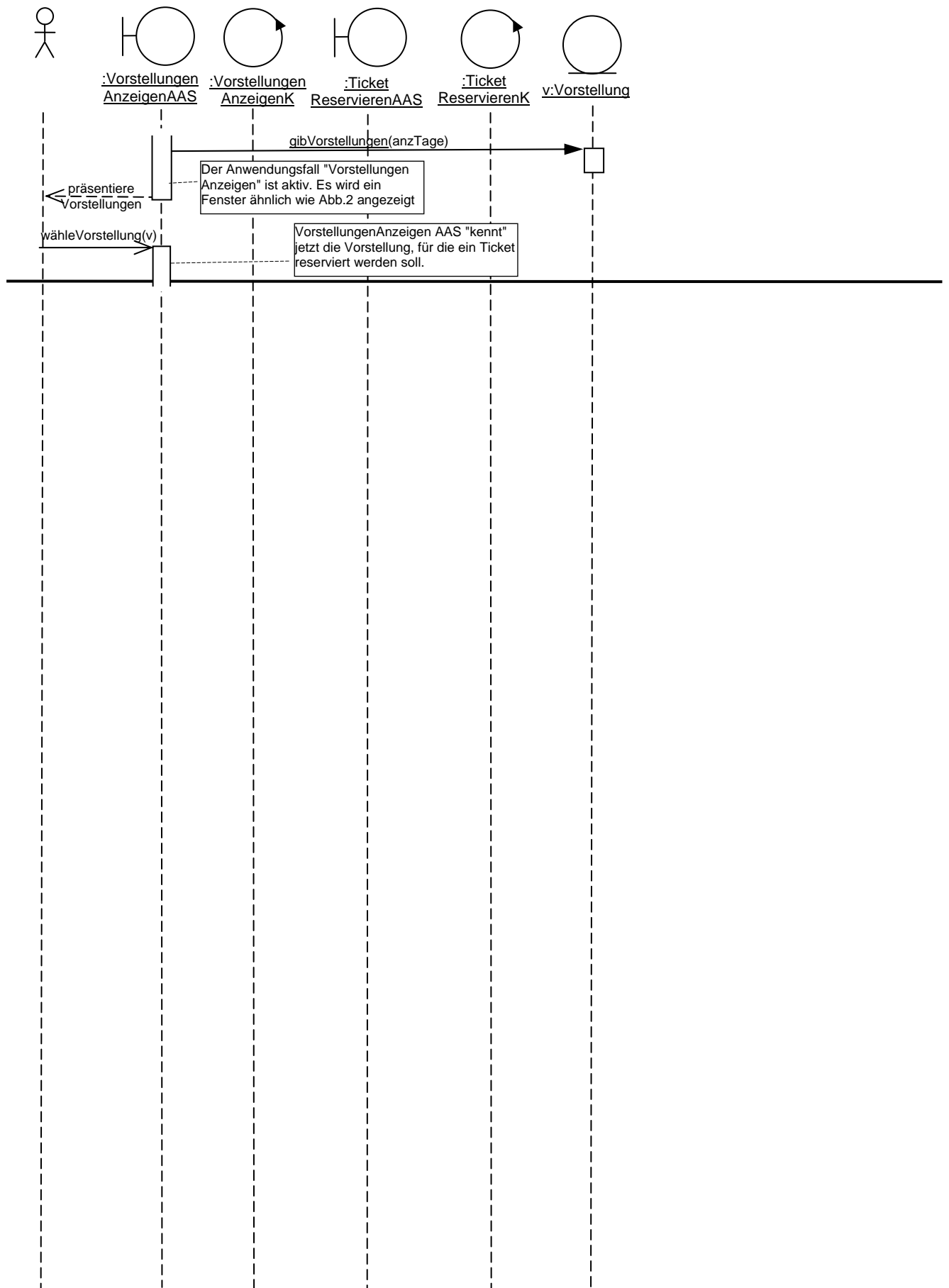


Abb. 6: Sequenzdiagramm zum Szenario „Kunde reserviert einzelnen Sitzplatz - erfolgreich“



**Kurs 1793 "Software Engineering I"**

Klausur am 08.08.2009



**Abb. 6a:** Sequenzdiagramm zum Szenario „Kunde reserviert einzelnen Sitzplatz - erfolgreich“

---

**Aufgabe 4** (20 Punkte)

Der Hersteller einer Software möchte dem Benutzer eine Funktionalität anbieten, über die er bei Fehlverhalten der Software mit minimalem Aufwand einen Fehlerbericht verfassen und an den Hersteller schicken kann. Ihre Aufgabe ist es, den Lebenszyklus eines solchen Fehlerberichts (nachdem er beim Hersteller eingegangen ist) in einem **Zustandsdiagramm** zu modellieren.

*Folgendes ist dabei zu beachten:*

Zunächst ist der Fehlerbericht als solcher im System zu registrieren. In regelmäßigen Abständen sichtet der für die Bewertung und Verteilung zuständige Mitarbeiter die neu eingegangenen Fehlerberichte. Meldungen zu bereits bekannten Fehler oder mit unbrauchbarem Inhalt werden gelöscht. Allen anderen Fehlerberichten wird entweder die Priorität 1 (hoch) oder 2 (niedrig) zugeordnet. Fehler mit hoher Priorität werden automatisch einem Mitarbeiter zugewiesen (ausgewählt wird dabei der Mitarbeiter mit der aktuell geringsten Auslastung, Details sind hier nicht von Interesse). Fehler mit niedriger Priorität werden zunächst zurückgestellt, also gewissermaßen auf eine Warteliste gesetzt.

Wenn ein Mitarbeiter "frei" wird, d.h. wenn er keine Fehlerberichte mehr zur Bearbeitung hat, bekommt er einen aus der Warteliste (also mit Priorität 2) zugeteilt.

Prioritäten können auch geändert werden. Fehlermeldungen, die von hoher auf niedrige Priorität zurückgesetzt werden, verbleiben standardmäßig beim zuständigen Mitarbeiter. Dieser kann Prio2-Fehler aber selbst auf die Warteliste zurücksetzen, d.h. er gibt dann seine Zuständigkeit ab. Fehlerberichte, deren Dringlichkeit angehoben wird, werden (wie oben beschrieben) automatisch einem Mitarbeiter zugewiesen. Von Zeit zu Zeit wird die Warteliste aufgeräumt, d.h. es werden Fehlerberichte aus dieser entfernt (gelöscht).

Sobald der Fehler (bzw. der Fehlerbericht) einem Mitarbeiter zugewiesen wurde, kann dieser mit der Beseitigung beginnen. Wenn er den Fehler korrigiert hat (bzw. der Meinung ist, den Fehler korrigiert zu haben), benachrichtigt er den Abteilungsleiter. Dieser wird zu gegebener Zeit das Ergebnis der Fehlerbeseitigung begutachten und dann entweder grünes Licht geben (d.h. den Fehler als behoben markieren) oder aber den Fehler erneut einem Mitarbeiter zur Bearbeitung zuweisen. Ein abgeschlossener Fehlerbericht bleibt noch mindestens ein halbes Jahr im System gespeichert. Erst nach Ablauf dieser Aufbewahrungsfrist kann er aus dem System gelöscht werden.

**Kurs 1793 "Software Engineering I"**Klausur am 08.08.2009

---

Orientieren Sie sich bei der Beschriftung der Zustandsübergänge an der folgenden Notation:

*Ereignis [Wächterbedingung] / Aktionsfolge*

Beispiel: Kartenautomat im Parkhaus

