

**Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"**  
Klausur am 29.3.2003

---

Wintersemester 2002  
Hinweise zur Bearbeitung der Klausur zum Kurs 1793  
"Software Engineering I - Grundkonzepte der OOSE"

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,
  - diese Hinweise zur Bearbeitung,
  - 1 Formblatt für eine Bescheinigung für das Finanzamt,
  - 4 Aufgaben (Seite 3 - Seite 7)
  
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 unbeschriebenem Konzeptpapier sind **Kurseinheiten, Einsendeaufgaben** sowie deren **Musterlösungen** als Unterlagen zugelassen.
  
6. Mit **Bleistift** geschriebene oder gezeichnete Lösungen werden **nicht korrigiert**. Verwenden Sie bitte für Ihre Lösungen nur Füller oder Kugelschreiber.
  
7. Es sind maximal 110 Punkte erreichbar.
  
8. Die Korrektur der Klausur kann aus organisatorischen Gründen etwa 2 Monate in Anspruch nehmen.

Wir wünschen Ihnen bei der Bearbeitung der Klausur viel Erfolg!

**Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"**Klausur am 29.3.2003

---

**Aufgabe 1** (20 + 7 + 3 = 30 Punkte)

Der Leiter eines Fitnessstudios möchte die Verwaltung von Kunden- und Kursdaten automatisieren und beschreibt die Anforderungen an ein entsprechendes Verwaltungssystem wie folgt:

"Wir bieten unseren Kunden Jahresverträge zu unterschiedlichen Tarifen an. Die Kunden können bei Vertragsabschluss den Vertragsbeginn und den Tarif frei wählen. Ein Tarif hat (neben der Bezeichnung) einen bestimmten Preis und ggf. eine oder mehrere Kursarten, für die eine Kursteilnahme im Preis enthalten ist. Zusätzlich kann ein individueller Rabatt für einen Jahresvertrag gewährt werden. Alle Kurse einer Kursart haben denselben Preis.

Damit ein Kunde an einem konkreten Kurs einer Kursart teilnehmen kann, muss er einen gültigen Jahresvertrag besitzen und sich vor Kursbeginn anmelden. Eine Kursanmeldung zu einem bestimmten Kurs ist nicht möglich, wenn die maximale Teilnehmeranzahl für diesen Kurs überschritten ist (wobei immer eine ausreichende Anzahl von alternativen Kursen einer Kursart angeboten werden). Jeder Kurs wird von einem Trainer geleitet.

Für den Versand von Prospekten und Geburtstagsgrüßen an unsere Kunden wollen wir automatisch Adressetiketten erstellen können. Unsere Prospekte enthalten u.a. die Namen und Kurzbeschreibungen der Trainer sowie, sortiert nach Kursart, eine Auflistung der Kurse unter Angabe der Kurs-ID, des Trainers, des Wochentags (1 bis 7), der Zeitangabe, sowie des ersten Kurstages und der Anzahl der Kurstage.

Das Verwaltungssystem soll in der Lage sein, folgende Anfragen zu beantworten:

1. An welchem Tag hat ein bestimmter Kunde das erste Mal einen Jahresvertrag abgeschlossen?
2. Wieviele Kunden haben jeweils an den Kursen der unterschiedlichen Kursarten teilgenommen?
3. Welcher Trainer hatte im aktuellen Jahr die meisten Kursteilnehmer?
4. Wann hat sich der erste Kunde zu einem bestimmten Kurs angemeldet?"

**Aufgabenstellung:**

- a) Erstellen Sie ein redundanzfreies Domänen-Klassendiagramm für die beschriebene Anwendungsdomäne. Geben Sie zu jeder Beziehung die Multiplizitäten an.
- b) Skizzieren Sie ein gültiges Beispiel-Objektdiagramm, bei dem sich ein Kunde zu einem Kurs angemeldet hat. Wertbelegungen von Attributen können Sie vernachlässigen.
- c) Begründen Sie (möglichst in einem Satz), warum in Objektdiagrammen keine Multiplizitäten eingezeichnet werden.

**Aufgabe 2** (20 + 7 + 3 = 30 Punkte)

Für ein bestimmtes Theater soll ein Programm zur Verwaltung der Kartenreservierungen erstellt werden. Das Theater verfügt über eine bestimmte Anzahl von Sitzen. Jeder Sitz hat eine Nummer, steht in einer bestimmten Reihe und gehört zu einer bestimmten Kategorie. Die Kategorien "Parkett" und "Loge" haben einen unterschiedlichen Preis je Ticket. Im Theater finden Vorstellungen und Premieren statt. Premieren sind besondere Vorstellungen, für welche die Anzahl der verkauften Tickets nachzuhalten ist. Für alle Vorstellungen soll das Datum und der Beginn der Vorstellung abgespeichert werden.

Kunden müssen Tickets reservieren, um an den Vorstellungen teilnehmen zu können, entweder als Einzelreservierung (für eine bestimmte Vorstellung) oder als Jahresabonnement für 6 bis 12 Vorstellungen in einer bestimmten Kategorie in einem Kalenderjahr. Bei Einzelreservierungen spielt die Identität des Kunden keine Rolle - er erhält eine Reservierungsnummer und ein Kennwort, mit welchen die Karte an der Abendkasse abgeholt werden kann. Bei Jahresabonnements muss der Kunde mit Namen und Adresse bekannt sein. Zusätzlich muß bei jedem Jahresabonnement die Bankverbindung (d.h. Kontonummer und Bankleitzahl) gespeichert werden, über welche die Bezahlung abgewickelt werden soll. Verfügt ein Kunde über mehrere Jahresabonnements, so kann er gleichzeitig nur eine Bankverbindung angeben. Bankverbindungen für nicht aktuelle Jahresabonnements werden nicht gespeichert. Kundendaten hingegen werden auch nach Ablauf aller Jahresabonnements des betreffenden Kunden gespeichert. Auf den Tickets sollen die Ticketnummer sowie Titel, Datum und Beginn der Vorstellung aufgedruckt werden.

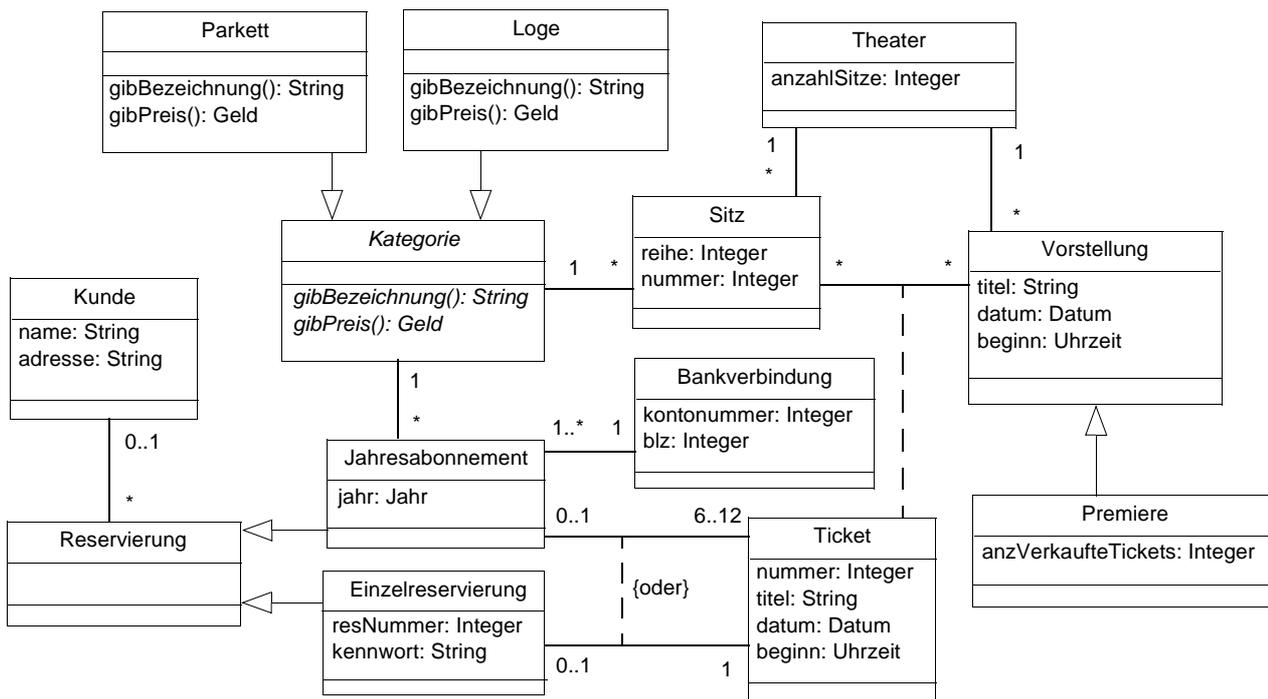


Abb. 1 Schlechtes Klassendiagramm zur Theaterverwaltung

**Kurs 1793 “Software Engineering I - Grundkonzepte der OOSE”**Klausur am 29.3.2003

---

Abb. 1 zeigt einen Versuch, die Problemwelt dieser Anwendung durch ein Klassendiagramm zu modellieren. Allerdings widerspricht dieses Klassendiagramm einigen Heuristiken aus dem Kurs, enthält überflüssige Klassen sowie Redundanzen und Modellierungsfehler.

***Aufgabenstellung***

- a) Ändern Sie das Klassendiagramm aus Abb. 1, so dass es
- möglichst den Heuristiken im Kurs entspricht,
  - möglichst wenige Klassen enthält und
  - keine Redundanzen enthält.
- b) Begründen Sie jede Ihrer Änderungen in maximal drei Sätzen.
- c) Ergänzen Sie in Ihrem Klassendiagramm die fehlenden Multiplizitäten.

### Aufgabe 3 (4 + 20 + 6 = 30 Punkte)

Gegeben ist ein Analyse-Klassendiagramm (Abb. 2) für eine Softwareanwendung zur Verwaltung eines Videoverleihs sowie folgende textuelle Spezifikation eines Anwendungsfalls:

**use case** Video ausleihen

**actors**

Mitarbeiter, Drucker

**precondition**

-

**main flow**

Der Mitarbeiter selektiert zunächst den Kunden und dann die zur Ausleihe gewünschten Videos. Dann startet der Mitarbeiter den Druck einer Leihbestätigung, die den Namen des Kunden sowie eine **sortierte** Liste aller (d.h. auch bereits früher) vom Kunden entliehenen (aber noch nicht zurückgegebenen) Videos mit Titel und Kategorie enthält.

**postcondition**

Das aktuelle Datum ist als Leihdatum für die neu entliehenen Videos vermerkt.

**end** Video ausleihen

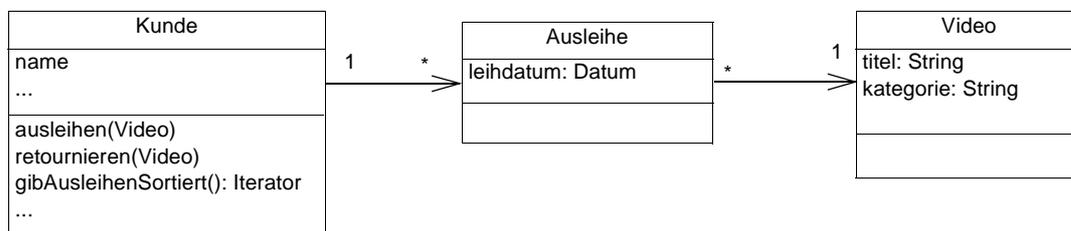


Abb. 2 Analyse-Klassendiagramm

### Aufgabenstellung:

- Zeichnen Sie den Mechanismus des Anwendungsfalls "Video ausleihen" für den Grobentwurf mit Schnittstellen-, Kontroll- und Entitätsklassen. Beginnen Sie dabei mit der entsprechenden Anwendungsfall-Akteur-Schnittstellenklasse. Attribute und Operationen können Sie vernachlässigen.
- Zeichnen Sie ein Sequenzdiagramm für folgendes Szenario zum Anwendungsfall "Video ausleihen": "Ein neuer Kunde entleiht sein erstes Video." Berücksichtigen Sie dabei auch die Aufrufe von Standardoperationen sowie die Nutzung eines Iterators.

Gehen Sie von der Existenz einer Instanz der Anwendungsfall-Akteur-Schnittstellenklasse sowie einer Instanz der zugeordneten Kontrollklasse aus und beginnen Sie die Darstellung des Sequenzdiagramms mit der Selektion des bereits erfassten Kunden durch den Akteur Mitarbeiter.

- Beantworten Sie die folgende Frage mit "ja" oder "nein": "Sind Sequenzdiagramme gut geeignet, den Ablauf von Anwendungsfällen (nicht gemeint sind damit Szenarios) darzustellen?" Begründen Sie Ihre Antwort in maximal 3 Sätzen.

**Kurs 1793 “Software Engineering I - Grundkonzepte der OOSE”**  
Klausur am 29.3.2003

---

**Aufgabe 4** (6 + 14 = 20 Punkte)

Gegeben ist das vereinfachte Grobentwurfs-Klassendiagramm in Abb. 3.

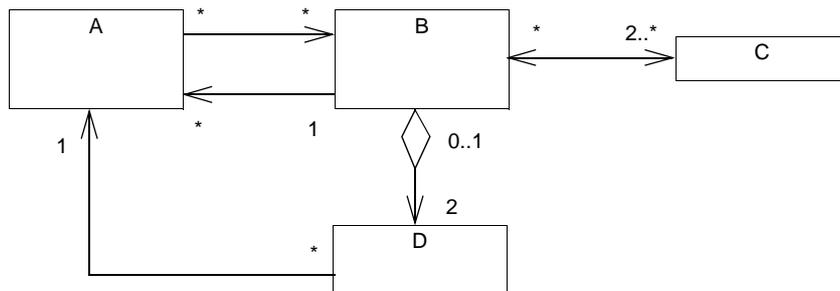


Abb. 3 Grobentwurfs-Klassendiagramm

**Aufgabenstellung**

- a) Geben Sie die echten Ganzes-Klassen an.
- b) Erstellen Sie, ausgehend von Abb. 3, ein Feinentwurfs-Klassendiagramm durch Anwendung der im Kurstext beschriebenen Transformationen.
  - Geben Sie alle Klassen, Attribute und Standardoperationen an, die sich aus den Transformationen ergeben.
  - Berücksichtigen Sie auch Sichtbarkeiten und Parametertypen.
  - Tragen Sie an den Assoziationsenden die Multiplizitäten ein. (Die Angabe von Assoziationsnamen ist nicht erforderlich.)

*Hinweis:* Assoziationen, die sich direkt auf ein Attribut abbilden lassen, sowie von Ordner- und Relationenklassen ausgehende Assoziationen müssen nicht weiter transformiert werden.