

Kurs 1793 "Software Engineering I"

Klausur am 11.8.2007

Hinweise zur Bearbeitung der Klausur zum Kurs 1793 "Software Engineering I" im Sommersemester 2007

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 15)
 - 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 (26 + 4 Punkte)

Ihre Aufgabe ist es, ein *Domänenklassenmodell* für die folgenden Anforderungen zu erstellen:

„An der Universität Musterstadt (Präsenzuniversität) hat eine Vorlesungsreihe den Stellenwert eines Kurses an der Fernuniversität in Hagen. Bestandteil einer Vorlesungsreihe sind neben den einzelnen Vorlesungen die Übungsveranstaltungen, die den Zweck haben, die Vorlesungsinhalte zu vertiefen und Beispielaufgaben zu bearbeiten. Diese Veranstaltungen finden in Übungsgruppen statt. Deren Größe darf 25 Personen nicht übersteigen, damit jeder Student die Möglichkeit hat, sich aktiv zu beteiligen und seine Fragen zu stellen.

Die Zusammensetzung der Übungsgruppen wird zu Semesterbeginn festgelegt, indem sich die Studenten beim jeweiligen Lehrgebiet in entsprechende Listen eintragen. Ein Student absolviert seine Übungsveranstaltungen also immer in derselben Gruppe, ein Wechsel ist nicht möglich. Jede Übungsgruppe wird von einem Tutor (Übungsleiter) betreut. Die Tutoren sind erfahrene Studenten, die in einem früheren Semester selbst Teilnehmer einer entsprechenden Übungsgruppe waren. Ein Student kann in einem Semester maximal eine Übungsgruppe als Tutor betreuen. Jedem Tutor wird für seine jeweilige Betreuungstätigkeit ein so genannter Mentor zur Seite gestellt; dies ist ein wissenschaftlicher Mitarbeiter des betreffenden Lehrstuhls, der dem Tutor bei fachlichen und administrativen Fragen behilflich ist.

Zu einer einzelnen Vorlesungsreihe werden bis zu 10 Übungsveranstaltungen angesetzt. Die Treffen einer Übungsgruppe finden stets am gleichen Wochentag um die gleiche Uhrzeit statt (also z.B. immer Dienstags von 15 bis 17 Uhr).

Der so genannte Übungsnachweis ist Voraussetzung für die Teilnahme an der Abschlussklausur. Einen solchen erhält jeder Student, der die Mindestpunktzahl bei den Hausübungen (entspricht den Einsendeaufgaben an der Fernuni, maximal 7 pro Semester) erreicht hat und regelmäßig anwesend war (max. zwei Mal unentschuldig gefehlt). Es gehört daher zu den Aufgaben eines Tutors, die Hausübungen seiner Gruppe zu korrigieren und zu bewerten, sowie die regelmäßige Anwesenheit der Studenten seiner Übungsgruppe zu überprüfen.“

Hinweise:

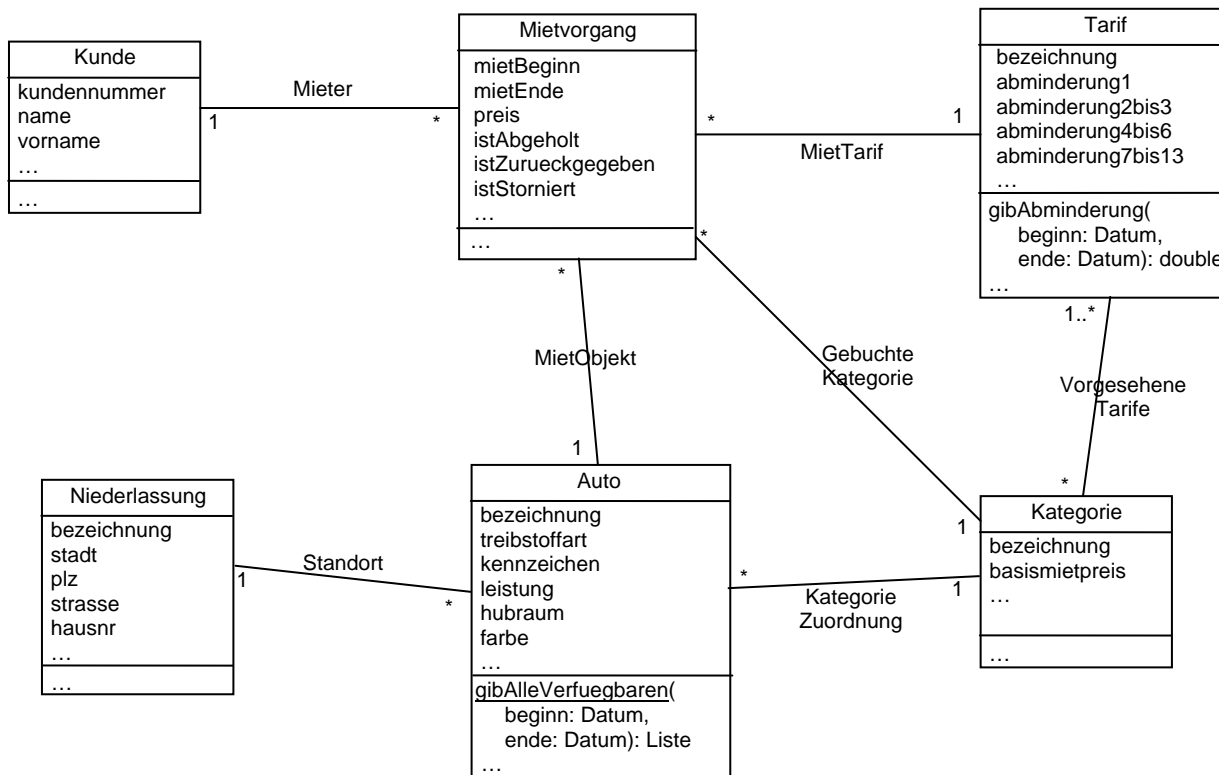
- Die Abschlussklausur selbst ist im Rahmen dieser Aufgabe nicht zu behandeln.
- Gehen Sie davon aus, dass bei regelmäßig stattfindenden Vorlesungsreihen für jedes betroffene Semester ein neues Vorlesungsreihe-Objekt erzeugt wird.

Aufgabenstellung:

- a) Erstellen Sie ein redundanzfreies Klassendiagramm für die beschriebene Anwendungsdomäne. Geben Sie zu jeder Beziehung die Multiplizitäten an. Geben Sie zu Ihren Klassen insgesamt mindestens 14 sinnvolle Attribute an (gewertet werden maximal drei Attribute pro Klasse)
- b) Beschreiben Sie anhand zweier ausgewählter Assoziationen, warum die Multiplizitäten alleine nicht ausreichen, um durch die Realität vorgegebene Einschränkungen bezüglich der erlaubten Verbindungen zwischen Objekten wiederzugeben.

Aufgabe 2 (11 + 11 + 4 = 26 Punkte)

Für die Autovermietung CarBerlin ist eine Software zur Unterstützung der Abwicklung von Mietvorgängen und der Verwaltung des Fahrzeugbestands zu entwickeln. Gegeben ist der in Abb. 1 dargestellte Ausschnitt des Analyseklassenmodells, der einige Entitätsklassen der Software zeigt.



Hinweise zu den Operationen:

Tarif.gibAbminderung(beginn: Datum, ende: Datum):double
Bestimmt für einen gegebenen Zeitraum den maßgebenden Abminderungsfaktor.

Auto.gibAlleVerfuegbaren(beginn: Datum, ende: Datum):Liste
Ermittelt alle im gegebenen Zeitraum verfügbaren Fahrzeuge und gibt eine Liste mit deren Bezeichnungen zurück.

Abb. 1: Entitätsklassen zur Aufgabe 2

Wir befinden uns im **Grobentwurf** und betrachten den Anwendungsfall KFZReservieren, genauer gesagt das Szenario „KFZReservieren erfolgreich – Kunde im System vorhanden, Kunde hat seine Kundennummer parat“. In Abb. 2 ist ein (derzeit noch unvollständiges) **Sequenzdiagramm** dargestellt, das diesen Ablauf modellieren soll. Ihre Aufgabe ist es, die noch fehlenden Teile zu ergänzen.

Berücksichtigen Sie dabei die folgenden Informationen aus der Domänenbeschreibung des Kunden:

„Unsere Autovermietung verfügt über eine große Anzahl an Fahrzeugen, die in verschiedene Preiskategorien eingeteilt sind. Für jede dieser Kategorien (z.B. Economy, Mittelklasse, Luxusklasse etc.) wird ein Basispreis festgelegt, der dem Preis für einen eintägigen Mietvorgang ohne Ermäßigung entspricht.“

Kurs 1793 "Software Engineering I"Klausur am 11.8.2007

Der Tagespreis hängt nicht nur von der Kategorie, sondern auch von der Mietdauer ab. Der oben beschriebene Basispreis wird über einen von der Mietdauer abhängigen Faktor reduziert, der sich aus dem gewählten Miettarif (s.u.) ergibt. Beispiel: Der Basispreis für einen Opel Astra, Kategorie Mittelklasse, beträgt 80 Euro pro Tag. Der Abminderungsfaktor für einen 4- bis 6-tägigen Mietvorgang beträgt beim anzusetzenden Tarif 0,75. Für eine 5-tägige Reservierung wären demnach 300 Euro anzusetzen ($80 \text{ Euro/Tag} * 0,75 * 5 \text{ Tage} = 300 \text{ Euro}$).

Bei den meisten Reservierungen wird der Standardtarif anzusetzen sein. Ausnahmen sind z.B. günstigere Tarife für Firmenkunden und (befristete) Sonderangebote, ... (weitere Details spielen hier keine Rolle). Wir gehen zunächst der Einfachheit halber davon aus, dass für jedes Fahrzeug prinzipiell jeder Tarif möglich ist. "

Hinweise:

- Gehen Sie davon aus, dass alle von Ihnen benötigten Operationen in den entsprechenden Klassen vorhanden sind bzw. hinzugefügt werden können.
- Die in Abb. 1 aufgeführten Operationen müssen nicht weiter spezifiziert werden. Sie können diese als gegeben ansehen.
- Falls Sie neben den in Abb. 2, 3 und 4 gegebenen Objekten noch weitere benötigen, können Sie diese in Ihrer Lösung ergänzen.
- Beim Aufruf von Klassenoperationen können Sie die Pfeile zur Lebenslinie einer Instanz dieser Klasse zeichnen. Sie müssen die Klassenobjekte (z.B. Kunde:Class) nicht separat in Ihr Diagramm mit aufnehmen.
- Beachten Sie auch die Informationen in den Textnotizen der gegebenen Diagramme
- Das Entwurfsprinzip des Arbeitens auf Kopien muss hier nicht berücksichtigt werden.
- Gehen Sie davon aus, dass der Akteur nur gültige Eingaben vornehmen kann.
 - Teil a: nur zulässige Datumsangaben.
 - Teil b: nur gültige Kundennummern, d.h. Nummern, zu denen Kunden existieren.
- Entsprechende Prüfungen müssen nicht modelliert werden.
- Wir empfehlen, die Lösung zunächst zu skizzieren und erst dann „sauber“ zu zeichnen, wenn der Ablauf und die Abmessungen klar sind. Für Ihre Skizzen können Sie beispielsweise die mit den Abbildungen 3 bzw. 4 identischen Abbildungen 3a und 4a verwenden.

Aufgabenstellung:

- a) Es soll der Betrag berechnet werden, der für einen einzugebenden Mietvorgang fällig wird. Der Akteur gibt zunächst den Mietbeginn und das Mietende ein. Anhand dieser Informationen werden die in diesem Zeitraum zur Verfügung stehenden Fahrzeuge ermittelt und dem Akteur in Form einer Auswahlliste präsentiert. Dieser wählt nun einen Eintrag aus, sowie einen der möglichen Tarife, und schließt die Eingabe mit dem Anklicken des Berechnen-Buttons ab. Danach wird die Berechnung durchgeführt. Ihre Aufgabe ist es, die beschriebenen Vorgänge in Form eines Sequenzdiagramms zu modellieren. Verwenden Sie für Ihre Lösung die Vorlage aus Abb. 3. Eine Übersicht über das gesamte Szenario finden Sie in Abb. 2. Der Berechnungsvorgang schließt damit ab, dass dem Akteur der berechnete Betrag zusammen mit den von ihm getätigten Eingaben in der Eingabemaske angezeigt werden („präsentiere Betrag und Eingaben“).

- b) Ergänzen sie die Abbildung 2 um den Ablauf der Erzeugung des Mietvorgangs. Verwenden Sie dafür die Vorlage aus Abb. 4. Orientieren Sie sich am Klassenmodell in Abb. 1. Gehen Sie der Einfachheit halber davon aus, dass der Akteur eine gültige Kundennummer eingibt und keine entsprechende Prüfung stattfindet.
- c) Normalerweise wird nach Kurstext im Grobentwurf auf Kopien der Entitätsobjekte gearbeitet, diese Entwurfsentscheidung sollte hier allerdings nicht berücksichtigt werden. Die Umsetzung dieses Entwurfsprinzips hat unter anderem zur Folge, dass die Objekte der AAS-Klassen die „echten“ Entitätsobjekte nicht kennen. Warum ist das wünschenswert bzw. welchen Nachteil hat die Modellierung in Abb. 2 und den Teilaufgaben a und b (ca. 3 Sätze)?

Kurs 1793 "Software Engineering I"

Klausur am 11.8.2007

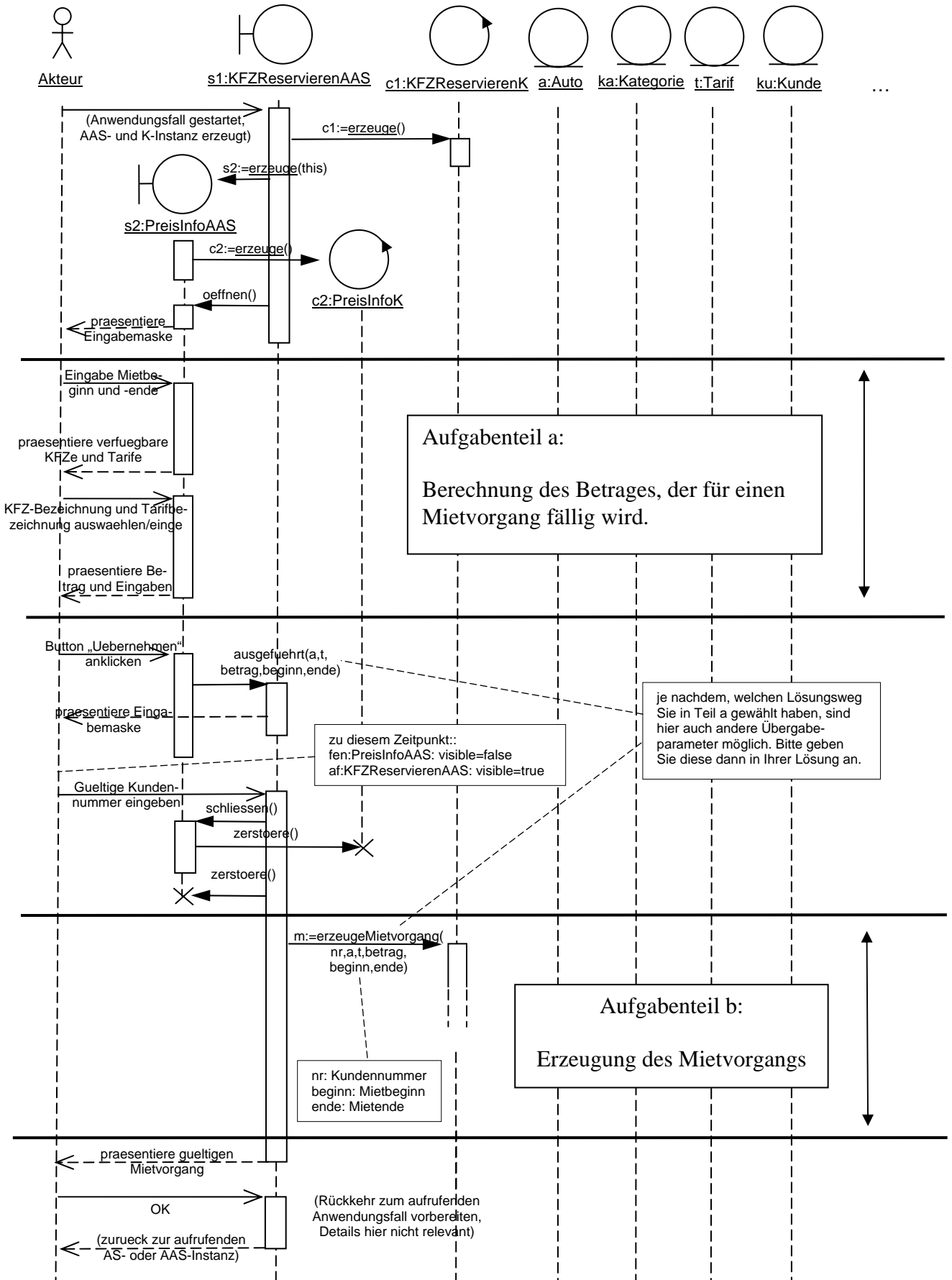


Abb. 2: Übersicht über das in Aufgabe 2 zu bearbeitende Szenario

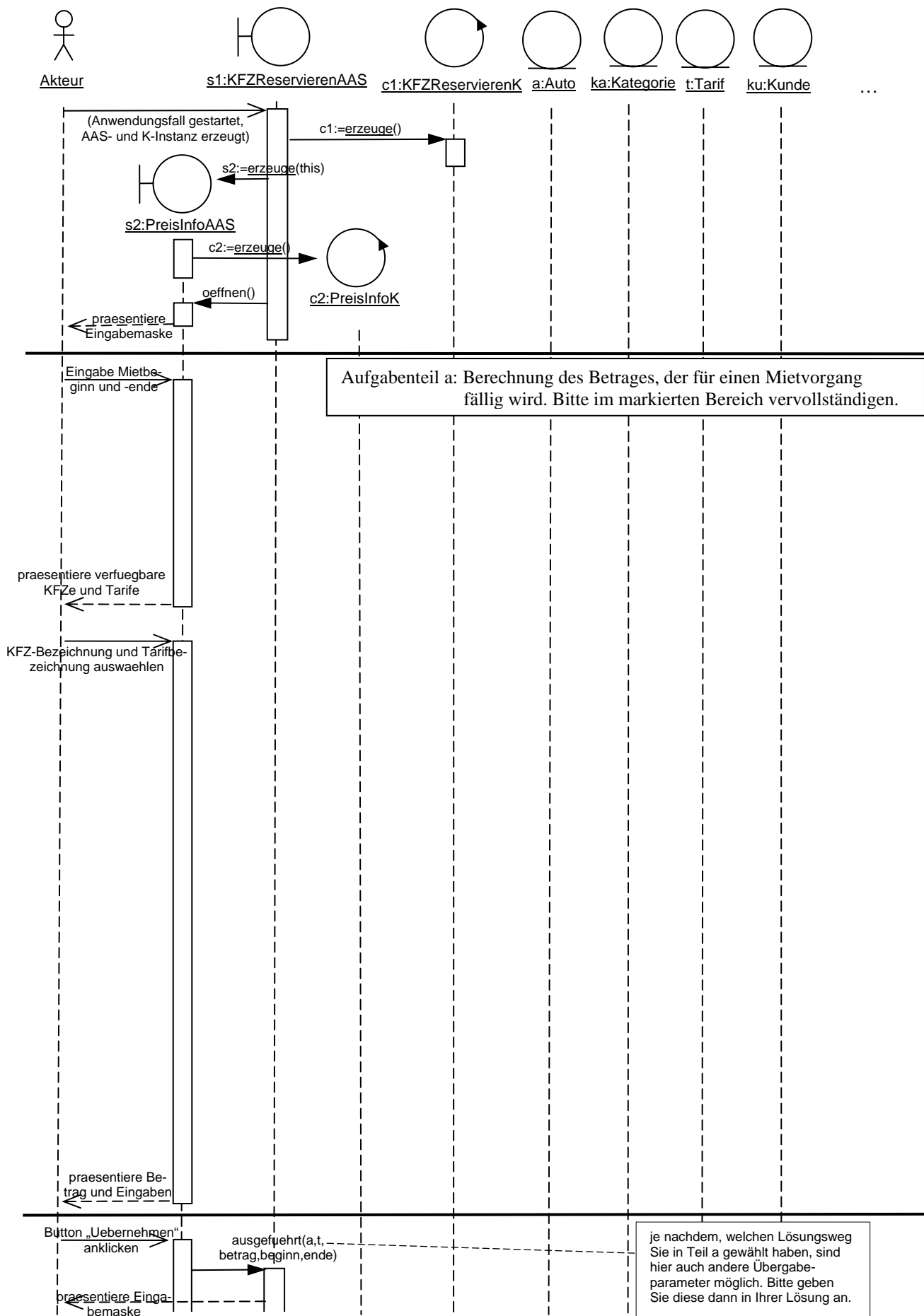


Abb. 3: zu vervollständigendes Sequenzdiagramm zur Aufgabe 2a

Kurs 1793 "Software Engineering I"

Klausur am 11.8.2007

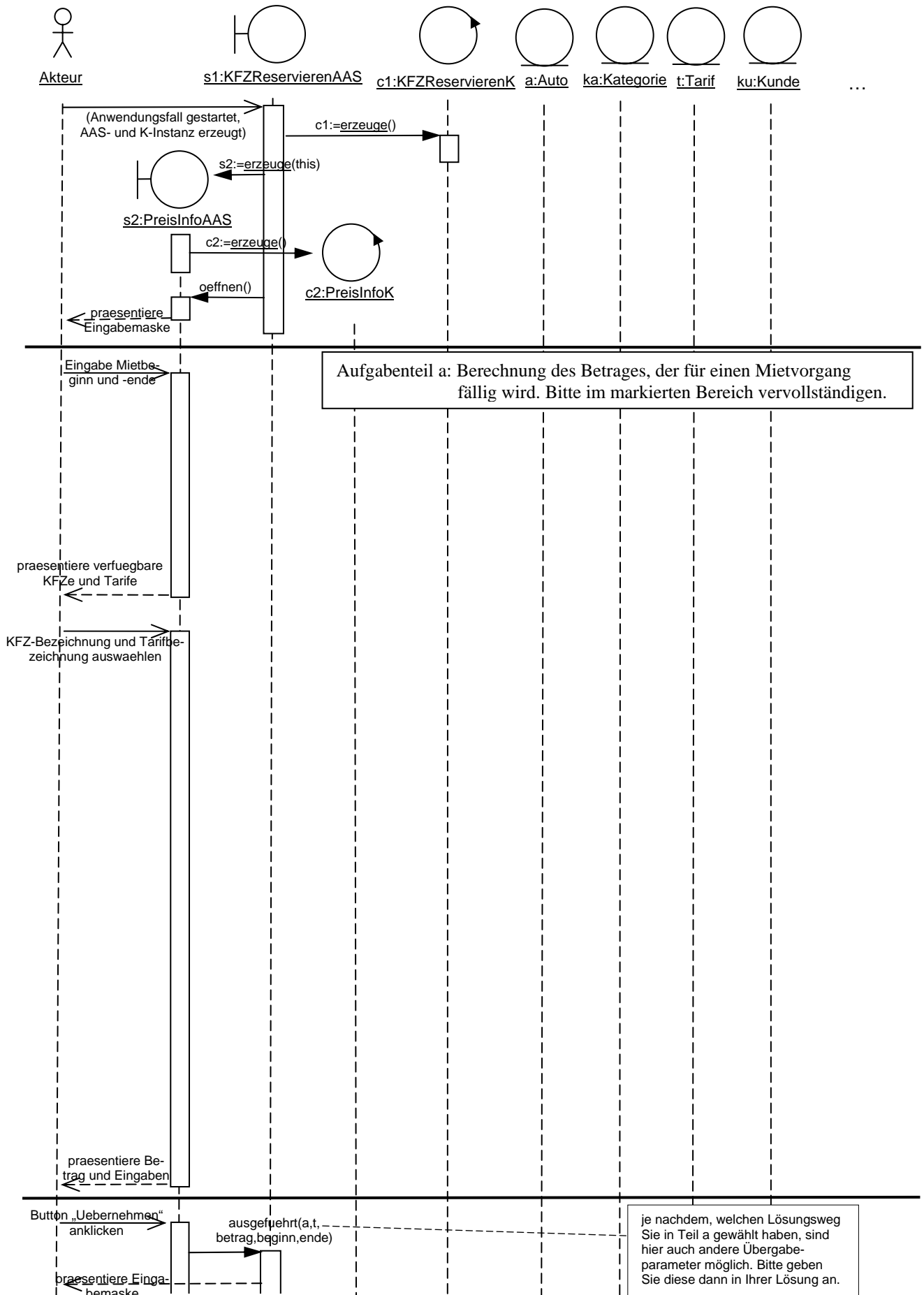


Abb. 3a: zu vervollständigendes Sequenzdiagramm zur Aufgabe 2a (Konzeptpapier)

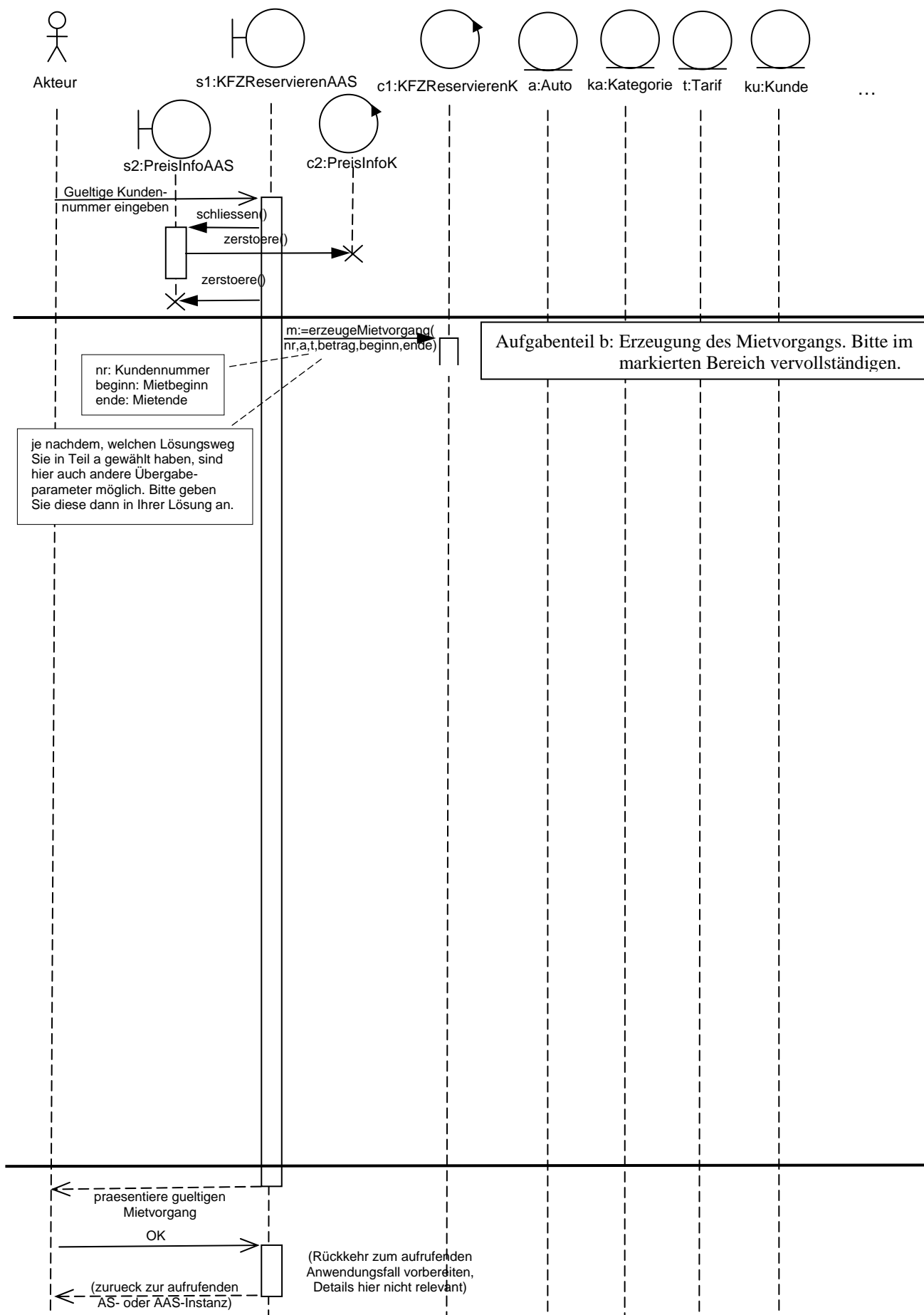


Abb. 4: zu vervollständigendes Sequenzdiagramm zur Aufgabe 2b

Kurs 1793 "Software Engineering I"

Klausur am 11.8.2007

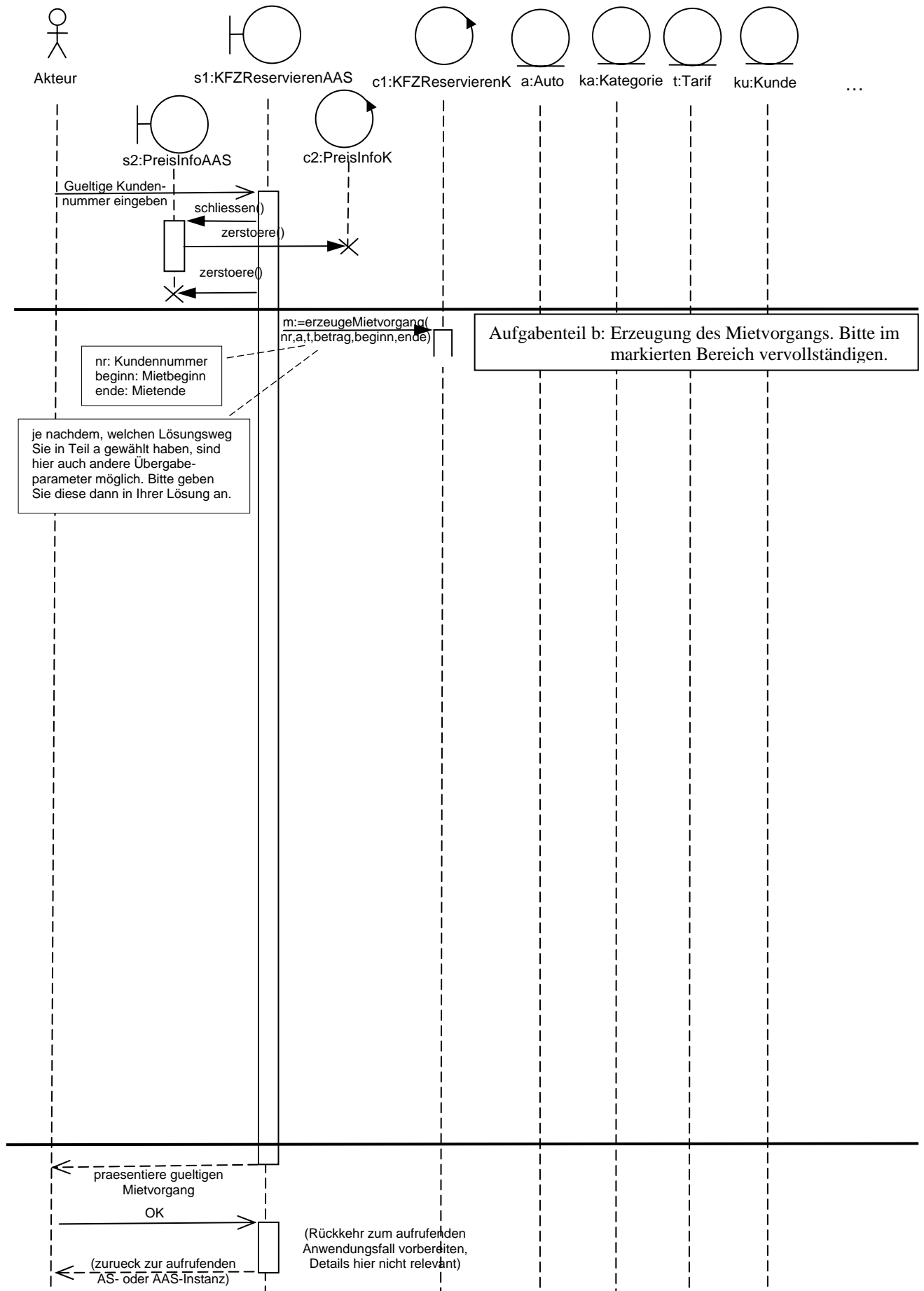


Abb. 4a: zu vervollständigendes Sequenzdiagramm zur Aufgabe 2b (Konzeptpapier)

Aufgabe 3 (20 Punkte)

Erstellen Sie ein **Zustandsdiagramm**, das den Lebenszyklus eines PKWs unter den folgenden Bedingungen modelliert:

Ein PKW steht zunächst neu und unbenutzt beim Händler und ist vom TÜV für den Verkehr zugelassen. Das Fahrzeug ist sofort einsatzbereit, d.h. der neue Besitzer kann unmittelbar nach dem Kauf einsteigen und losfahren (Zulassung und Haftpflichtversicherung müssen bei der Modellierung nicht berücksichtigt werden).

Die Nutzung des PKWs kann dadurch unterbrochen werden, dass ein Schaden durch Verschleiß eintritt, der Besitzer einen Unfall baut oder die Zweijahresfrist für die nächste Hauptuntersuchung beim TÜV überschreitet. In diesen Fällen kann oder darf das Fahrzeug nicht mehr benutzt werden.

Im Fall der Fristüberschreitung kann der Besitzer den PKW selbst beim TÜV vorfahren. Wird das Fahrzeug dort als nicht verkehrssicher eingestuft, muss er es in die Werkstatt bringen oder stilllegen. In der Werkstatt wird dann die Verkehrssicherheit wieder hergestellt und die TÜV-Plakette vergeben.

Im Fall eines Unfalls oder Schadens muss sich der Besitzer überlegen, ob sich eine Reparatur lohnt. Entscheidet er sich dafür, kann er den Wagen in einer Werkstatt reparieren lassen. Andernfalls kann er das Fahrzeug auch stilllegen, womit der Lebenszyklus des PKW endet.

Hinweise:

Gehen Sie von folgenden vereinfachenden Annahmen aus:

- Der Besitzer repariert sein Fahrzeug nicht selbst.
- Wird ein Fahrzeug in eine Werkstatt gebracht, dann gelingt die Reparatur oder die Abnahme der Hauptuntersuchung in jedem Fall.
- Eine vorübergehende Stilllegung ist nicht vorgesehen.

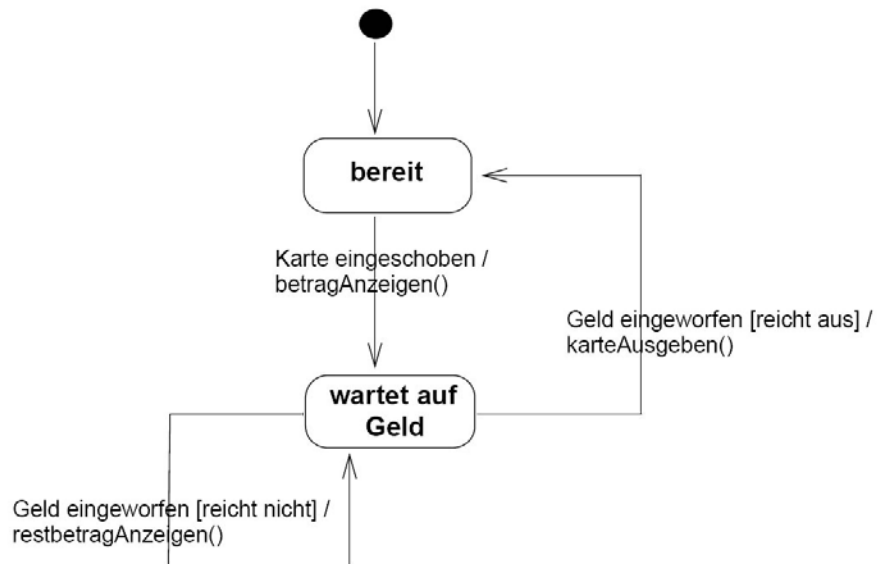
Kurs 1793 "Software Engineering I"

Klausur am 11.8.2007

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

Ereignis [Wächterbedingung] / Aktionsfolge

Beispiel: Kartenautomat im Parkhaus



Aufgabe 4 (24 Punkte)

Erstellen Sie, ausgehend von Abb. 5, ein *Feinentwurfs-Klassendiagramm* durch Anwendung der im Kurstext beschriebenen Transformationsregeln. Beachten Sie dabei die folgenden Punkte:

- Die Assoziationen "GruppenInRunde", "Zuteilung" und "Gruppenspiel" brauchen Sie **nicht** zu transformieren.
- 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.

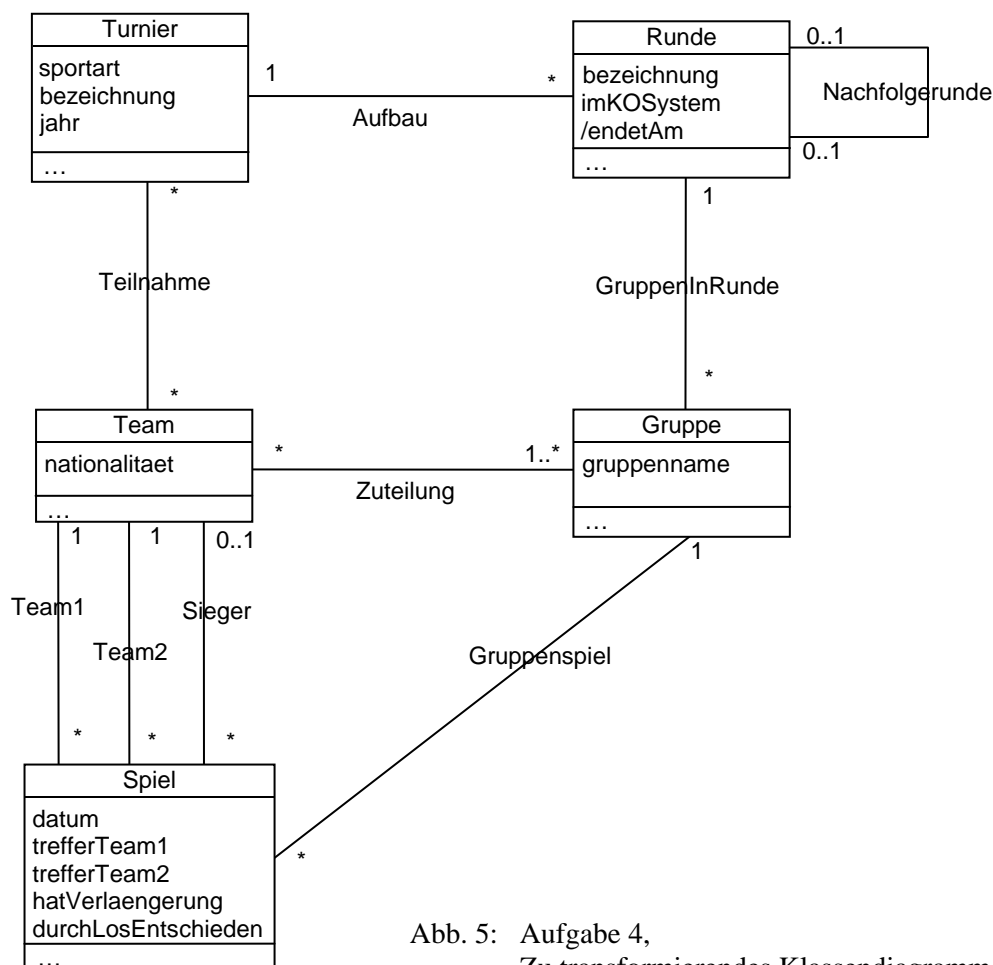


Abb. 5: Aufgabe 4,
Zu transformierendes Klassendiagramm

Kurs 1793 "Software Engineering I"Klausur am 11.8.2007

Zum besseren Verständnis des Klassendiagramms aus Abb. 5 finden Sie im Folgenden eine Beschreibung der zugehörigen Domäne. Für die Lösung der Aufgabe ist deren Verständnis jedoch nicht unbedingt erforderlich.

„Ein internationales Turnier wird durch die zugehörige Sportart, eine aussagekräftige Bezeichnung (z.B. "Weltmeisterschaft") und das Jahr beschrieben, in dem es stattfindet. Zu jedem teilnehmenden Team wird die Nationalität festgehalten.

Ein Turnier wird in verschiedene Runden (z.B. Runde 1, Runde 2, Viertelfinale, Halbfinale, Finale) unterteilt. In den Vorrunden (z.B. Runde 1, Runde 2) werden die qualifizierten Mannschaften in mehrere Gruppen aufgeteilt, von denen jede über einen eindeutigen Gruppennamen identifiziert wird. Beispielsweise könnte man der Runde 1 die Gruppen "Runde1GruppeA", ..., "Runde1GruppeF" zuordnen. In den Endrunden (z.B. Viertelfinale, Halbfinale, Finale) gibt es jeweils nur eine Gruppe, beispielsweise im Halbfinale die Gruppe "Halbfinalisten".

In den Vorrunden spielt innerhalb einer Gruppe jeder gegen jeden. Anhand der Spielergebnisse werden dann die Teams ermittelt, die am besten abgeschnitten haben. Diese qualifizieren sich für die nächste Runde und werden einer der Gruppen für die nächste Runde zugeteilt. Die Mannschaften, die sich nicht qualifizieren konnten, scheiden aus dem Turnier aus.

In den Endrunden wird jeder Mannschaft nur ein Gegner zugeteilt. Die Sieger der Begegnungen kommen eine Runde weiter, die Verlierer scheiden aus.

Ein Spiel findet an einem bestimmten Datum statt. Das Ergebnis eines Spiels ergibt sich anhand der Treffer (Tore, Körbe, ...), die von der jeweiligen Mannschaft erzielt wurden. In den Vorrunden ist es möglich, dass ein Spiel unentschieden ausgeht, d.h. dass beide Teams gleich viele Treffer erzielt haben. In den Endrunden ist dies nicht möglich, daher wird im Fall eines Unentschiedens nach der regulären Spielzeit eine Verlängerung angesetzt. Bringt auch diese keine Entscheidung, bestimmt das Los, wer als Sieger aus der Begegnung hervorgeht. Als solches wird auch ein Elfmeterschießen o.ä. angesehen, Treffer im Elfmeterschießen zählen jedoch nicht zum Spielergebnis.“