

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"Klausur am 13.8.2005

**Sommersemester 2005
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,
- 1 Formblatt für eine Bescheinigung für das Finanzamt,
- diese Hinweise zur Bearbeitung,
- 4 Aufgaben (Seite 3 - Seite 14)

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!

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"

Klausur am 13.8.2005

Aufgabe 1 (8 + 11 + 9 + 7 = 35 Punkte)

Gegeben sind die folgenden Ausschnitte einer Anforderungsspezifikation für eine Software zur Verwaltung einer Verkehrssünderkartei. Ihre Aufgabe ist es, darauf aufbauend einige Tätigkeiten der *Analyse* durchzuführen.

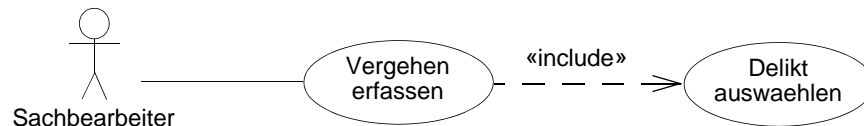


Abb. 1 Ausschnitt aus dem Anwendungsfalldiagramm

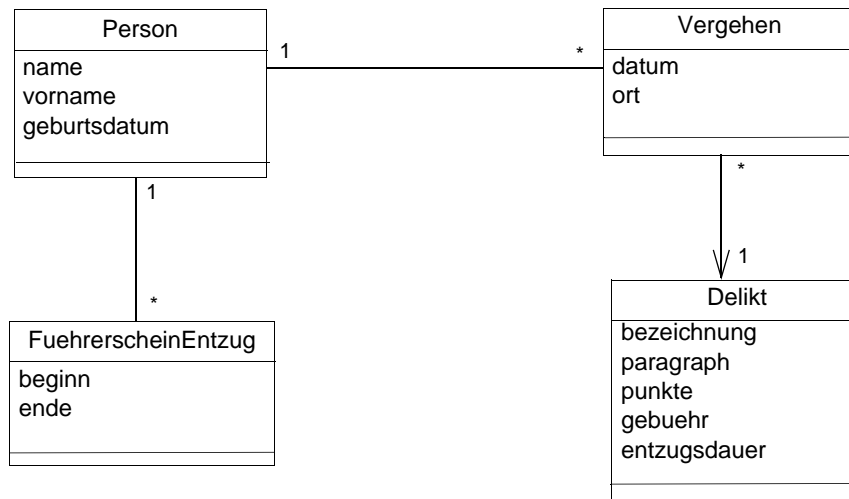


Abb. 2 Ausschnitt aus dem Domänen-Klassendiagramm

Textuelle Spezifikationen der Anwendungsfälle "Vergehen erfassen" und "Delikt auswaehlen":

use case Vergehen erfassen

actors

Sachbearbeiter

precondition

Die Person, die das zu erfassende Vergehen begangen hat, ist bereits ausgewählt. Deren Daten sind überprüft und aktuell.

main flow

Der Sachbearbeiter wählt zunächst das begangene Delikt aus einem Katalog aus (include "Delikt auswaehlen"). Nach erfolgter Auswahl gibt er im Bearbeitungsfenster die Daten für das zu speichernde Vergehen ein. Im Bearbeitungsfenster sollen aus Gründen der Anschaulichkeit auch relevante Attribute der betreffenden Person und des ausgewählten Delikts angezeigt werden. Nach Abschluss des Eingabevorgangs wird das Vergehen im System registriert.

postcondition

Das Vergehen ist erfasst, die Daten des Verkehrssünder sind aktuell.

end Vergehen erfassen

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"
Klausur am 13.8.2005

use case Delikt auswählen

actors

Sachbearbeiter

precondition

-

main flow

Der Sachbearbeiter wählt im Auswahlfenster zunächst aus einer Liste von Delikten eines aus. Die Attribute des markierten Delikts werden dem Sachbearbeiter daraufhin im Auswahlfenster angezeigt. Kommt der Sachbearbeiter zu dem Schluss, das richtige Delikt markiert zu haben, beendet er den Auswahlvorgang.

postcondition

Ein Delikt wurde ausgewählt.

end Delikt auswählen

Anmerkung: Auf die Angabe der Exceptional Flows für nicht erfolgreiche Erfassungs- und Auswahlvorgänge wurde hier verzichtet.

Aufgabenstellung:

Hinweis: Die einzelnen Teilaufgaben können weitgehend unabhängig voneinander gelöst werden.

- a) Zeichnen Sie einen Mechanismus zur Realisierung des Anwendungsfalls "Vergehen erfassen".
- b) Zeichnen Sie ein Sequenzdiagramm für das Szenario "Vergehen erfassen - erfolgreich" des Anwendungsfalls "Vergehen erfassen".

Gehen Sie dabei davon aus, dass die Instanz des betreffenden Verkehrssünder bereits ausgewählt wurde und bei der Erzeugung des Bearbeitungsfensters zum Anwendungsfall "Vergehen erfassen" als Parameter übergeben wird.

Bei der Detaillierung Ihres Diagramms können Sie sich an Abb. 3 auf Seite 6 orientieren. Sie brauchen keine Aktivierungsbalken darzustellen. Bitte geben Sie aber, im Unterschied zur Vorlage, bei der Erzeugung von Entitätsobjekten auch die Standardoperationen zur Realisierung der Assoziationen mit an.

Hinweis:

Bei dem zu Abb. 3 gehörenden Anwendungsfalldiagramm war der Vorgang der Firmenauswahl nicht als separater Anwendungsfall modelliert worden.

- c) Die Funktionalität des Anwendungsfalls "Vergehen erfassen" soll erweitert werden.
"Beim Erfassungsvorgang soll automatisch überprüft werden, ob infolge des neuen Vergehens der Führerschein der betreffenden Person gesperrt werden muss. Die Prüfung soll so aussehen, dass ... (Details hier nicht von Bedeutung). Ergibt die Prüfung, dass keine Sperre (bzw. Entzug) des Führerscheins erforderlich wird, endet der Anwendungsfall wie gehabt. Andernfalls muss eine Sperre erzeugt und deren Eigenschaften eingegeben werden. ... (auf die Eigenschaften der Sperre wird hier nicht im Detail eingegangen.)"

Ergänzen Sie das Anwendungsfalldiagramm aus Abb. 1 entsprechend. Überarbeiten Sie die gegebene textuelle Spezifikation an den Stellen, an denen dies infolge der Erweiterungen not-

Kurs 1793 “Software Engineering I - Grundkonzepte der OOSE”Klausur am 13.8.2005

wendig wird. Dies beinhaltet auch die Spezifikation evtl. neu dazugekommener Elemente.

Hinweis:

Bei der Prüfung, ob eine Sperre erfolgen muss, können Sie sich in den betreffenden Flows auf Angaben beschränken wie "jetzt findet die Prüfung gemäß den Vorgaben statt". Gleiches gilt für die Festlegung der Eigenschaften der Sperre.

- d) Erstellen Sie ein Sequenzdiagramm, das die Vorgaben aus Teilaufgabe c berücksichtigt und das Szenario "Vergehen erfassen erfolgreich, Führerscheinenzug erforderlich" behandelt. Gehen Sie dabei nicht auf Details des Prüfungsvorgangs ein. Knüpfen Sie an Ihre Lösung zur Teilaufgabe b an und zeichnen Sie nur die Abläufe, die sich davon unterscheiden bzw. neu dazukommen.

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"

Klausur am 13.8.2005

Das in Abb. 3 dargestellte Sequenzdiagramm ist als Vorlage für die Detaillierung des in Aufgabenteil c anzufertigenden Sequenzdiagramms gedacht.

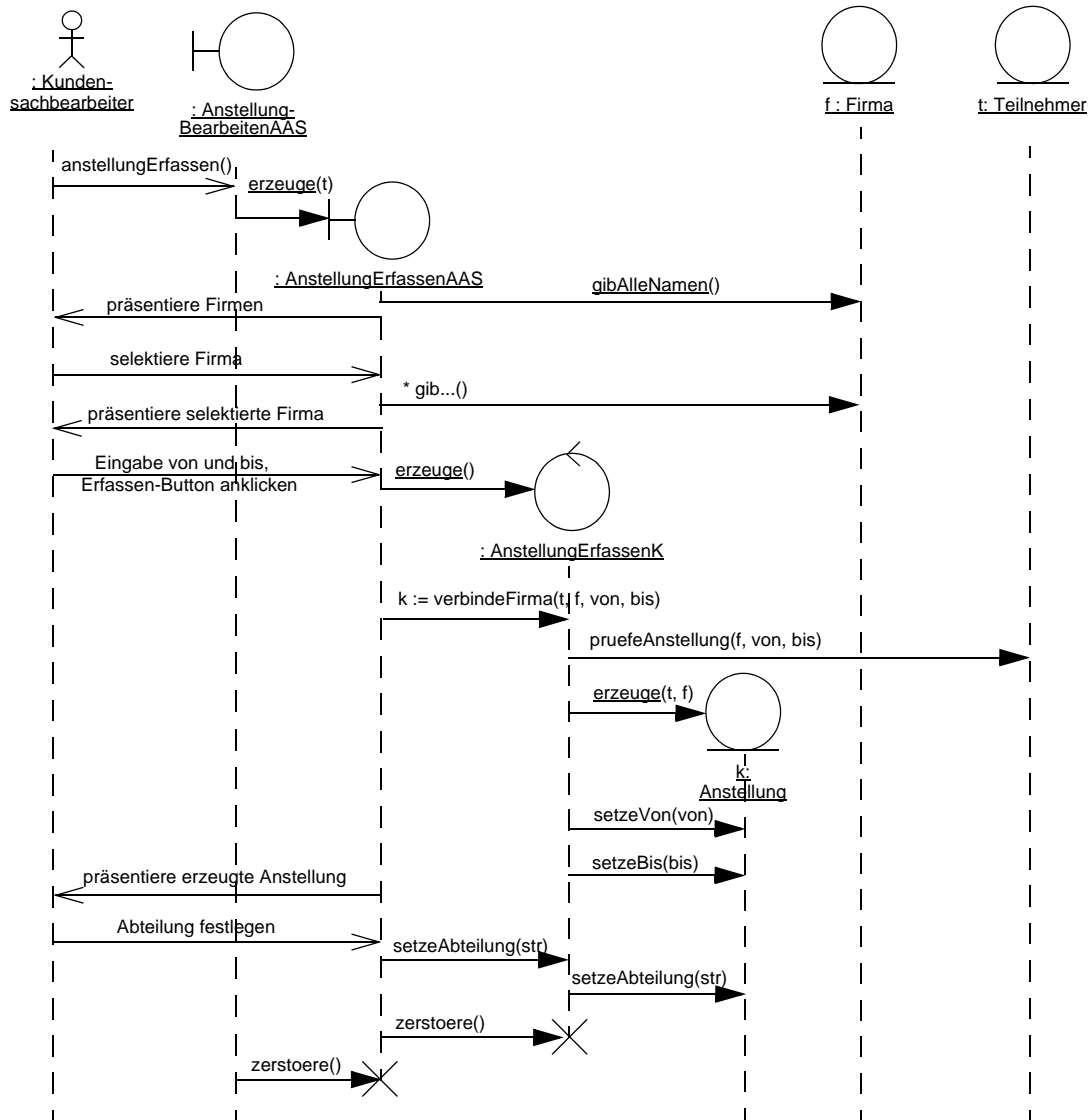


Abb. 3 Beispiel-Sequenzdiagramm

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"Klausur am 13.8.2005

Aufgabe 2 (20 + 4 = 24 Punkte)

Aus aktuellem Anlass möchte ein großer deutscher Medienkonzern dem unentschlossenen Wähler eine Software online zur Verfügung stellen, die ihn dabei unterstützt, sich im Parteiendschungel zurechtzufinden und eine Wahlentscheidung zu treffen. Ihre Aufgabe wird es dabei sein, ein **Domänenklassenmodell** zu erstellen, auf dessen Basis die unten genannten Anforderungen realisiert werden können:

"Aus Sicht des Benutzers sieht der Ablauf des Vorgangs folgendermaßen aus:

- Zunächst wählt er aus einer Liste von hierarchisch organisierten Themenbereichen denjenigen aus, der ihn besonders interessiert, z.B. Europapolitik.
- Dem Benutzer werden dann verschiedene Aussagen zu dem gewählten Themenbereich angezeigt, eine solche Aussage könnte z.B. sein: "Die Türkei soll EU-Mitglied werden." Diese Aussage bewertet der Benutzer dann in Form von Zahlen zwischen 0 und 5 (5 = volle Zustimmung, 0 = volle Ablehnung). Dazu gibt er außerdem an, wie wichtig ihm dieses Thema ist (5 = sehr wichtig, 0 = uninteressant).
- Anhand dieser Eingaben wird dann vom System berechnet, wie groß die Übereinstimmung (bzw. die Meinungsdivergenz) mit welcher Partei für den gewählten Themenbereich ist. Auf der Grundlage dieser Berechnungen wird dem Benutzer eine Wahlempfehlung mitgeteilt (*Hinweis*: Gehen Sie im Moment davon aus, dass diese aus einer einfachen Textausgabe besteht und machen Sie sich keine Gedanken um deren Darstellung).

Um den genannten Ablauf zu ermöglichen, ist die Software folgendermaßen aufzubauen:

Jedem Themenbereich werden aus einem Pool von Aussagen diejenigen zugeordnet, die den Themenbereich inhaltlich betreffen. Eine Aussage kann dabei auch in mehreren Themenbereichen enthalten sein. Wie bereits oben beschrieben, kann ein Themenbereich auch andere Themenbereiche enthalten.

Zwischen jeder Partei und jeder Aussage wird eine Bewertung vorgenommen, die aus dem Grad der Zustimmung (5 = volle Zustimmung, 0 = volle Ablehnung) und der Wichtigkeit dieser Aussage für die Partei besteht (5 = sehr wichtig, 0 = uninteressant). Mit diesen Bewertungen werden später die Benutzereingaben verglichen, um die Partei zu ermitteln, mit der der Benutzer die geringsten Differenzen hat.

Die Software sollte so entwickelt werden, dass der Änderungsaufwand beim Hinzufügen oder Entfernen von Aussagen und Parteien möglichst gering ist.

Aus Datenschutzgründen dürfen keine persönlichen Benutzerdaten persistent gespeichert werden. Allerdings sollten für statistische Auswertungen die "anonymen" Daten einer Benutzersession festgehalten werden wie das nachgefragte Themengebiet und die abgegebenen Bewertungen."

Ergibt sich im Rahmen eines Programmablaufs die Situation, dass eine Aussage in dem ausgewählten Themenbereich mehrfach enthalten ist (z.B. in zwei Unter-Themenbereichen), dann sollte sie der

Benutzer nur einmal bewerten müssen.

Anmerkung:

Selbstverständlich ist sichergestellt, dass die Erstellung der Aussagen und der Bewertungen von einer kompetenten, unabhängigen und überparteilichen Organisation vorgenommen wird.

Aufgabenstellung:

- a) Erstellen Sie ein Domänen-Klassendiagramm für die beschriebenen Anforderungen. Geben Sie zu jeder Beziehung die Multiplizitäten an. Verteilen Sie insgesamt mindestens 5 sinnvolle Attribute auf Ihre Klassen.
- b) Beschreiben Sie anhand zweier ausgewählter Assoziationen, warum Multiplizitäten alleine nicht ausreichen, um durch die Realität vorgegebene Bedingungen bezüglich der Verbindungen zwischen Objekten wiederzugeben.

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"

Klausur am 13.8.2005

Aufgabe 3 (12 + 2 = 14 Punkte)

Gegeben ist das Kollaborationsdiagramm in Abb. 4, welches das Szenario "Vertrag erfassen - erfolgreich" einer Software zur Verwaltung von studentischen Hilfskräften (Hiwis) im Stadium des *Grobentwurfs* darstellt.

Die Schnittstellenklasse Vertrag ErfassenAAS stellt dem Benutzer eine Eingabemaske mit Textfeldern für die Vertragsdaten (Name des Hiwis, Bezeichnung der Kostenstelle, Wochenstunden, Stundenlohn, Vertragsbeginn, Vertragsende) zur Verfügung. Diese Strings werden dann der Operation vertragErfassen(String[]) als Parameter mit der Bezeichnung "eingaben" übergeben.

Anmerkung zu den übergebenen Parametern: eingaben[0] ist der Name des Hiwis, eingaben[1] die Bezeichnung der Kostenstelle.

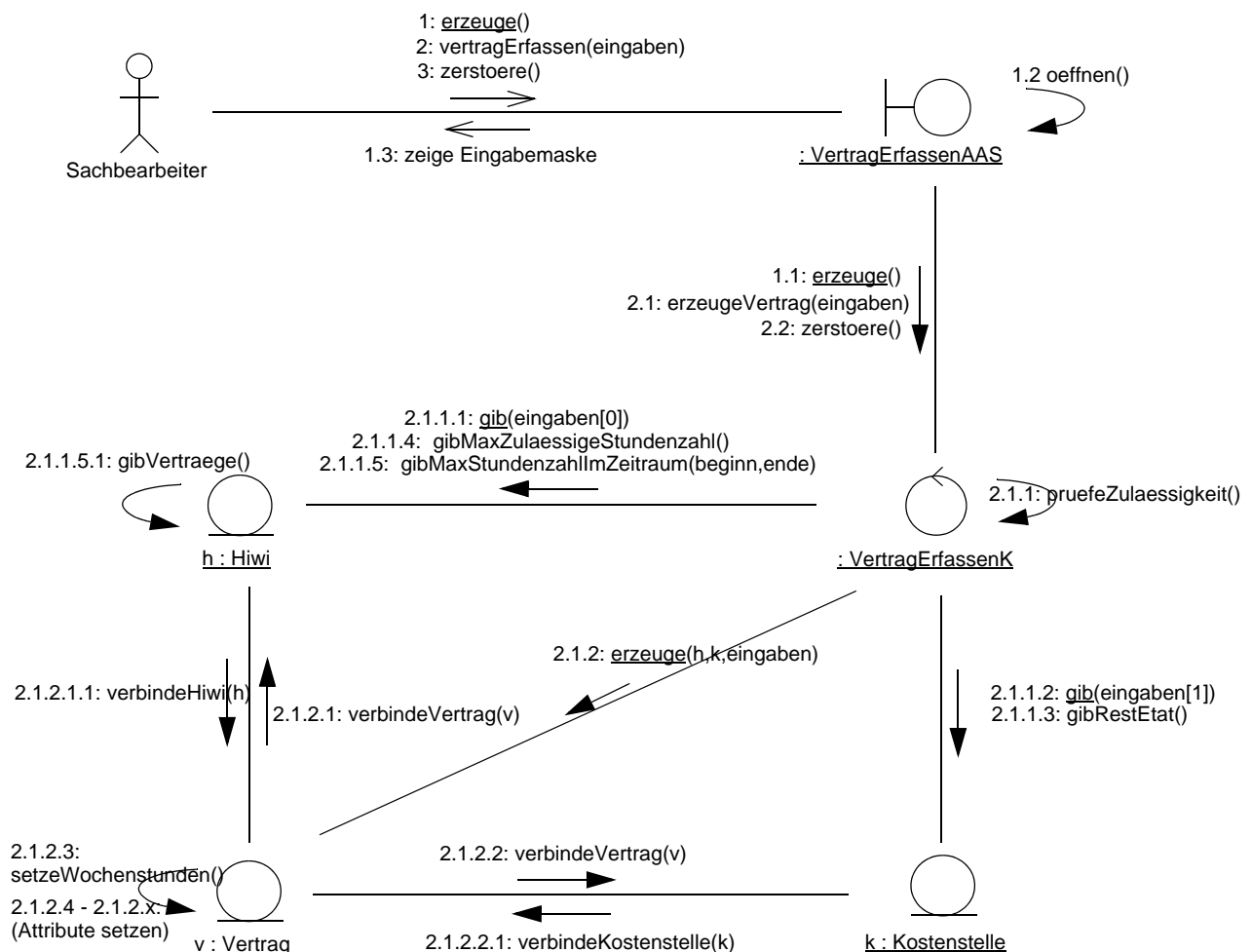


Abb. 4 Kollaborationsdiagramm zum Szenario "Vertrag erfassen - erfolgreich"

- a) Übertragen Sie das Kollaborationsdiagramm aus Abb. 4 in ein äquivalentes Sequenzdiagramm. Geben Sie dabei auch die Aktivierungsbalken mit an. Verwenden Sie dazu die Abb. 5 auf den nächsten beiden Seiten.
- b) Welche grundsätzliche, im Grobentwurf getroffene Entwurfsentscheidung wird von dem in Abb. 4 dargestellten Kollaborationsdiagramm verletzt?

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"

Klausur am 13.8.2005

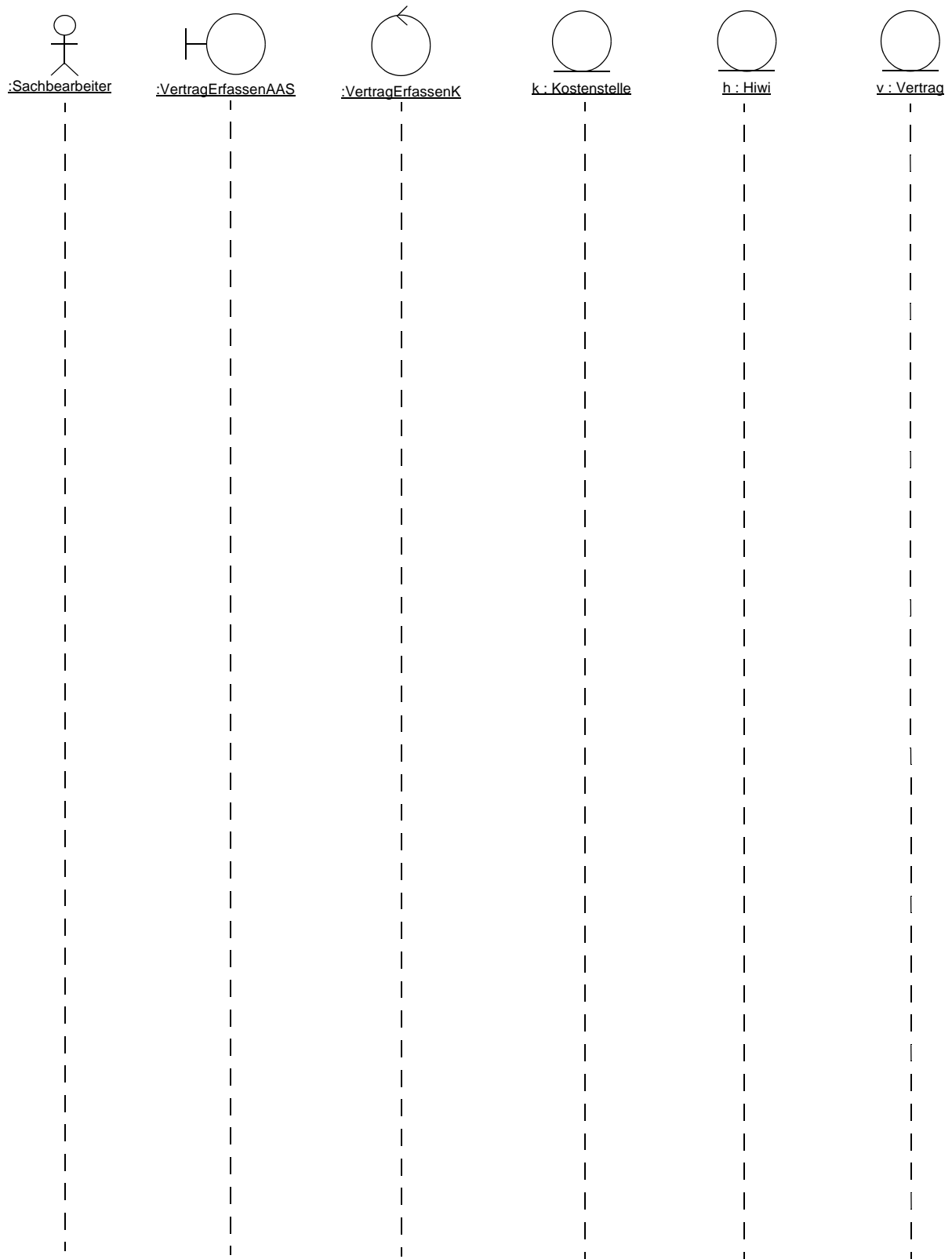


Abb. 5 Sequenzdiagramm zu Aufgabe 3a

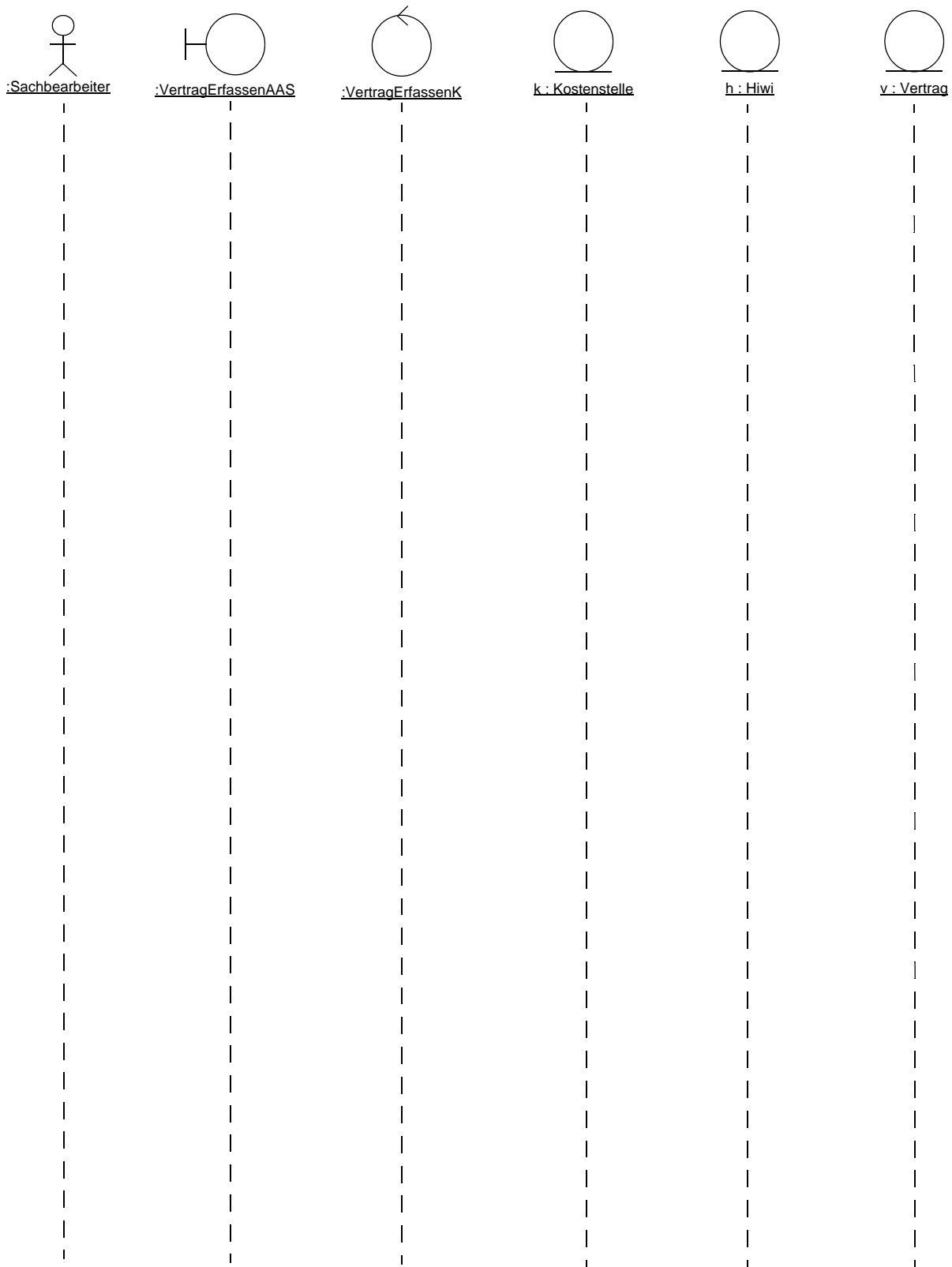


Abb. 5 Sequenzdiagramm zu Aufgabe 3a

Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"

Klausur am 13.8.2005

Aufgabe 4 (9 + 6 + 2 + 6 + 4 = 27 Punkte)

Im "Lernraum Virtuelle Universität" kann jeder Benutzer eine oder mehrere Rollen einnehmen (Student, Betreuer, Mentor, Administrator, Gast, ...), die seine Zugriffsrechte festlegen. In Abb. 6 ist dies in einem Klassendiagramm dargestellt.

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

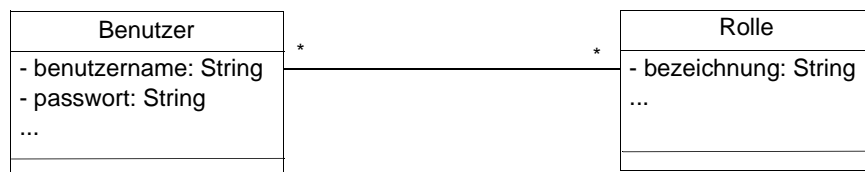


Abb. 6 Klassendiagramm zu Aufgabe 4

- a) Erstellen Sie, ausgehend von Abb. 6, ein **Feinentwurfs-Klassendiagramm** durch Anwendung der im Kurstext beschriebenen Transformationsregeln.
 - Stellen Sie in Ihrem Diagramm alle Elemente dar, die durch die Transformation der Assoziation und eventueller Echtes-Ganzes-Klassen neu hinzugekommen sind. Geben Sie zu Ihren Klassen insgesamt mindestens 10 Operationen mit Sichtbarkeiten und Parametern an.
 - Tragen Sie an den Assoziationsenden die Multiplizitäten ein.
 - Assoziationen, die sich direkt auf ein Attribut abbilden lassen, sowie von Ordner-, Relations- und Mengenklassen ausgehende Assoziationen müssen nicht weiter transformiert werden.
- b) Dem Benutzer *b* wird die Rolle *r* neu zugewiesen. Stellen Sie diesen Vorgang unter Verwendung Ihres Klassendiagramms aus Aufgabenteil a) in einem Sequenzdiagramm dar. Vervollständigen Sie dazu die Abb. 7 auf Seite 14 und geben Sie dabei alle verwendeten Standardoperationen mit an.
- c) Gehen Sie für diese und die nachfolgenden Teilaufgaben d) und e) davon aus, dass die Assoziation in Abb. 6 nicht bidirektional, sondern einseitig navigierbar von Benutzer nach Rolle ist. Was ändert sich dadurch an Ihrer Lösung zu Aufgabenteil a)?
- d) Welche Transformation wäre im Fall einer unidirektionalen Assoziation noch möglich gewesen? Welche Vorgehensweise halten Sie für sinnvoller? Skizzieren Sie die alternative Transformation und begründen Sie Ihre Aussage.
- e) Angenommen, es steht ein fester Pool an Rollen zur Verfügung. Die Rollen werden nicht mehr geändert und es werden nachträglich keine Rollen mehr hinzugefügt oder gelöscht. Ergibt sich dadurch eine andere Möglichkeit der Transformation? (Assoziation ist einseitig navigierbar von Benutzer nach Rolle.)

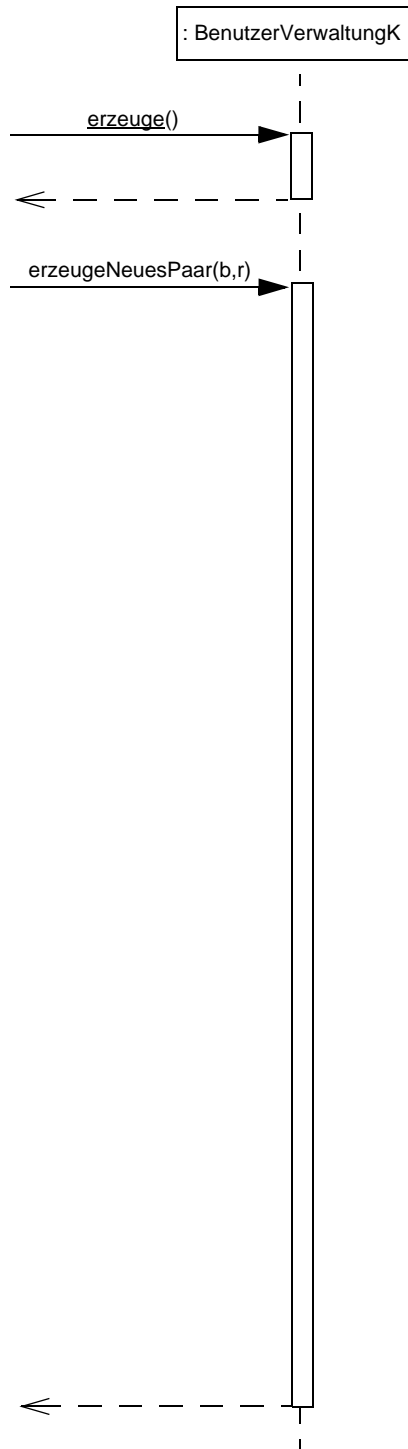


Abb. 7 Sequenzdiagramm zur Aufgabe 4b