

**Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"**Klausur am 31.7.2004

---

Sommersemester 2004

**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 10)

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 31.7.2004

---

**Aufgabe 1** (16 + 4 + 7 + 8 = 35 Punkte)

Ein Ausrichter großer Sportveranstaltungen möchte ein System entwickeln, das ihn bei den Tätigkeiten unterstützt, die im Rahmen der Durchführung internationaler Turniere in Sportarten wie Fußball oder Basketball anfallen. Ihre Aufgabe wird es dabei sein, ein Domänenklassenmodell zu erstellen, auf dessen Basis die unten genannten Anforderungen realisiert werden können.

"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."

**Aufgabenstellung:**

- a) Erstellen Sie ein redundanzfreies Klassendiagramm für die beschriebene Anwendungsdomäne. Versuchen Sie, möglichst wenige Klassen zu definieren, und geben Sie zu jeder Beziehung die Multiplizitäten an. Verteilen Sie mindestens 10 sinnvolle Attribute auf Ihre Klassen.
- 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.

- c) Skizzieren Sie ein gültiges Objektdiagramm (mit Attributwerten) für die folgende Situation:  
Beim ersten Spiel der Fußball-WM 2006 am 25.5.2006 trafen Deutschland und Österreich in der Gruppe B der Vorrunde aufeinander. Das Spiel endete 1:1 unentschieden.
- d) Erweitern Sie das in Teil a) erstellte Klassendiagramm für die folgenden Anforderungen:  
Ein Büro für Sportwetten möchte seinen Kunden die Möglichkeit anbieten, online Wetten für Sportveranstaltungen abzugeben.  
Eine Wette ist gekennzeichnet durch einen Auftraggeber, den eingesetzten Betrag, das Datum und eine Quote (Faktor, mit dem bei Eintreten des gewetteten Ereignisses der Einsatz multipliziert wird). Wetten der folgenden Art muss das System anbieten können:
- Der Kunde tippt, welche Mannschaft ein bestimmtes Spiel gewinnt oder ob es unentschieden ausgeht. Auf konkrete Trefferzahlen kann nicht gewettet werden.  
Beispiel: Vorrundenspiel Deutschland-Österreich endet unentschieden.
  - Der Kunde tippt, ob ein Team eine bestimmte Runde erreicht.  
Beispiel: Norwegen erreicht das Halbfinale.

Wetten dürfen auch während des laufenden Turniers noch abgeschlossen werden, sofern sie sich auf zukünftige Ereignisse beziehen. Beispielsweise muss es möglich sein, nach Feststehen der Endspielteilnehmer auf einen Sieg einer dieser Mannschaften zu tippen.

**Kurs 1793 “Software Engineering I - Grundkonzepte der OOSE”**Klausur am 31.7.2004

---

**Aufgabe 2 (15 Punkte)**

In dieser Aufgabe sollen Sie den Lebenszyklus eines Buches in einer Bibliothek mit Hilfe eines Zustandsdiagrammes modellieren.

Von einem neuen Buch, das in den Bestand einer Bibliothek aufgenommen wird, erfasst man zunächst die relevanten Daten. Danach ist es für die Interessenten zugänglich und kann ausgeliehen werden. Es gibt zwei verschiedene Arten der Ausleihe: die Normal- und die Projektausleihe.

Bei der Normalausleihe darf der Leser das Buch für eine bestimmte Zeit mit nach Hause nehmen. Auf Antrag kann die Ausleihfrist verlängert werden. Hat ein Besucher Interesse an einem Buch, das momentan über Normalausleihe verliehen ist, so hat er die Möglichkeit, sich dieses vormerken zu lassen. Der derzeitige Ausleiher kann die Ausleihfrist dann nicht mehr verlängern.

Bei der Projektausleihe richtet sich die Ausleihfrist nach der Laufzeit eines Projektes. Da es erfahrungsgemäß schwierig ist, die Dauer eines Projektes abzuschätzen, sind hierbei beliebig viele Fristverlängerungen möglich. Bücher in Projektausleihe kann man nicht vormerken lassen.

Bei der Rückgabe eines Buches wird zunächst überprüft, ob es beschädigt ist. Beschädigte Bücher werden aus dem Verkehr gezogen, mit den anderen wird wie folgt verfahren: Nicht vorgemerkte Bücher werden wieder an der vorgesehenen Stelle in der Bibliothek positioniert und stehen für eine erneute Ausleihe zur Verfügung. Vorgemerkte Bücher werden über einen bestimmten Zeitraum für den entsprechenden Interessenten zur Abholung bereitgelegt, außerdem wird dieser schriftlich über den Eingang des von ihm bestellten Buches informiert. Falls der Interessent das Buch nicht fristgemäß abholt, wird es wie ein nicht vorgemerktes Buch behandelt.

Vereinfachend können Sie von folgenden Annahmen ausgehen:

- Von jedem Buch ist nur ein Exemplar vorhanden.
- Alle Bücher sind ausleihbar (d.h. es gibt keinen Präsenzbestand)

**Aufgabenstellung:**

Erstellen Sie ein Zustandsdiagramm, das den Lebenszyklus eines Buches unter den gegebenen Bedingungen modelliert.

Ihr Diagramm sollte die folgenden Ereignisse berücksichtigen: Vormerkwunsch, Leser holt vorgemerktes Buch ab, Rückgabe, neues Buch liegt vor, Fristverlängerung, Beschädigung festgestellt, Ausleihwunsch, Abholfrist abgelaufen.

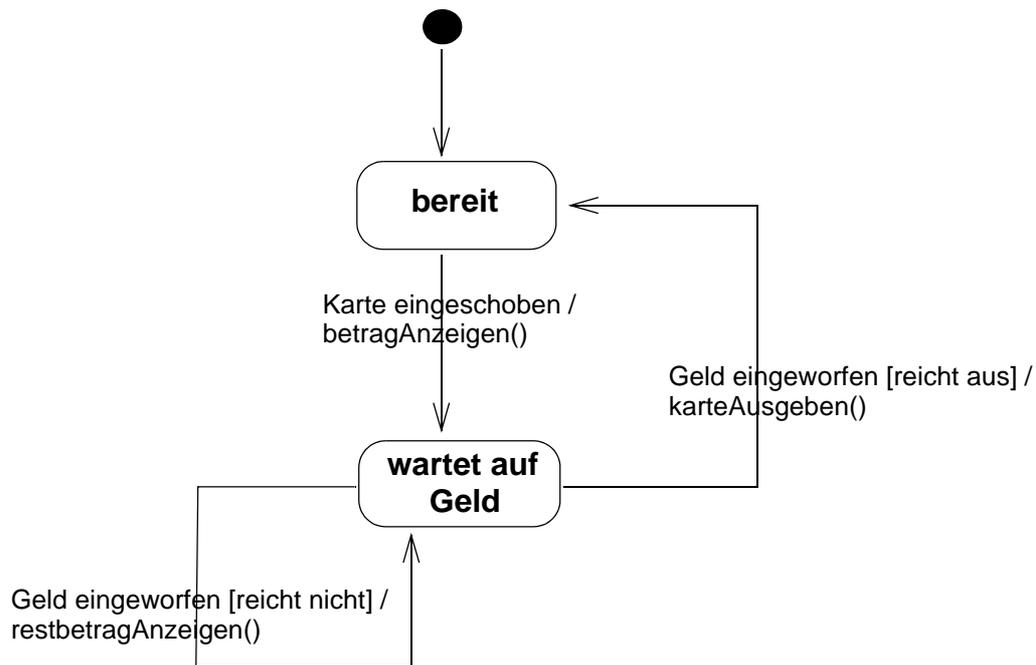
***Hinweis:***

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

*Ereignis [Wächterbedingung] / Aktionsfolge*

Beispiel:

Kartenautomat im Parkhaus



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

Klausur am 31.7.2004

**Aufgabe 3** (8 + 7 + 7 = 22 Punkte)

Das Entitäts-Klassendiagramm in Abb. 1 modelliert verschiedene Beziehungen zwischen den Angestellten einer Firma. Zwei Angestellte heißen *Projektkollegen*, wenn Sie zu irgendeinem Zeitpunkt am gleichen Projekt gearbeitet haben.

Ihre Aufgabe ist es, dieses Klassendiagramm in ein Feinentwurfs-Klassendiagramm zu transformieren, das direkt in Java implementiert werden kann. Alle Assoziationen sind bidirektional.

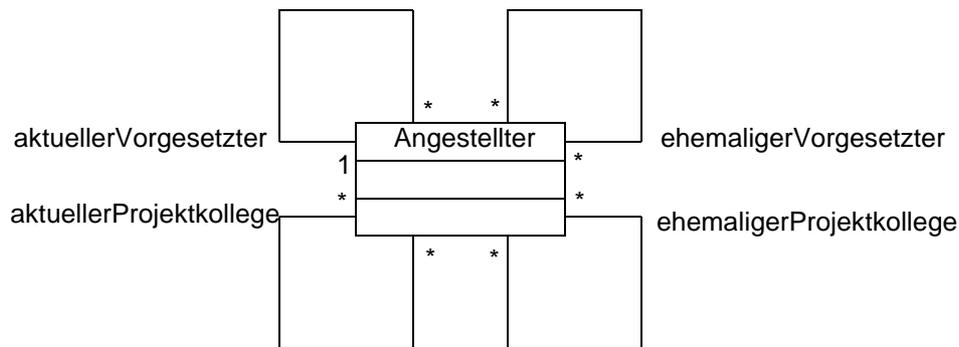


Abb. 1 Entitäts-Klassendiagramm

- a) Transformieren Sie zunächst den in Abb. 2 dargestellten Ausschnitt des Klassendiagramms. Verwenden Sie dabei Transformationen aus dem Kurstext, die ohne Paarklassen auskommen.

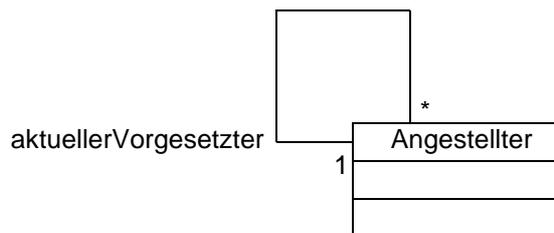


Abb. 2 Ausschnitt des Entitäts-Klassendiagramms

- b) Ergänzen Sie alle Ihre Klassen um die Operationen, die zum Erstellen und Lösen von Verbindungen notwendig sind.
- c) Transformieren Sie nun die restlichen 3 Assoziationen aus Abb. 1, ohne Paarklassen zu verwenden. Vermeiden Sie dabei die Definition redundanter Klassen und Assoziationen. Auf die Angabe von Operationen dürfen Sie verzichten.

**Aufgabe 4** (10 + 1 + 6 + 10 + 1 = 28 Punkte)

In dieser Aufgabe sollen Sie zwei Entwürfe vergleichen, indem Sie die Abhängigkeiten mit Hilfe von Sequenz- und Kollaborationsdiagrammen untersuchen.

Gegeben ist das Klassendiagramm in Abb. 3. Die beiden Akteur-Anwendungsfallschnittstellenklassen entsprechen den Fenstern in Abb. 4.

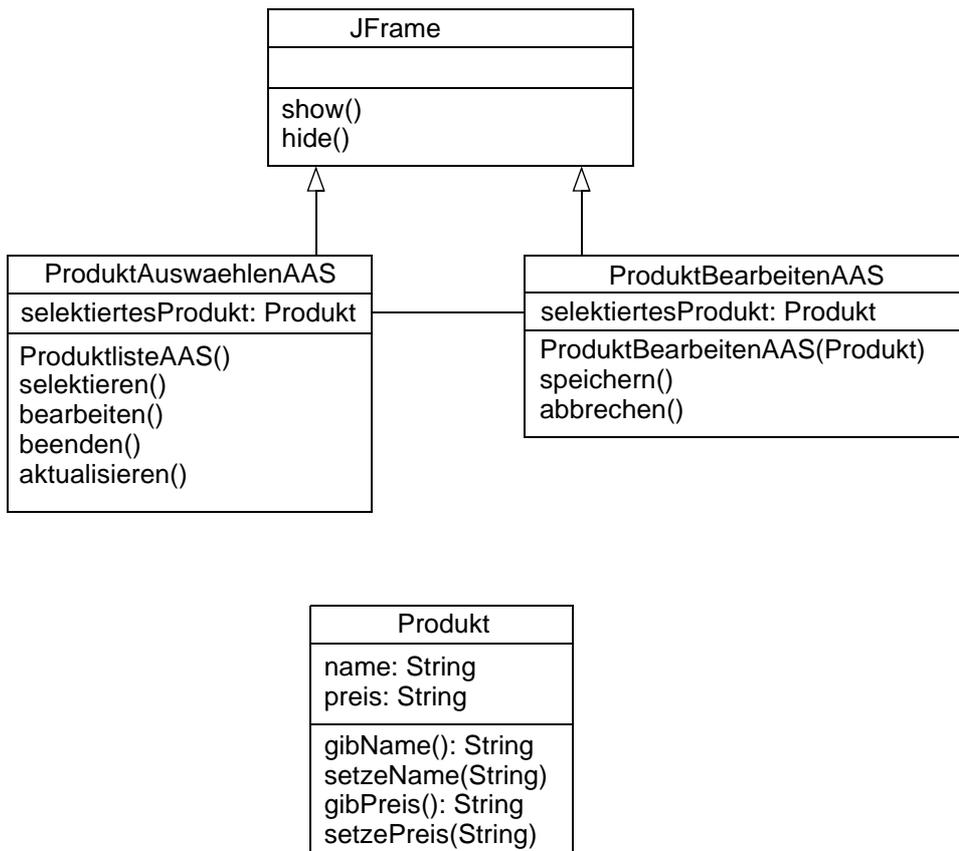
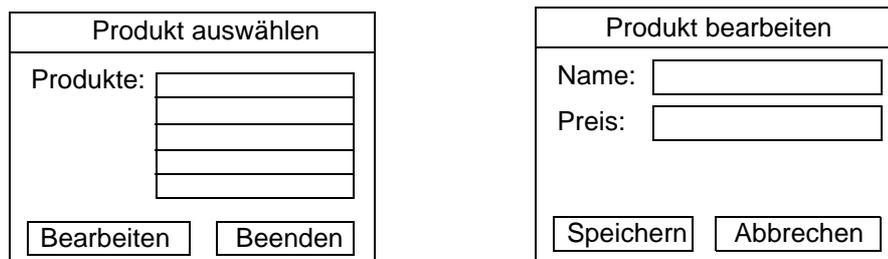


Abb. 3 Die erste Entwurfsvariante

Abb. 4 Die von den Klassen `ProduktAuswaehlenAAS` und `ProduktBearbeitenAAS` dargestellten Fenster

**Kurs 1793 "Software Engineering I - Grundkonzepte der OOSE"**

Klausur am 31.7.2004

Abb. 4 zeigt den Ablauf, wenn der Benutzer ein Produkt auswählt und anschließend seine Attribute bearbeitet. Die Zugriffe auf die Oberflächenelemente (Liste, Eingabefelder) haben wir der Übersichtlichkeit halber weggelassen. Die Operationen show() und hide() dienen dazu, ein Fenster am Bildschirm anzuzeigen bzw. vom Bildschirm zu entfernen.

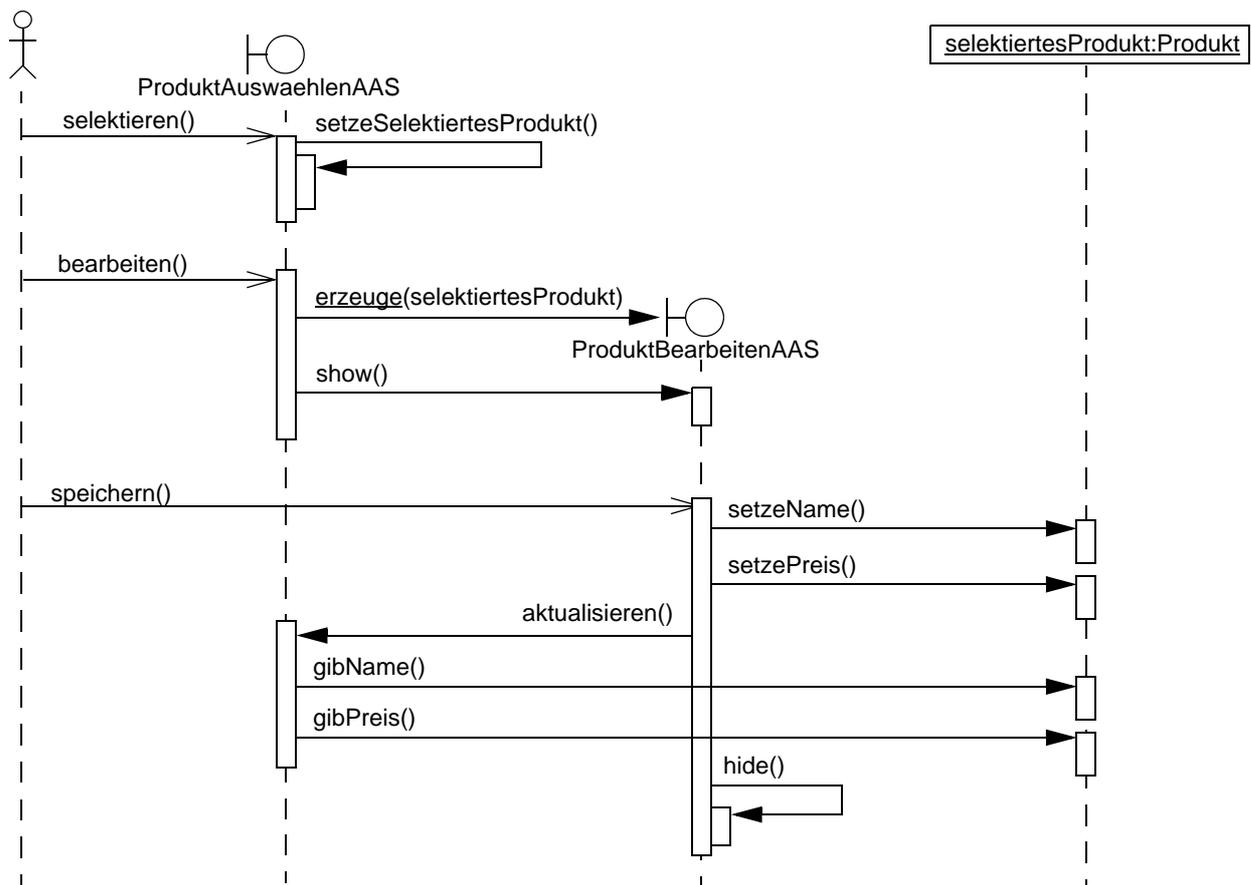


Abb. 5 Der Ablauf als Sequenzdiagramm

- Übertragen Sie das Sequenzdiagramm in Abb. 5 in ein äquivalentes Kollaborationsdiagramm. Verwenden Sie dabei die hierarchische Dezimalnotation.
- Zwischen den beiden AAS-Klassen besteht eine zyklische Abhängigkeit. Nennen Sie mindestens einen Nachteil einer solchen zyklischen Abhängigkeit.
- Die zyklische Abhängigkeit kann durch Einsatz des Beobachtermusters (vgl. Kurstext S. 396/397) beseitigt werden. Die Klasse ProduktAuswaehlenAAS implementiert dabei das Beobachter-Interface, das aus der Operation aktualisieren() besteht. Die Klasse Produkt implementiert das Subjekt-Interface mit den Operationen verbinde(Beobachter), loese(Beobachter) und benachrichtige(). Skizzieren Sie das in dieser Weise modifizierte Klassendiagramm. Sie dürfen dabei die "Stecknadelkopfnotation" verwenden (Kurstext S. 296, Abb. 35.20).

- d) Durch die Änderung in c) ändert sich auch der Ablauf der Operation speichern(). Stellen Sie den neuen Ablauf als Sequenzdiagramm dar.
- e) Welchen Nachteil hat der in Aufgabe c) erstellte Entwurf?