

Kurs 1793 "Software Engineering I"Klausur am 12.8.2006

Sommersemester 2006**Hinweise zur Bearbeitung der Klausur zum Kurs 1793 "Software Engineering I"**

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,
- 3 Aufgaben (Seite 3 - Seite 16)

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"Klausur am 12.8.2006

Aufgabe 1 (18 + 8 + 4 = 30 Punkte)

Für die Autovermietung CarBerlin, die mehrere Niederlassungen in Deutschland hat, soll eine Software zur Durchführung von Online-Reservierungen und zur Verwaltung des Fahrzeugbestands und der Mietvorgänge entwickelt werden. Ihre Aufgabe wird es dabei sein, ein *Domänenklassenmodell* zu erstellen, auf dessen Basis die unten genannten Anforderungen realisiert werden können:

"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.

Der Kunde soll die Möglichkeit haben, online bei einer beliebigen Niederlassung für einen frei wählbaren Zeitraum ein Auto zu reservieren. Die Rückgabe des Fahrzeugs muss bei der gleichen Niederlassung erfolgen wie die Übergabe. Der mit der Reservierung gestartete Mietvorgang kann jederzeit gegen eine Gebühr storniert werden. Reservierungen, zu denen das betreffende Fahrzeug bis 24 Stunden nach dem angesetzten Mietbeginn nicht abgeholt wurde, werden automatisch storniert. Sollte auf Grund eines Unfalls oder einer unvorhergesehenen Reparatur das reservierte Fahrzeug bei der Übergabe nicht zur Verfügung stehen, wird dem Kunden ein gleich- oder höherwertiges (höhere Kategorie) Auto ohne Aufpreis zur Verfügung gestellt.

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. In Sachen Sondertarife ist zu beachten, dass sie häufig nicht für alle Fahrzeugkategorien gelten. So sollten Firmenkundentarife z.B. nicht für Wohnmobile zur Verfügung stehen und die "Cabrio-Wochen" sollten sich auf die entsprechenden Kategorien beschränken. Der Preis für einen Mietvorgang darf ab der Durchführung der Reservierung nicht mehr zum Nachteil des Kunden geändert werden, auch wenn die Gültigkeit des angesetzten Tarifs zwischenzeitlich abläuft.

Die Software soll in der Lage sein, folgende Anfragen zu beantworten:

- Wie hoch ist der Preis für die Reservierung eines VW Golf TDI (Kategorie Mittelklasse) vom 02.08.2006 bis zum 11.08.2006 bei Zugrundelegung des Standardtarifs?
- War das Fahrzeug mit dem Kennzeichen B-LA-926 am 12.04.2006 vermietet? Wenn ja, wie ist der Name des Kunden? Wenn nein, in welcher Niederlassung (Stadt) befand es sich?
- Wieviel Prozent aller Reservierungen wurden zwischen Januar und Mai 2006 storniert?
- Wieviele Autos der Kategorie "Economy" waren im April 2006 vermietet?"

Aufgabenstellung:

- a) Erstellen Sie ein redundanzfreies Klassendiagramm für die beschriebene Anwendungsdomäne. Definieren Sie dabei nicht unnötig viele Klassen und geben Sie zu jeder Beziehung die Multiplizitäten an.
- b) Ergänzen Sie in Ihren Klassen alle Attribute, die zur erfolgreichen Abarbeitung der in der Aufgabenstellung genannten Anfragen benötigt werden.
- c) 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.

Kurs 1793 "Software Engineering I"Klausur am 12.8.2006

Aufgabe 2 (17 + 9 + 8 = 34 Punkte)

Gegeben sind die folgenden Dokumente zur *Anforderungsermittlung* eines Projekts, in dessen Verlauf eine Software zur Verwaltung einer Bibliothek erstellt werden soll. Ihre Aufgabe ist es, diese Dokumente zu vervollständigen und im Anschluss einige Tätigkeiten der *Analyse* durchzuführen.

"Das Bibliothekssystem soll den Studenten die Möglichkeit bieten, den Buchbestand online zu durchsuchen und sich bei Bedarf Buchexemplare reservieren (Student holt Buch selbst ab) oder zuschicken zu lassen. Die Ausleihvorgänge eines Studenten lassen sich über dessen Benutzerkonto nachvollziehen.

Sind alle Exemplare des gewünschten Buchs verliehen, kann sich der Student für das nächste fällig werdende Exemplar vormerken lassen, was ebenfalls in seinem Benutzerkonto vermerkt wird. Der aktuelle Entleiher erhält eine entsprechende Benachrichtigung und kann von nun an die Leihfrist nicht mehr verlängern. Unmittelbar nach der Rückgabe des Buchexemplars wird der vorgemerkte Student davon in Kenntnis gesetzt. Ab diesem Moment hat er für begrenzte Zeit als einziger die Möglichkeit, dieses Exemplar auszuleihen.

Die Leihfrist für ein Buchexemplar kann, sofern keine Vormerkungen vorliegen, bis zu zwei Mal verlängert werden.

Vereinfachende Annahmen für die Aufgabenstellung:

- Alle Nutzer des Systems müssen sich zu Beginn einer Session einloggen, eine Büchersuche als anonymer Nutzer ist nicht vorgesehen.
- Gehen Sie davon aus, dass die Übergabe von Buchexemplaren nur durch das Bibliothekspersonal erfolgen darf und der Student keinen direkten Zugang zu den Bücherregalen hat. Die Büchersuche und -reservierung läuft somit auch vor Ort nach dem oben beschriebenen Schema ab, eine entsprechende Rechnerumgebung steht den Studenten zur Verfügung.

Aufgabenstellung:

- a) Abb. 1 zeigt einen Ausschnitt des Anwendungsfalldiagramms für die Bibliothekssoftware, in dem noch keine Beziehungen zwischen den Anwendungsfällen angegeben sind. Ergänzen Sie die Abb. 1 um sinnvolle Beziehungen, sodass sich die beschriebenen Anforderungen realisieren lassen.

Abb. 2, Abb. 3 und Abb. 4 zeigen die unvollständigen textuellen Spezifikationen der Anwendungsfälle "Buchtitel suchen" "Buchdetails anzeigen" und "Buch reservieren zur Ausleihe". Vervollständigen Sie diese so, dass sie zu Ihrem Anwendungsfalldiagramm passen.

Hinweise:

- Die Einführung zusätzlicher Anwendungsfälle ist nicht erforderlich. Falls Ihre Lösung jedoch weitere Anwendungsfälle benötigt, können Sie diese in der Abb. 1 ergänzen.
- Evtl. hinzukommende extend relationships brauchen nicht textuell spezifiziert zu werden.

Kurs 1793 "Software Engineering I"

Klausur am 12.8.2006

- Evtl. auftretende technische Probleme müssen Sie in der textuellen Spezifikation nicht berücksichtigen, also z.B. keinen exceptional flow "Fehler bei Ermittlung der Buchdetails".
- Lassen Sie sich nicht davon irritieren, wenn Sie in den Abbildungen 2 bis 4 für Ihre Lösungen deutlich weniger Platz benötigen, als wir frei gelassen haben und auch nicht an allen Positionen etwas zu ergänzen haben. Es sind unterschiedliche Lösungsansätze denkbar, was wir durch das Layout des Textes nicht einschränken wollten.

b) Mit der Fertigstellung des Anwendungsfalldiagramms, der textuellen Spezifikation der Anwendungsfälle und dem Domänenklassenmodell aus Abb. 5 schließen wir die Anforderungsermittlung ab und gehen zur *Analyse* über.

Im Folgenden wird der Anwendungsfall "Buch reservieren zur Ausleihe" näher betrachtet. Zeichnen Sie einen Mechanismus zur Realisierung dieses Anwendungsfalls. Wenn Sie Aufgabenteil a) nicht bearbeitet haben, können Sie Annahmen darüber treffen, welche weiteren Anwendungsfälle noch daran beteiligt sind.

c) Das Analysesklassenmodell unterscheidet sich vom Domänenklassenmodell unter anderem dadurch, dass die Klassen Operationen enthalten.

- 1) Wie lassen sich die Nicht-Standardoperationen ermitteln? (ca. 3 Sätze)
- 2) Nennen Sie mindestens zwei Nicht-Standardoperationen, die Sie in den Klassen des in Aufgabenteil b) entwickelten Mechanismus ergänzen würden. Erläutern Sie kurz, wie Sie auf die jeweilige Operation gekommen sind und warum Sie diese der jeweiligen Klasse zugeordnet haben.
- 3) Was liefert einen guten Anhaltspunkt dafür, dass ihr Analysesklassenmodell alle benötigten Operationen enthält? (ca. 2 Sätze)

Kurs 1793 "Software Engineering I"

Klausur am 12.8.2006

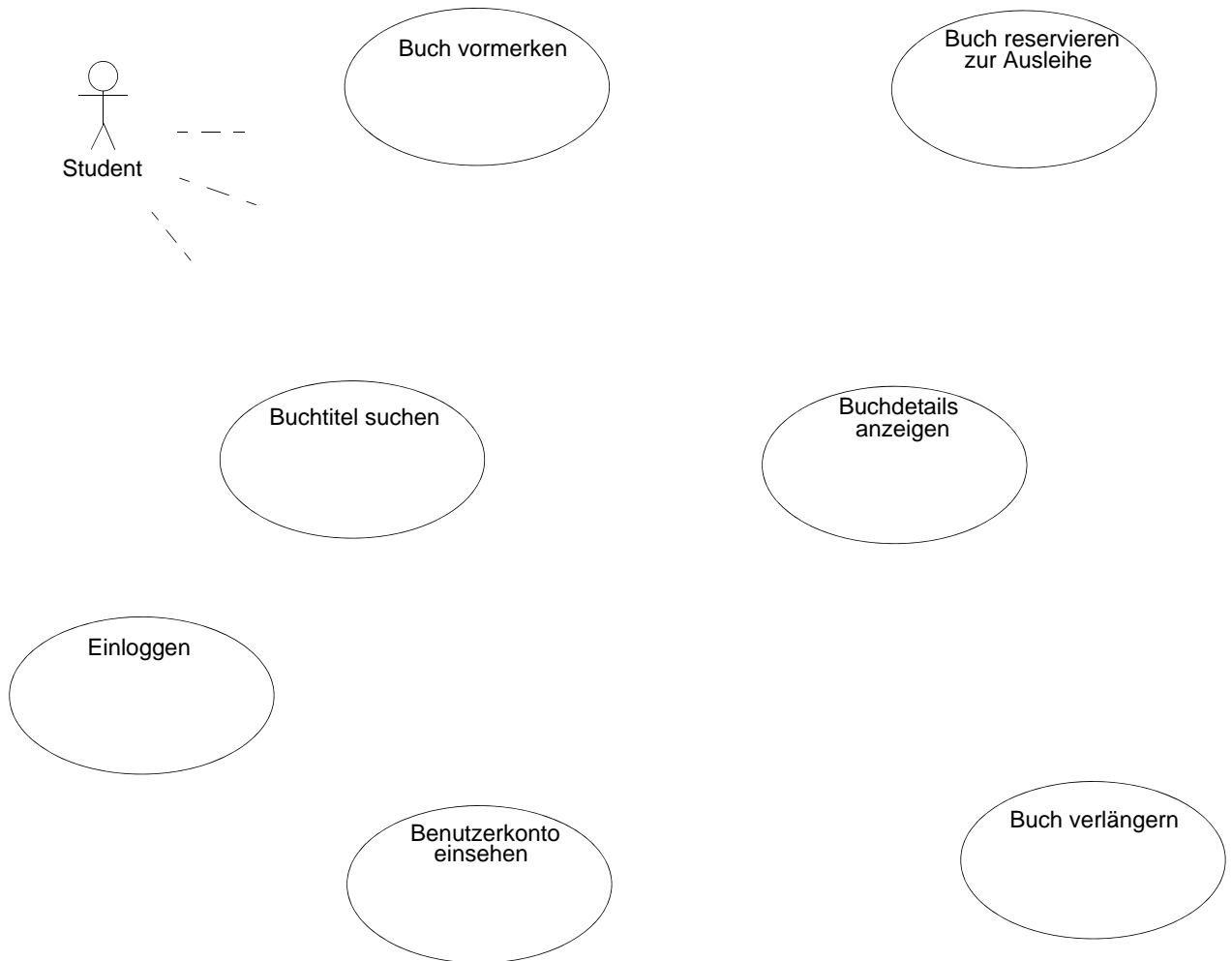


Abb. 1 Zu vervollständigendes Anwendungsfalldiagramm

Kurs 1793 "Software Engineering I"
Klausur am 12.8.2006

use case Buchtitel suchen

actors

Student

precondition

Der Student ist eingeloggt.

main flow

Der Student gibt in der Suchmaske in einem oder mehreren der zur Verfügung stehenden Textfelder (Titel, Autor, ISBN, ...) Begriffe ein und startet den Suchvorgang. Als Ergebnis des Suchvorgangs werden ihm 0..* Buchtitel mit Autor, Erscheinungsjahr und Auflage angezeigt.

end Buchtitel suchen

Abb. 2 Zu vervollständigende textuelle Spezifikationen des Anwendungsfalls "Buchtitel suchen"

Kurs 1793 "Software Engineering I"Klausur am 12.8.2006

use case Buchdetails anzeigen**actors**

Student

precondition

Der Student ist eingeloggt.

main flow

Das System ermittelt alle relevanten Informationen über den gewählten Buchtitel. Neben den bereits angesprochenen Informationen wie Titel, Autor, Auflage und Erscheinungsjahr sind dies beispielsweise die Anzahl und die IDs der Exemplare, deren Standorte und deren Status (verliehen bis, vorgemerkt, ausleihbar, in Reparatur).

end Buchdetails anzeigen

Abb. 3 Zu vervollständigende textuelle Spezifikationen des Anwendungsfalls "Buchdetails anzeigen"

Kurs 1793 “Software Engineering I”
Klausur am 12.8.2006

use case Buch reservieren zur Ausleihe

actors

Student

precondition

Der Student ist eingeloggt.

main flow

Dem Studenten wird eine Seite angezeigt, die neben den bereits zuvor angezeigten Buchdetails auch Informationen wie das fällige Rückgabedatum anzeigt. Außerdem kann der Student hier auswählen, ob er das Buch selbst abholen möchte (Standardfall) oder ob man es ihm per Post zustellen soll. Bei Selbstabholung meldet sich der Student beim Bibliothekspersonal, die weiteren Details der Übergabe sind hier nicht zu betrachten.

alternative flow Postzustellung gewünscht

Der Student hat sich dafür entschieden, dass ihm das Buch per Post zugestellt werden soll. Die weiteren Einzelheiten zu diesem Vorgang sind hier nicht zu betrachten.

end Buch reservieren zur Ausleihe

Abb. 4 Zu vervollständigende textuelle Spezifikationen des Anwendungsfalls "Buch reservieren zur Ausleihe"

Kurs 1793 “Software Engineering I”

Klausur am 12.8.2006

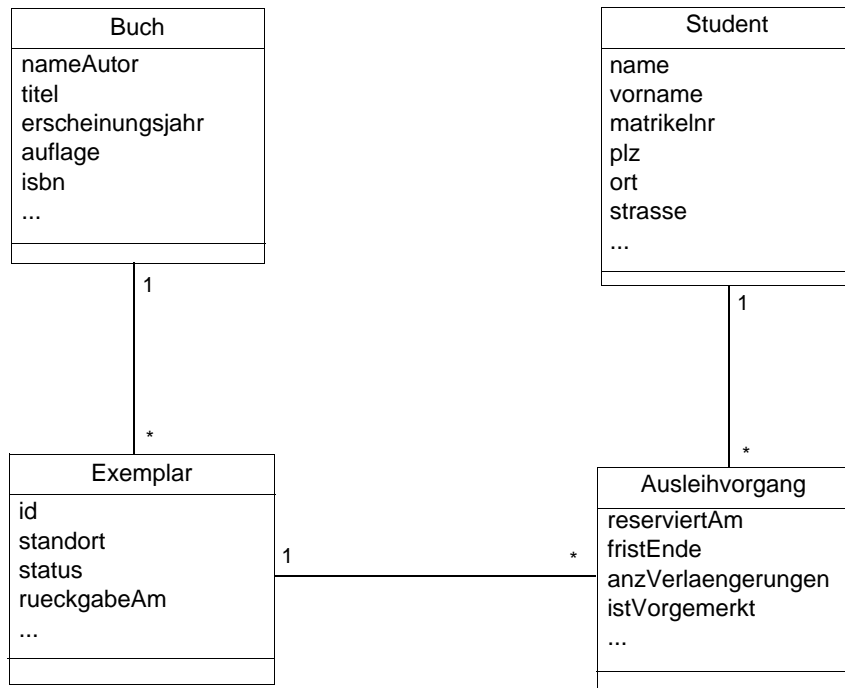


Abb. 5 Ausschnitt aus dem Domänen-Klassendiagramm für das Bibliotheksystem

Aufgabe 3 (16 + 18 + 2 = 36 Punkte)

Für die bereits in Aufgabe 2 behandelte Bibliothek sind nun einige Tätigkeiten des *Feinentwurfs* durchzuführen (beide Aufgaben sind unabhängig voneinander, Informationen aus Aufgabe 2 werden nicht benötigt).

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

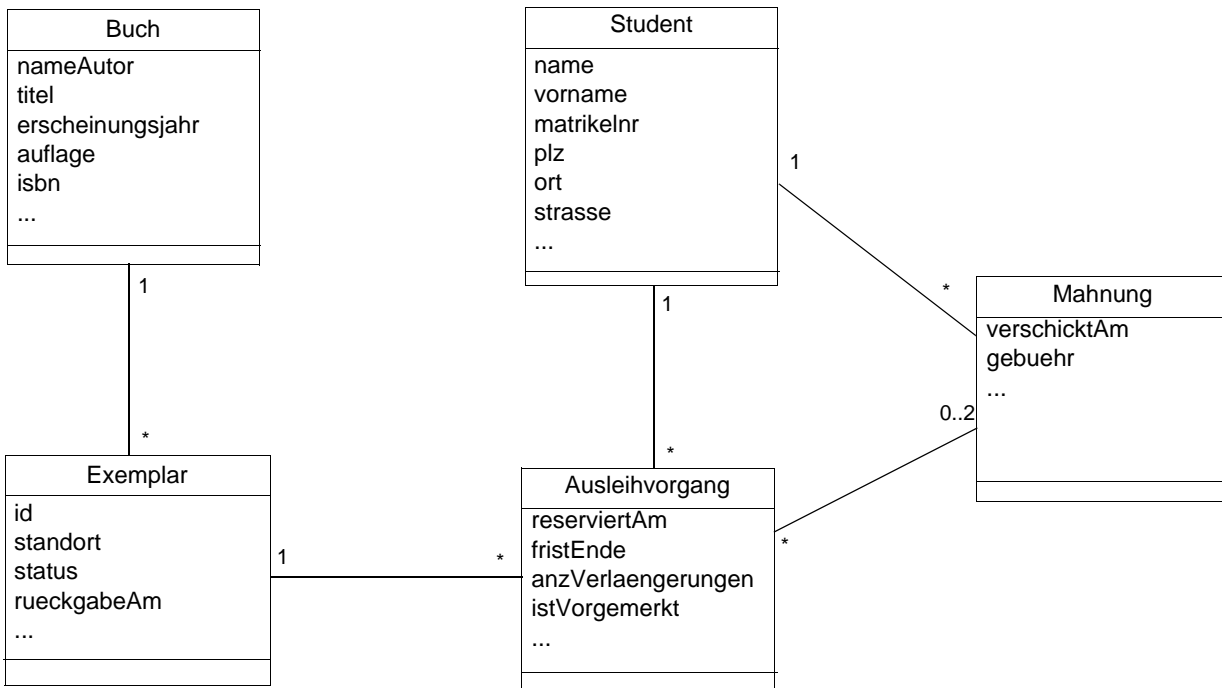


Abb. 6 Zu transformierendes Klassendiagramm, Aufgabe 3

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 Assoziationen und eventueller echter Ganzes-Klassen neu hinzugekommen sind.
- Geben Sie auch die hinzugekommenen Attribute und Standardoperationen (mit Sichtbarkeiten und Parametertypen) an.
- Tragen Sie an den Assoziationsenden die Multiplizitäten ein.
- Von Ordner-, Relations- und Mengenklassen ausgehende Assoziationen müssen nicht weiter transformiert werden.

Kurs 1793 "Software Engineering I"Klausur am 12.8.2006

- b) Erstellen Sie für die Operation `erstelleMahnungen():Iterator` der Klasse `MahnungenVerschickenK` ein Sequenzdiagramm, vervollständigen Sie dazu die Abb. 7.

Innerhalb dieser Operation sind alle Studenten zu ermitteln, die Ihre Leihfrist bei mindestens einem Ausleihvorgang überschritten haben. Für jeden betroffenen Studenten soll genau eine Mahnung erstellt werden, die alle seine Ausleihvorgänge mit Fristüberschreitung umfasst. Sie müssen dabei nicht berücksichtigen, dass lt. Abb. 6 maximal 2 Mahnungen pro Ausleihvorgang verschickt werden dürfen.

Gehen Sie davon aus, dass alle von Ihnen benötigten Operationen zur Verfügung stehen. Gleiches gilt für Klassen, die Sie evtl. zusätzlich zu denen benötigen, die Sie in Ihrer Lösung zu Aufgabenteil a angegeben haben.

Hinweise zu Abb.7:

Die Operation `erstelleMahnungen():Iterator` erzeugt zu Beginn eine leere Liste, in welche dann im weiteren Verlauf die erzeugten Mahnungen einzufügen sind. Der Iterator über diese Liste, der den Rückgabeparameter darstellt, wird ganz zum Schluss über die Operation `iterator()` erzeugt.

Den `StudentIterator` können Sie als gegeben ansehen, die Abläufe zu dessen Ermittlung wurden hier aus Platzgründen vernachlässigt. Gehen Sie davon aus, dass er über alle Studenten iteriert (unabhängig davon, ob sie gemahnt werden sollen oder nicht).

Wir empfehlen, die Lösung zunächst auf Konzeptpapier zu skizzieren und erst dann "sauber" zu zeichnen, wenn der Ablauf und die Abmessungen klar sind. Für Ihre Skizzen können Sie beispielsweise die mit der Abb. 7 identischen Abbildungen 7a und 7b verwenden.

- c) In Abb. 7 wurde nicht dargestellt, wie der `StudentIterator` ermittelt wird. Erläutern Sie kurz (ca. 1-2 Sätze), wie man diesen erhält.

Kurs 1793 "Software Engineering I"

Klausur am 12.8.2006

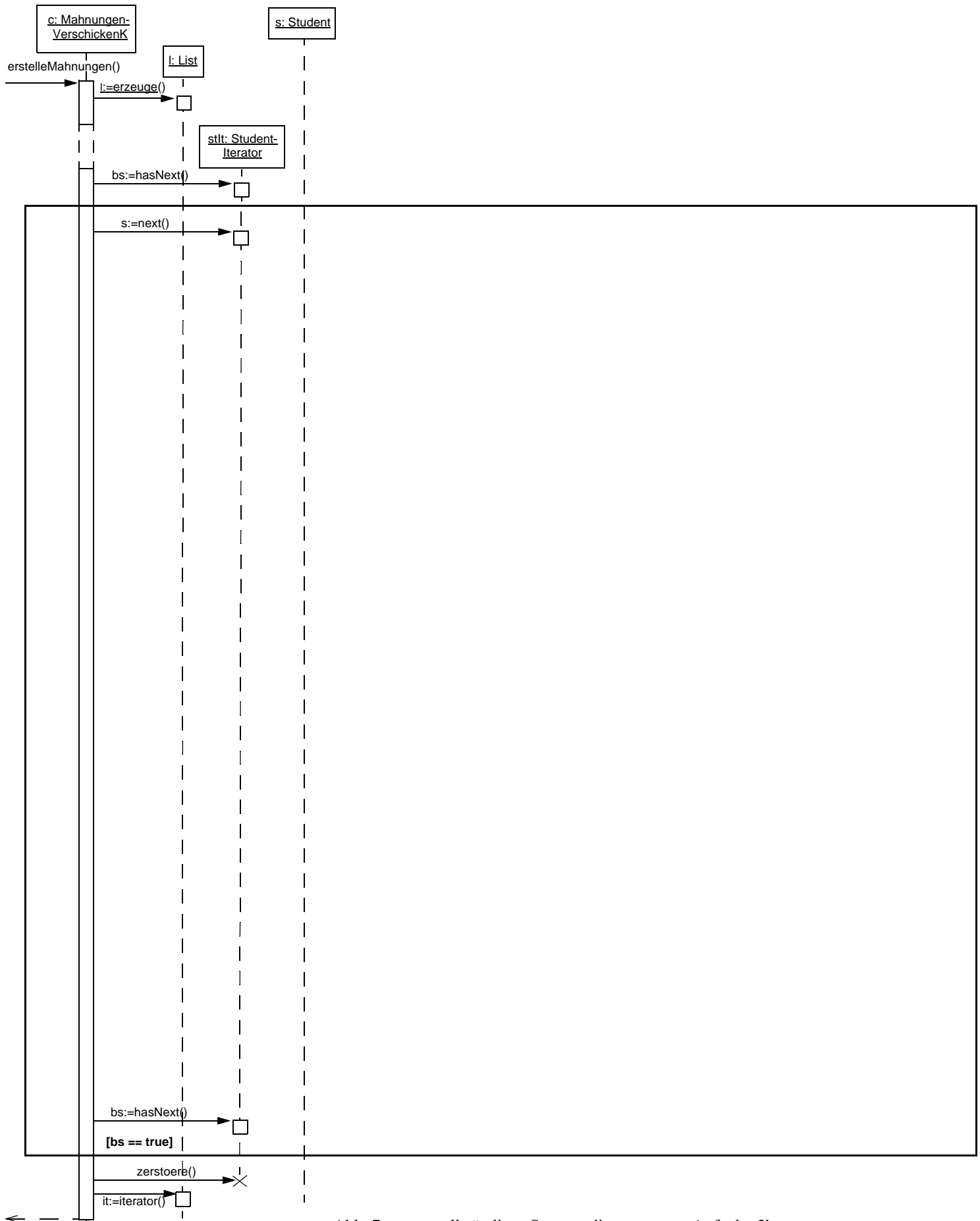


Abb. 7 zu vervollständigtes Sequenzdiagramm zur Aufgabe 3b

Kurs 1793 "Software Engineering I"

Klausur am 12.8.2006

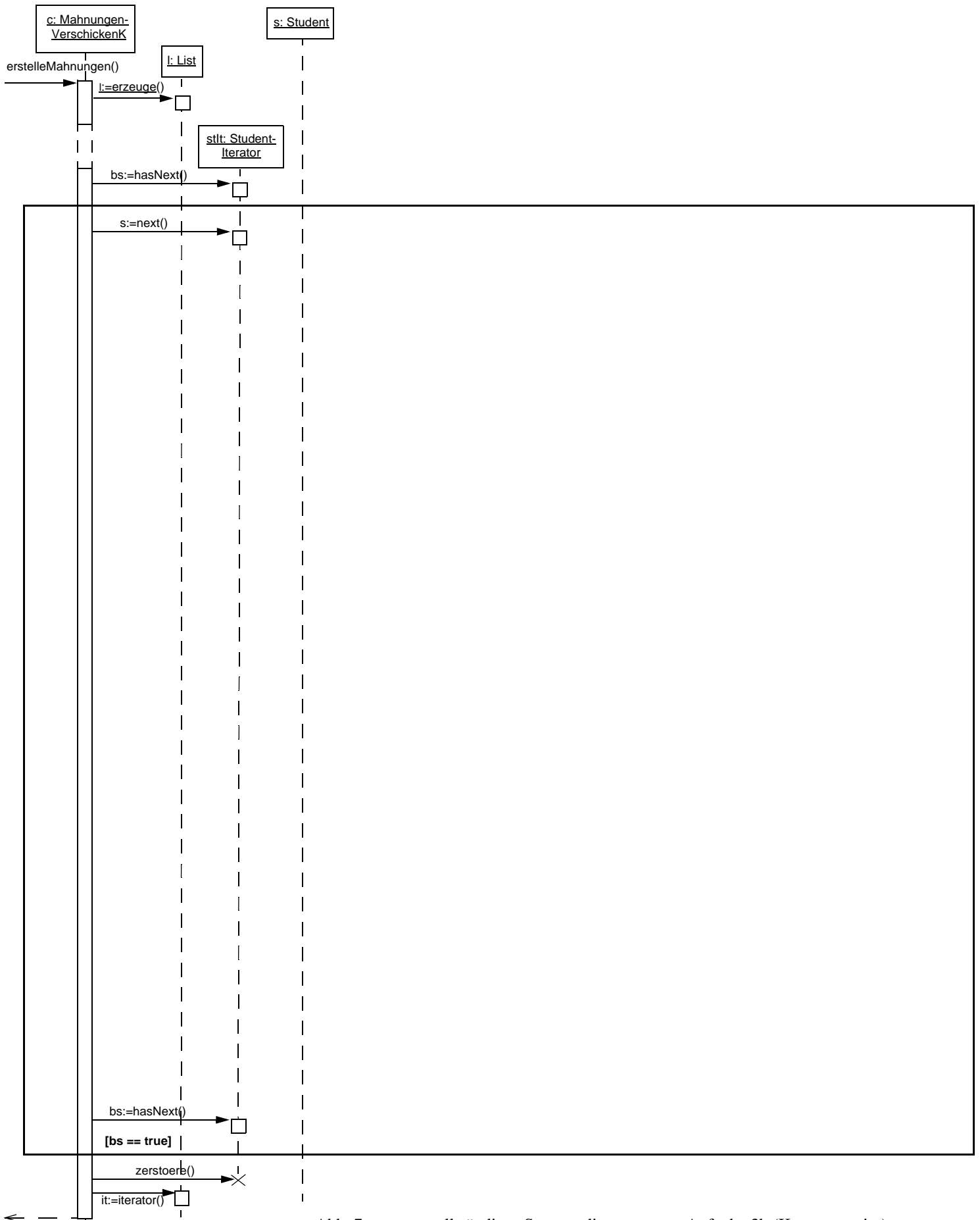


Abb. 7a zu vervollständiges Sequenzdiagramm zur Aufgabe 3b (Konzeptpapier)

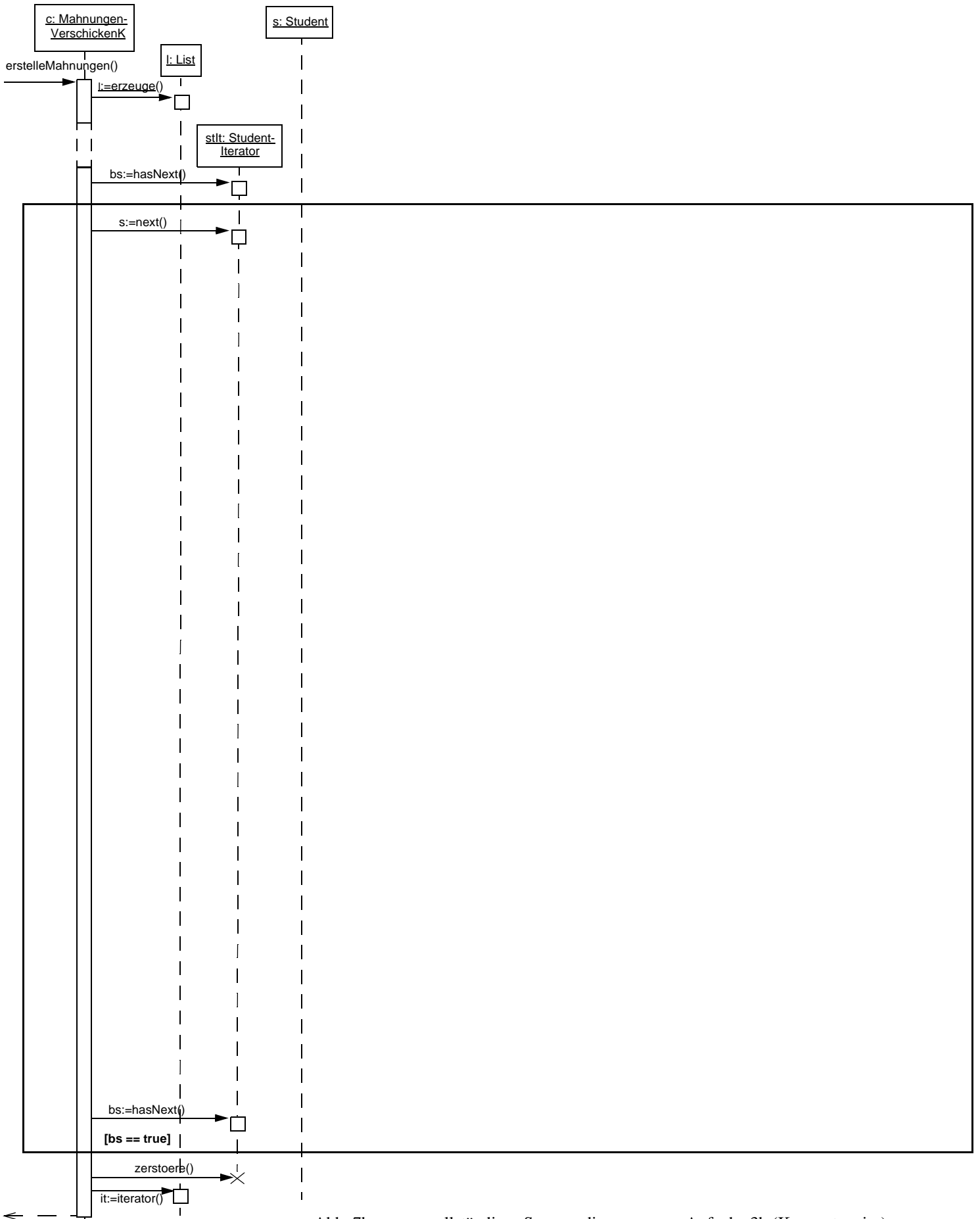


Abb. 7b zu vervollständiges Sequenzdiagramm zur Aufgabe 3b (Konzeptpapier)