Sammlung von Probe-Klausuraufgaben für das Fach Softwaretechnik 1

Stand 22.03.2017

Hinweise

Die nachfolgende Liste enthält eine Liste von Probe-Klausuraufgaben zur Vorbereitung auf die Klausur „Softwaretechnik 1“.

Sinn der Sammlung ist, Ihnen Aufgaben zur Vorbereitung auf die Klausur an die Hand zu geben und Ihnen einen **Eindruck** von den in der Klausur abgefragten Fertigkeiten zu geben.

Bei der Nutzung die Aufgabensammlung bitte ich Folgendes zu beachten:

- Ich stelle keine Lösungen zu den Aufgaben bereit. Da die Beispiele im Wesentlichen der Vorlesung und dem Praktikum entlehnt sind, sollten Sie dort Hinweise zur Lösung finden.

- Die Beispiele in den tatsächlichen Klausuraufgaben werden im Allgemeinen aus anderen thematischen Bereichen kommen, also möglicherweise nicht auf die Softwaresysteme „Restaurantmanagement-System“, „Konzertportal“ oder „Selfservice-Portal für Versicherungen“ Bezug nehmen.

- Die Klausuraufgaben werden vom Typ her den Aufgaben dieser Sammlung ähneln. Allerdings kann (und werde) ich nicht garantieren, dass *alle* Aufgaben der Klausur auch so ähnlich in dieser Sammlung vorkommen. Mit anderen Worten: Diese Aufgabensammlung ist eine **Hilfe**, keine **Referenz** für die Klausur. (Wenn Sie eine Referenz wollen: Am Ende der Vorlesung habe ich Ihnen Hinweise gegeben, welche Inhalte Sie für die Klausur beherrschen sollen.)

- Diese Sammlung enthält **deutlich mehr** Aufgaben als die Klausur. Ich werde also einen Teil der Aufgabentypen für die tatsächliche Klausur auswählen. Zur Orientierung: In der Klausur werden Sie Punktzahlen zu den Aufgaben finden. Es gibt insgesamt 90 Punkte, genauso viele wie Minuten zur Bearbeitung.

Inhalt

Aufgabe 1. Grundlagen der Softwaretechnik 3

Aufgabe 2. Grundlagen von UML-Modellierung 4

Aufgabe 3. UML-Typen 4

Aufgabe 4. Beziehungen zwischen Use Cases 5

Aufgabe 5. Use Case aus einem Anforderungstext ableiten 6

Aufgabe 6. Klassendiagramme - Kardinalitäten und Beziehungstypen 8

Aufgabe 7. Erstellung eines fachlichen Glossars und fachlichen Datenmodells 9

Aufgabe 8. Fünf elementare Entwurfsprinzipien 10

Aufgabe 9. Komponenten und ihre Beziehungen 11

Aufgabe 10. Erstellung einer Kompontenstruktur aus dem Clusterdiagramm 12

Aufgabe 11. Aktivitätsdiagramme 13

Aufgabe 12. Modellierung eines Geschäftsprozesses als Aktivitätsdiagramm 14

Aufgabe 13. UML-Zustandsdiagramme 16

Aufgabe 14. Modellung von Geschäftsobjekt-Zuständen 17

Aufgabe 15. Antipatterns im logischen Datenmodell 18

Aufgabe 16. Transformation des fachlichen ins logische Datenmodell 19

Aufgabe 17. Bounded Contexts 22

Aufgabe 18. Komponenten-Schnittstellen 24

Aufgabe 19. Datenaustausch zwischen Komponenten als Sequenzdiagramm 25

Aufgabe 1. Grundlagen der Softwaretechnik

Bewerten Sie die folgenden Aussagen mit „richtig“ (r) oder „falsch“ (f).

- Wenn Sie die Aussage für **richtig** halten, begründen Sie Ihre Einschätzung bitte jeweils in einem Satz.
- Wenn Sie die Aussage für **falsch** halten, beschreiben Sie die richtige Lösung bitte in einem Satz.

Aussage	**r / f**	**Begründung**
1. Ein Fehler in der Anforderungsdefinition ist 1/10 so teuer zu beheben wie ein Fehler in der Architektur (Verzehnfachungsregel)		
2. Der Hauptteil der Architekturarbeit fällt in der Entstehungs- und Entwicklungsphase eines Softwaresystems an. Während der nachfolgenden Evolutionsphase wird die Architektur meist nicht mehr stark verändert.		
3. Ein wesentliches Ziel von Architektur ist die Erhöhung der Komplexität der Aufgabe.		
4. Auch organisatorische Bedingungen (Stakeholder und politische Prozesse, regionale Eigenheiten, Kompetenzen, Skills, …) können ein Komplexitätstreiber für Software sein.		
5. Wenn man SW-Projekte ohne die Ansätze aus der Softwaretechnik durchführt, geht es schneller.		

Aufgabe 2. Grundlagen von UML-Modellierung

Bewerten Sie die folgenden Aussagen mit „richtig“ (r) oder „falsch“ (f).

- Wenn Sie die Aussage für **richtig** halten, begründen Sie Ihre Einschätzung bitte jeweils in einem Satz.
- Wenn Sie die Aussage für **falsch** halten, skizzieren Sie die richtige Lösung bitte in einem Satz.

Aussage	**r / f**	**Begründung**
1. Wenn ein UML-Modell fachgerecht spezifiziert ist, dann ist es sowohl intuitiv verständlich wie auch formal soweit korrekt, dass man es z.B. zur Generierung von Code verwenden kann.		
2. Das Gegenstück zur Entwicklungsperspektive ist die Verhaltensperspektive.		
3. Use-Case-Diagramme gehören sowohl zur Verhaltungs- wie zur Entwicklungsperspektive.		
4. Eine Architekturspezifikation muss zwingend alle UML-Diagrammtypen enthalten.		

Aufgabe 3. UML-Typen

In der Vorlesung haben wir uns auf sechs der 14 UML-Diagrammtypen konzentriert, weil sie besonders praxisrelevant sind.

Nennen Sie diese sechs Diagrammtypen, und beschreiben Sie jeweils in 1-2 Sätzen, was man mit dem jeweiligen UML-Diagrammtyp modelliert.

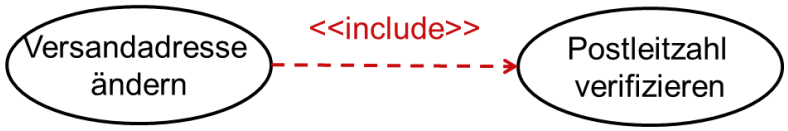
Aufgabe 4. Beziehungen zwischen Use Cases

In einem Webshop kann der Nutzer eine Versandadresse als Default hinterlegen und diese natürlich auch ändern. Welches Use-Case-Diagramm passt zu diesem Sachverhalt?

Kreuzen Sie die Ihrer Meinung nach richtige Lösung an und begründen Sie kurz in einem Satz.

a.

b.



c.

Aufgabe 5. Use Case aus einem Anforderungstext ableiten

Für ein neues Softwaresystem liegt folgender Anforderungstext vor (Auszug):

Ein Versicherungskonzern möchte ein neues Self-Service-Portal erstellen. In diesem Portal können seine Kunden Schäden melden. Der Kunde gibt Datum und Beschreibung des Schadens ein. Das Portal leitet den Schaden weiter an die Backend-Systeme, wo die eigentliche Geschäftslogik angesiedelt ist. Nach deren Bestätigung schickt das Portal einen Eingangsbericht an den Kunden. Zusätzlich deckt das Portal die Änderung der Kundenadresse ab. Die Änderung wird vom Portal in die unternehmensweite Kundendatenbank eingetragen.

a) Identifizieren Sie einen oder mehrere Use Cases, die sich aus dem untenstehenden Anforderungstext ergeben.

- Gehen Sie dabei nach der Methode der Vorlesung vor (Markieren von Textstellen)
- Zeichnen Sie ein entsprechendes Use-Case-Diagramm.

b) Füllen Sie das Use-Case-Template für den Use Case aus, der das Melden von Schäden zum Inhalt hat.

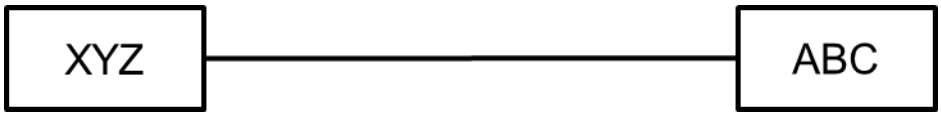
- Gehen Sie dabei nach der Methode der Vorlesung vor (Markieren von Textstellen, syntaktische / semantische Bereinigung)
- Nutzen Sie bitte das untenstehende Template. Wenn Sie mehr Platz benötigen, erweitern Sie das Template bitte auf der nachfolgenden Seite.

Name			
auslösender Aktor			
Auslöser			
Vorbedingung			
Nachbedingung			
Haupt-Szenario	*Benutzer*	*System*	*Externes System*
Alternativ-Szenario			
Ausnahme-Szenario			

Aufgabe 6. Klassendiagramme - Kardinalitäten und Beziehungstypen

a) Welche Kardinalität hat die Assoziation zwischen den Klassen XYZ (zu der die Objekte x, y, und z gehören) und ABC (mit den Objekten a, b und c) ?

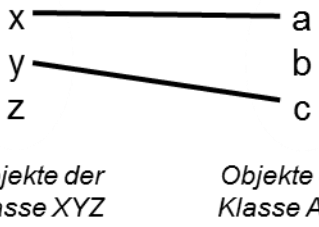
Zeichnen Sie die Kardinalitäten in korrekter UML-Notation ein:



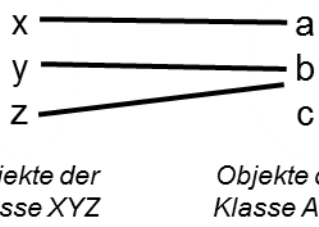
Anmerkungen:

- Die Objekte der Klassen können in der untenstehenden Weise verbunden sein.
- Nennen Sie bitte die **am wenigsten allgemeine** Kardinalität!
  (Damit ist gemeint: „* - *“ passt ja auf jede Art der Beziehung. Falls die untenstehenden Beziehungen aber eine eingeschränktere Kardinalität nahelegen, dann wählen Sie bitte diese.)

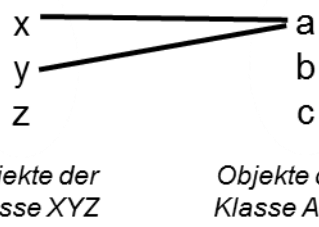
Möglichkeit 1



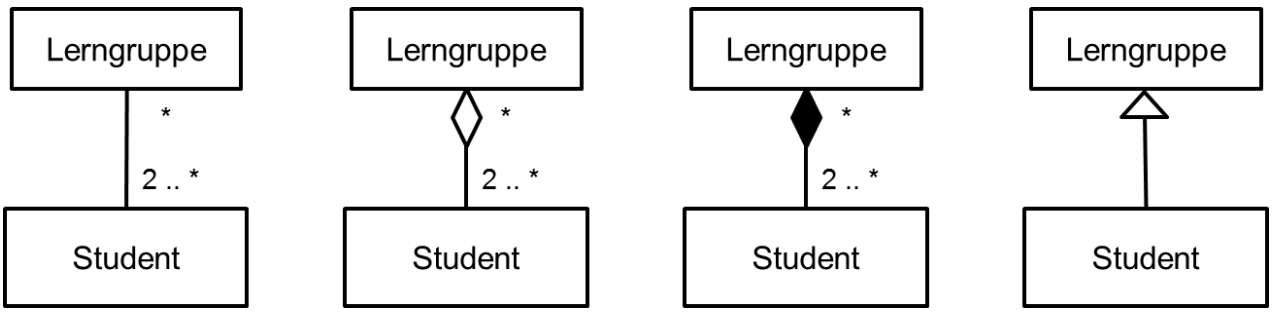
Möglichkeit 2



Möglichkeit 3



b) Welcher Beziehungstyp ist am sinnvollsten? Markieren Sie diesen und begründen Ihre Wahl in 1-2 Sätzen.



Aufgabe 7. Erstellung eines fachlichen Glossars und fachlichen Datenmodells

Ein neues Softwaresystem soll (unter anderem) folgende Anforderungen abdecken:

Durch Anklicken eines Konzerts kann ein Nutzer Konzerttickets kaufen. Nach Eingabe seiner Kreditkartendaten (Name, Kredit-kartennummer und Gültigkeit) schickt das System die Zahlungs-daten an die externen Kreditinstitute weiter. Diese veranlassen die Auszahlung. Das System druckt dann die Tickets, kuvertiert sie, und verschickt sie an die vom Nutzer angegebene Adresse.

a) Stellen Sie nach der Methode der Vorlesung (Anstreichen im Text, Bereinigung) das fachliche Glossar zusammen. Nutzen Sie das untenstehende Template.

Anmerkung: Auf die Beschreibung der Geschäftsobjekte können Sie hier aus Zeitgründen verzichten.

Geschäftsobjekt	Attribut	Beschreibung
		Beschreibung kann aus Zeitgründen in der Klausur weggelassen werden

b) Erstellen Sie aus den Elementen des fachlichen Glossars das **fachliche Datenmodell** als einfaches UML-Klassendiagramm.

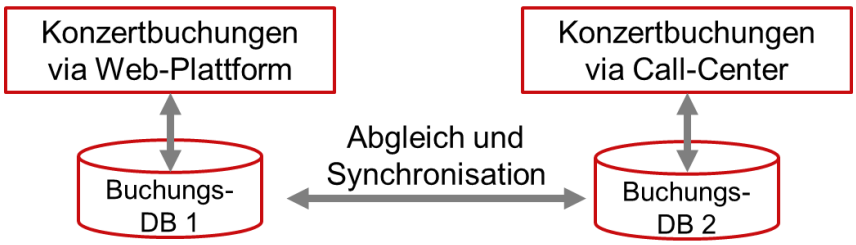
Aufgabe 8. Fünf elementare Entwurfsprinzipien

a) In der Vorlesung haben wir uns mit fünf elementaren Prinzipien beschäftigt, die sich in den verschiedenen Phasen des Entwurfsprozesses immer wieder anwenden lassen.

Nennen Sie diese fünf Prinzipien und beschreiben Sie jeweils kurz in 1-2 Sätzen, was diese aussagen.

b) Bei einem Konzertbuchungsportal soll der Nutzer sowohl über ein Call-Center wie auch über eine Web-Plattform Konzertbuchungen vornehmen können. Call-Center-Buchungen sollen anschließend auch über die Web-Plattform weiterbearbeitet werden können, und umgekehrt.

Die Softwarekomponenten werden wie folgt modelliert:



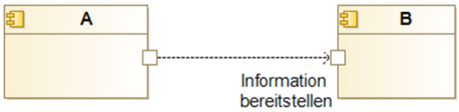
Welches elementare Entwurfsprinzip wird bei diesem Komponentendesign **verletzt**? Bitte begründen Sie kurz in 1-2 Sätzen.

Aufgabe 9. Komponenten und ihre Beziehungen

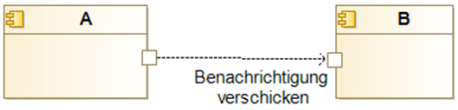
a) Bitte begründen Sie kurz (2-3 Sätze): Warum sollte bei Komponentendiagrammen (in dem Stil wie in der Vorlesung vorgestellt, mit der Nutzung von Ports zur Charakterisierung von Schnittstellen) **auch die Aufrufstelle ein Port sein**?

b) Welche Aufrufart (synchron, asynchron, Pub-Sub/Event) bieten sich in den untenstehenden Fällen (1) und (2) an? Bitte begründen Sie jeweils kurz in 1-2 Sätzen.

1 A fragt eine Information von B ab.
B antwortet zuverlässig und schnell.



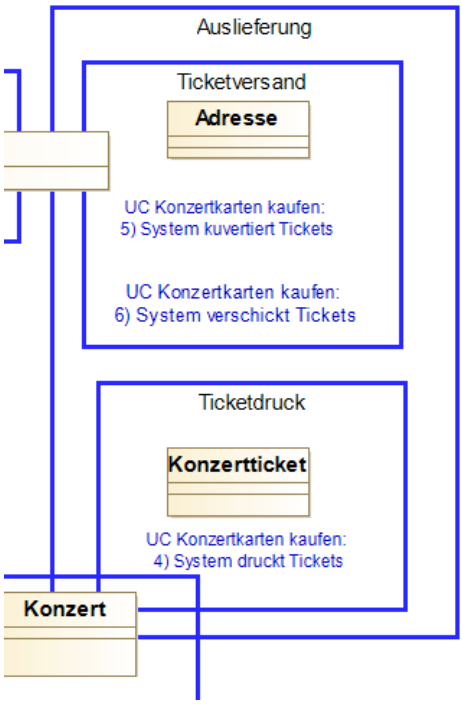
2 A teilt das Ergebnis einer Berechnung mit. Viele Komponenten bekommen diese Benachrichtigung, darunter B.



Aufgabe 10. Erstellung einer Kompontenstruktur aus dem Clusterdiagramm

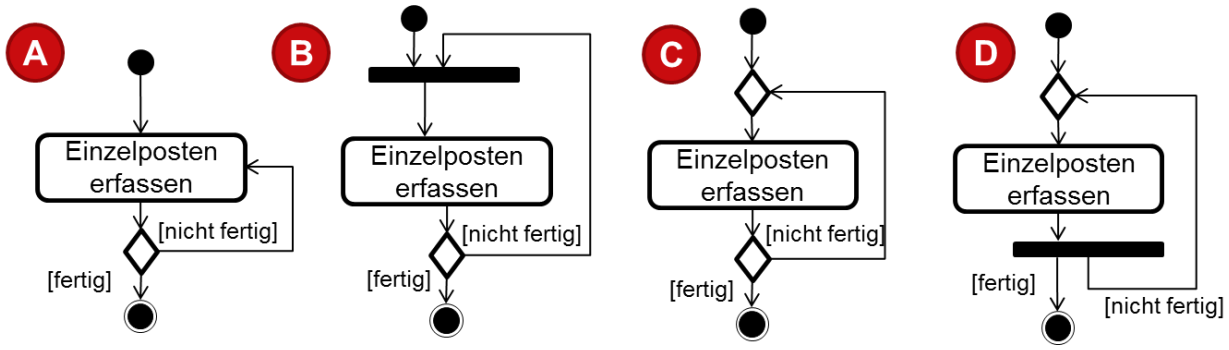
Beim Entwurf eines Softwaresystems zum Bestellen von Konzertkarten hat das Clustering von Use-Case-Szenarien und fachlichem Datenmodell zu folgendem Diagramm geführt (Auszug). Transformieren Sie dieses Diagramm in ein **Komponentendiagramm** gemäß Vorgehen der Vorlesung.

- Sie können dabei noch auf Interfaces verzichten; Ports genügen.
- Wenn Sie bei der Komponentenmodellierung von den Clustergrenzen abweichen, so kann das (als Ihre Modellierungsentscheidung) valide Gründe haben. In diesem Fall sollten Sie das kurz in 1-2 Sätzen begründen.



Aufgabe 11. Aktivitätsdiagramme

a) Bei welchen der Modellierungen A – D wird **mindestens ein Einzelposten** erfasst? Bitte begründen Sie Ihre Antwort kurz.



b) Sie modellieren Geschäftsprozesse als **Aktivitätsdiagramme mit Swimlanes**: Welche der nachfolgenden Aussagen stimmen? Bewerten Sie diese bitte mit „richtig“ (r) oder „falsch“ (f).

- Wenn Sie die Aussage für **richtig** halten, begründen Sie Ihre Einschätzung bitte jeweils in einem Satz.
- Wenn Sie die Aussage für **falsch** halten, beschreiben Sie die richtige Lösung bitte in einem Satz.

Aussage	**r / f**	**Begründung**
1. Wenn die Swimlanes für Aktoren (Nutzer des Systems) stehen, dann ist der Fokus der Darstellung auf der Rolle des jeweiligen Nutzers im Gesamtablauf.		
2. Swimlanes stehen immer für eigene Komponenten des IT-Systems		
3. Swimlanes helfen beim Erkennen, welche Komponenten miteinander interagieren, ähnlich wie beim Sequenzdiagramm		
4. Interaktionen von Nutzern mit dem System finden meist in den Swimlanes für UI-Komponenten statt		

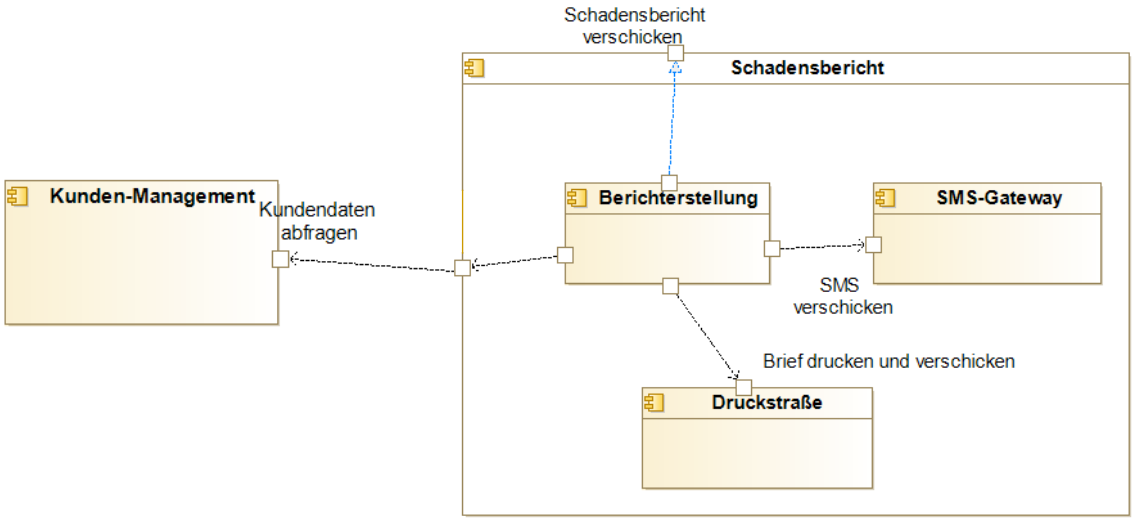
Aufgabe 12. Modellierung eines Geschäftsprozesses als Aktivitätsdiagramm

Sie planen ein Self-Service-Portal für eine Versicherung, in dem die Kunden Schäden melden können. Ein Feature dieses Portal ist es, dass der Kunde als Eingangsbestätigung seinen Schadensbericht „zurückgespiegelt“ bekommt, und zwar je nach seiner Präferenz in Papierform und/oder als kurze, zusammenfassende SMS. Unten sehen Sie den entsprechenden Ausschnitt aus dem Use Case (nur das Hauptszenario dargestellt) und Ihre bis dahin modellierte Komponentenstruktur.

Modellieren Sie den im Use-Case-Ausschnitt dargestellten Teilprozess als **UML-Aktivitätsdiagramm**. Hinweise:

- Nutzen Sie dabei **Partitionen**, die Sie bitte wie in der Vorlesung vorgestellt selbst festlegen.
- Die Komponente „Kunden-Management“ verwaltet u.a. folgende Daten:
  - präferierter Kommunikationskanal des Kunden (Brief / SMS)
  - Adresse des Kunden
  - Mobilfunknummer des Kunden (für SMS)
- Der Startpunkt Ihres Teilprozesses ist der Aufruf der Schnittstelle „Schadensbericht verschicken“ (siehe Port oben in Komponente „Schadensbericht“). Den Aufruf selbst brauchen Sie **nicht** zu modellieren.
  - Ihr Startknoten ist also mit dem Event „Schadensbericht angefordert“ beschriftet.
  - Die Schadensdaten werden beim Aufruf übergeben. Diese müssen Sie also nicht woanders abfragen.

	Benutzer	***System***	***Externes System***
Haupt-Szenario		1) Das System prüft den/die präferierten Kommunikationskanal/-kanäle des Kunden.	
		2) Wenn ein Kanal „Brief“ ist, dann formatiert das System die Schadensdaten als Brief und erzeugt ein PDF.	
		3) Das System ruft die Adressdaten des Kunden aus der Datenbank den präferierten Kommunikationskanal des Kunden ab.	
		4) Das System übergibt das Brief-PDF und die Adresse an die Druckstraße zum Druck und Versand.	
		5) Wenn ein Kanal „SMS“ ist, dann ruft das System die Telefondaten des Kunden aus der Datenbank ab.	
		6) Es übergibt die Telefondaten mitsamt einer Kurzfassung der Schadensmeldung an das SMS-Gateway zum Versand.	

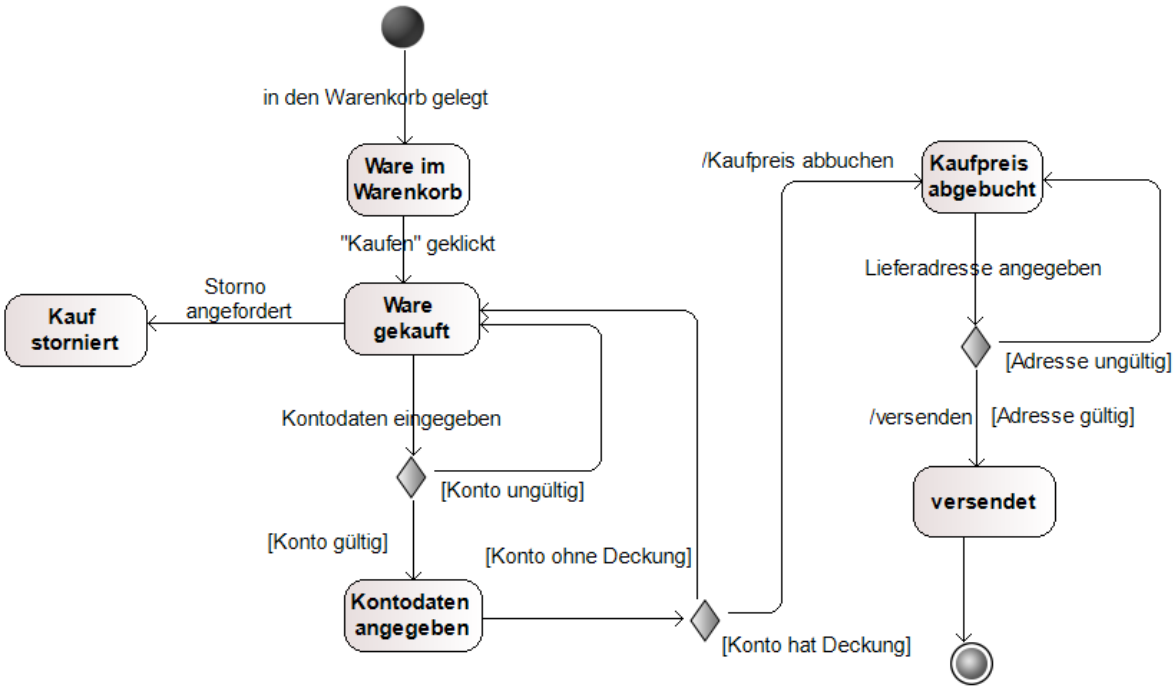

Schadensbericht verschicken
Schadensbericht
Kunden-Management
Kundendaten abfragen
Berichterstellung
SMS-Gateway
SMS verschicken
Brief drucken und verschicken
Druckstraße

Aufgabe 13. UML-Zustandsdiagramme

a) Betrachten Sie folgendes UML-Zustandsdiagramm. Nehmen Sie an, Sie wären schon dabei, Ihre Kontodaten einzugeben, entscheiden sich aber dann noch gegen einen Kauf. Wenn der Webshop genauso implementiert ist wie in diesem Diagramm, **können Sie dann noch stornieren? Wenn ja, wie?**

- Wenn ja: wie?
- Wenn nein: warum nicht?

Bitte begründen Sie Ihre Antwort kurz in 1-2 Sätzen.



b) Sowohl Geschäftsobjekt-Zustände (als UML-Zustandsdiagramm) wie auch Geschäftsprozesse (also UML-Aktivitätsdiagramm) beschreiben Verhaltensaspekte eines IT-Systems.

Bitte erklären Sie kurz, wie diese beiden Sichten zusammenhängen. Wie kann man die beiden ineinander überführen?

Aufgabe 14. Modellierungung von Geschäftsobjekt-Zuständen

Bitte modellieren Sie für das Geschäftsobjekt „Konzert-Sitzplatz“ (für das Konzertkartenportal) die nachfolgend dargestellte Zustandsfolge in Form eines UML-Zustandsdiagramms:

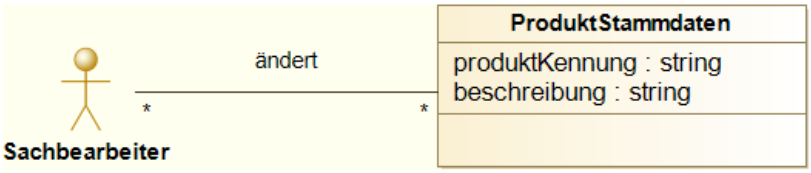
- Ein Konzert-Sitzplatz ist zunächst frei
- Im Konzert-Webportal kann der Kunde den Platz reservieren
- Er kann den Platz dann entweder
  - wieder freigeben oder
  - in den Warenkorb legen
- Auch aus dem Warenkorb kann der Kunde den Platz freigeben
- Andernfalls kauft er den Konzertplatz aus dem Warenkorb
- Einen gekauften Konzert-Sitzplatz kann der Nutzer dann stornieren, wenn es eine Warteliste gibt
  - Der Platz wird dann dem Ersten der Warteliste als "reserviert" gemeldet
  - Andernfalls muss der Kunde die Karte behalten
- Wenn keine Storno vorliegt, gilt der Sitzplatz ab Tag des Konzerts als eingenommen (egal ob Kunde hingeht oder nicht)

Aufgabe 15. Antipatterns im logischen Datenmodell

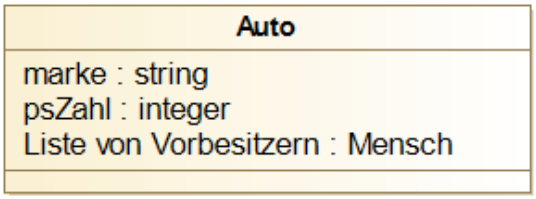
Die unten dargestellten Ausschnitte aus einem logischen Datenmodell verstoßen jeweils in einem wesentlichen Aspekt gegen die Regeln für ein logisches Datenmodell.

Nennen Sie diesen Aspekt, begründen Sie kurz, warum man ein logisches DM nicht so modellieren sollte, und zeichnen Sie jeweils eine „verbesserte“ Version als UML-Klassendiagramm.

a)



b)



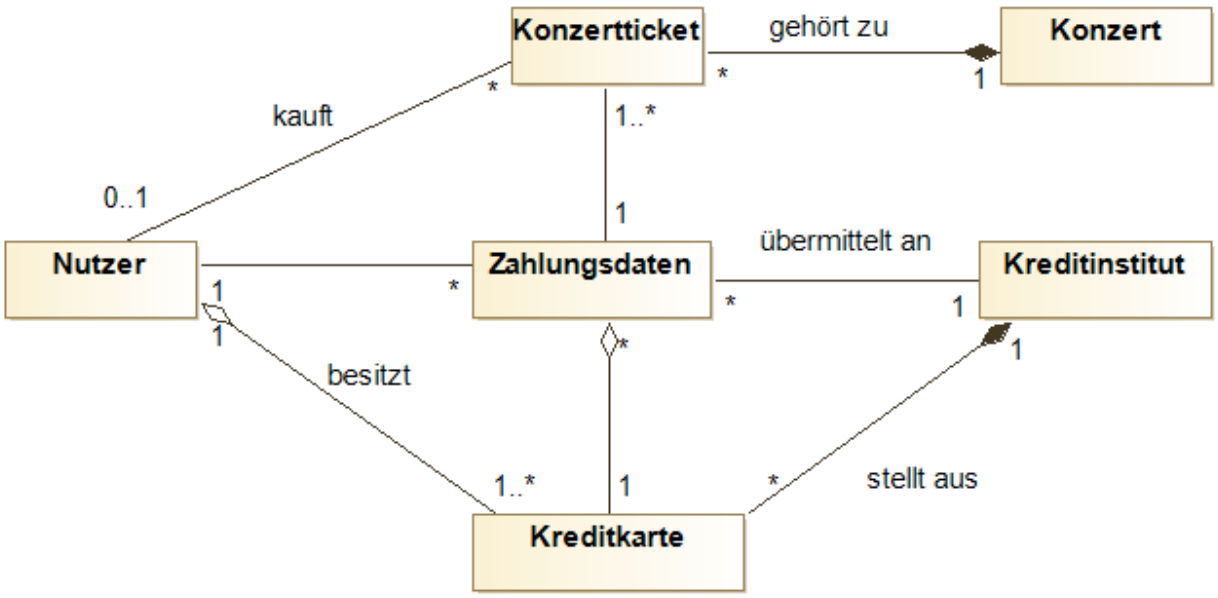
Aufgabe 16. Transformation des fachlichen ins logische Datenmodell

Hier sehen Sie einen Ausschnitt aus dem fachlichen Datenmodell zu einem Portal zum Erwerb von Konzertkarten. Der Einfachheit halber brauchen Sie sich in dieser Aufgabe **nicht um die Attribute** der Geschäftsobjekte zu kümmern.

Beachten Sie für die Transformation noch folgende Hinweise:

- Es soll möglich sein, ein Konzert von einem Veranstaltungsort zu einem anderen verlegen, ohne dass die Konzerttickets neu gedruckt und verschickt werden müssen.
- Auf absehbare Zeit wird es nur ein einziges angeschlossenes Kreditinstitut geben, das alle Kreditkarten-Inkassovorgänge für das Portal abwickelt.
- Das Portal soll auf große Nutzerzahlen ausgelegt werden. Für Analysen soll die Query, welche Konzerte ein Nutzer besucht hat, so performant wie möglich ausführbar sein.

Führen Sie mit diesem Teilmodell einige Schritte der **Transformation vom fachlichen zum logischen DM** durch (Details siehe Teilaufgaben a) bis c))



a) Finden Sie für die in der untenstehenden Tabelle aufgeführten Transformationsschritte vom fachlichen zum logischen DM **jeweils ein Beispiel** (mit kurzer Begründung).
Zeichen Sie noch kein neues Klassendiagramm!

Transformationsschritt	**Modelländerung**	**Begründung**
fehlende Klassen oder Beziehungen ergänzen		
für die Implementierung irrelevante Klassen / Beziehungen entfernen		
Optimierung des logischen DM für die Implementierung, insb. gemäß nichtfunktionaler Anforderungen		

b) Identifizieren Sie **Stamm-, Änderungs-, Bestands- und Bewegungsdaten**. Tragen Sie diese in der untenstehenden Tabelle ein, beschreiben Sie die nötigen Modelländerungen und begründen Sie diese kurz.
Zeichen Sie noch kein neues Klassendiagramm!

Datenart	**Klasse / Beziehung im fachlichen DM**	**nötige Änderung in der Transformation zum logischen DM**
Stammdaten		
Änderungsdaten		
Bestandsdaten		
Bewegungsdaten		

c) Wenden Sie die in a) und b) festgestellten notwendigen Änderungen an und zeichnen Sie das logische DM als UML-Klassendiagramm (ausgehend von dem fachlichen DM am Anfang der Aufgabe).

Aufgabe 17. Bounded Contexts

a) In einem logischen Datenmodell legen Sie zum Abschluss der Modellierung bezüglich Ihrer Subsysteme deren **Bounded Contexts** fest. Welche der nachfolgenden Aussagen zu Bounded Contexts stimmen? Bewerten Sie diese bitte mit „richtig“ (r) oder „falsch“ (f).

- Wenn Sie die Aussage für **richtig** halten, begründen Sie Ihre Einschätzung bitte jeweils in einem Satz.
- Wenn Sie die Aussage für **falsch** halten, skizzieren Sie die richtige Lösung bitte in einem Satz.

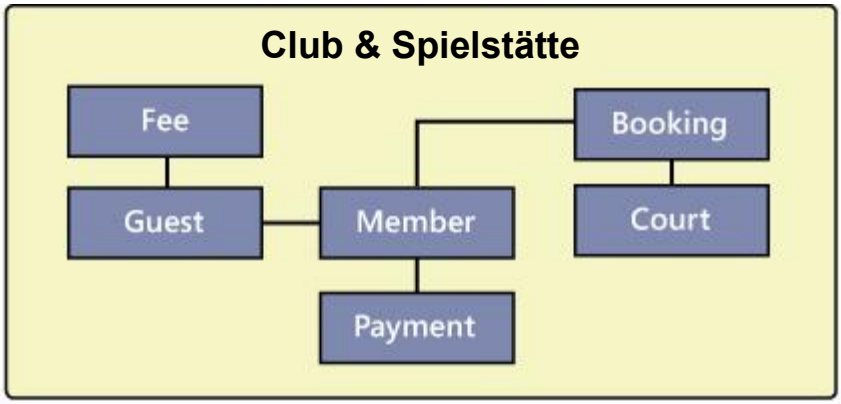
Aussage	**r / f**	**Begründung**
1) Ein Bounded Context ist die Kontextgrenze einer Anwendungsdomäne, innerhalb der sie eigenständig modelliert werden kann		
2) Bounded Contexts dürfen sich nicht überlappen		
3) Der Abgrenzungstyp "Separate Ways" bedeutet, dass keinerlei Abstimmung des Modells zwischen zwei Bounded Contexts nötig ist		
4) Bei externen Systemen, die nicht beeinflusst werden können, ist "Customer/Supplier" meist der geeignetste Abgrenzungstyp		

b) Sie sehen hier das (in informeller Notation) dargestellte logische Datenmodell eines Softwaresystems für einen Tennisclub.

- Der Tennisclub verfügt über Spielstätten
- Eine Spielstätte muss von den Mitgliedern gegen Bezahlung gebucht werden
- Ein Mitglied kann einen Gast gegen eine weitere, zusätzliche Gebühr mitbringen

Die Software verfügt über zwei Subsysteme mit den nachfolgend stichwortartig beschriebenen Aufgaben:

- *Club:* Mitgliederverwaltung, Eintreiben und Verbuchung von Gast-Gebühren
- *Spielstätte:* Verwaltung von Platz-Buchungen, Eintreiben und Verbuchung von Gebühren zur Platzbenutzung



Bitte führen Sie folgende Schritte durch:

1. Markieren Sie die Bounded Contexts von Club und Spielstätte in obigem Datenmodell, z.B. durch Einkreisen.
2. Wenn es Überlappung(en) gibt, dann wählen Sie bitte einen passenden Abgrenzungstyp.
    a. Begründen Sie Ihre Wahl kurz in 1-2 Sätzen.
    b. Bei asymmetrischen Typen (Customer/Supplier, Conformist), legen Sie bitte unbedingt fest, welches Subsystem welche Rolle übernimmt.

Aufgabe 18. Komponenten-Schnittstellen

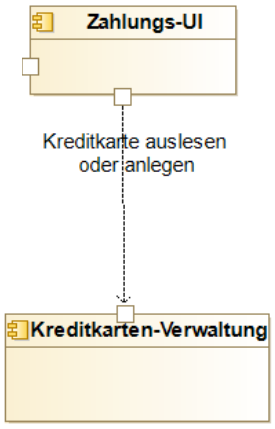
a) Im ersten Entwurf der Komponentendiagramme für ein Softwaresystem brauchen nur Ports zur Kennzeichnung der Aufrufstellen verwendet zu werden. Diese Ports werden im weiteren Verlauf des Entwurfs durch formal spezifizierte Interfaces ersetzt.

In welchen Fällen können Sie darauf **verzichten**, einen (von woanders aufgerufenen) Port in Ihrem Komponentendiagramm durch ein Interface zu ersetzen?

b) Wofür können Sie das logische Datenmodell verwenden, wenn Sie Komponenten-Interfaces spezifizieren?

c) Sie sehen hier einen Ausschnitt aus dem Komponentenmodell des ersten Entwurfs, bei dem noch keine Interfaces spezifiziert wurden. Spezifizieren Sie daher bitte ein Interface **IKreditkartenVerwaltung** für die Komponente Kreditkarten-Verwaltung, das folgende Funktionalitäten bereitstellt:

- Abfragen einer existierenden Kreditkarte für einen Nutzer
- Anlegen einer neuen Kreditkarte für einen Nutzer



Geben Sie für das Interface bitte die Signatur an.

Zeichnen Sie bitte eine neue Version des rechts stehenden Komponentendiagramms, unter Verwendung des neuen Interfaces.

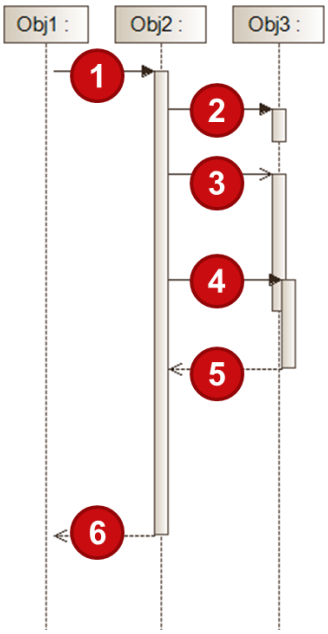
Hinweis: Das logische Datenmodell enthält u.a. die Entitäten Konzert, Konzertticket, Nutzer, Zahlungsdaten, Kreditkarte.

Aufgabe 19. Datenaustausch zwischen Komponenten als Sequenzdiagramm

a) Welche Aussagen über das Sequenzdiagramm im Bild rechts sind korrekt? Bewerten Sie diese bitte mit „richtig“ (r) oder „falsch“ (f).

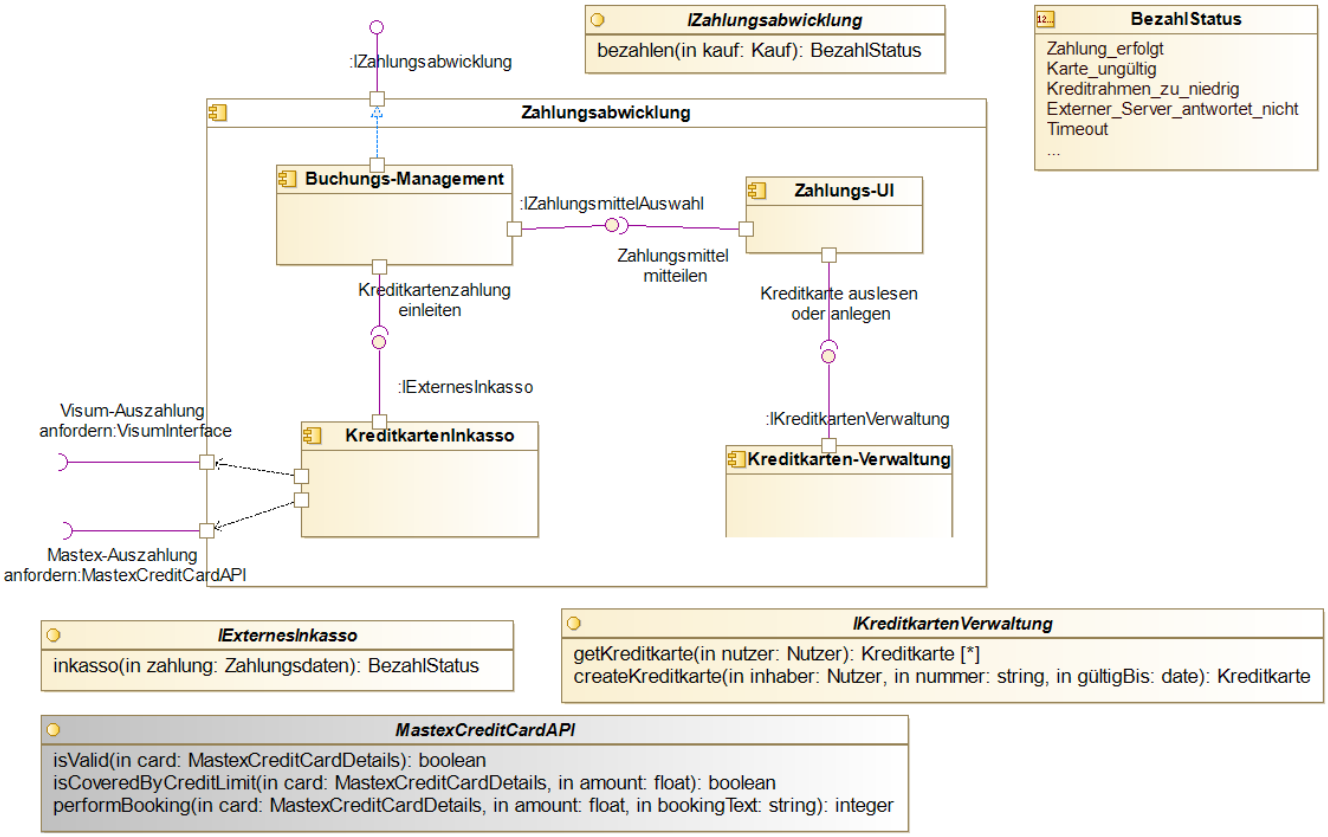
- Wenn Sie die Aussage für **richtig** halten, begründen Sie Ihre Einschätzung bitte jeweils in einem Satz.
- Wenn Sie die Aussage für **falsch** halten, skizzieren Sie die richtige Lösung bitte in einem Satz.

Aussage	**r / f**	**Begründung**
(1), (2) und (4) sind asynchrone Aufrufe		
(2) ist kein synchroner Aufruf, weil die Return-Message fehlt		
Aufruf (3) kann erst getätigt werden, wenn (2) abgeschlossen ist		
Aufruf (4) kann erst getätigt werden, wenn (3) abgeschlossen ist		



b) Nachfolgend ist die Komponente „Zahlungsabwicklung“ eines Portals zum Kauf von Konzertkarten abgebildet, mitsamt den dafür spezifizierten Interfaces.

Bilden Sie folgende Aufrufsequenz (Erfolgsfall) in einem UML-Sequenzdiagramm ab:

1) Die Komponente **Buchungs-Management** stößt das Inkasso bei der Komponenten **KreditkartenInkasso** an.
2) **KreditkartenInkasso** prüft nacheinander die Gültigkeit der Karte und die Abdeckung des Betrags durch das Kreditkartenlimit. Danach wird die Buchung durchgeführt.

Verwenden Sie im Sequenzdiagramm bitte die Methoden der dargestellten Interfaces.