

Name:	GIEßER		
Vorname:	FRANJO		
Matrikelnummer:			
		Punkte von max. 73	
		Punkten:	

Die bei jeder Frage maximal erreichbare Punktzahl ist in Klammern angegeben.

(insgesamt 9 Hauptfragen auf 9 Seiten)

Als Hilfsmittel sind 4 handgeschriebene Seiten erlaubt.

Klausuraufgaben bitte handschriftlich (ggf. per Tablet) beantworten.

Antworten entweder in digitale oder ausgedruckte Klausur eintragen oder in separater Datei bzw. auf leere Blätter schreiben, dann bitte jeweils die Nummer der Aufgabe dazuschreiben.

Später die Klausur selbst handschriftlich korrigieren und bis Mo. 31.1., 12 Uhr auf ELLI hochladen (wird nicht benotet!!)

✓ 1. Modelle und UML in der Softwareentwicklung:
(Bitte in ganzen Sätzen erläutern)

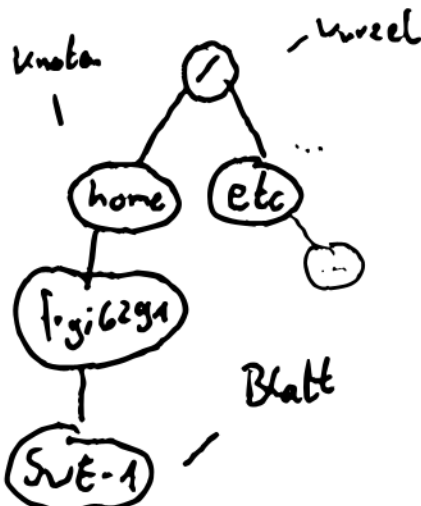
(10 Punkte)

10/10

- ✓ a. Aus welchen wesentlichen Elementen bestehen Bäume in der Informatik, und was kann damit modelliert werden? Zeichnen Sie dazu auch ein kleines Beispiel. (4)

4/4

Wesentliche Elemente eines Baums in der Informatik ist die Wurzel (erster / oberster Knoten), die Knoten, die durch Kanten verbunden werden und die Blätter (die letzten Knoten an den Ästen). Bäume können hierarchische Beziehungen zwischen Elementen modellieren (Beispiel: Dateisystem einer Festplatte).



✓ 1b. Was wird unter dem pragmatischen Merkmal von Modellen (nach Stachowiak) verstanden? (2)

2/2

Das pragmatische Merkmal von Modellen ist, dass Modelle je nach Zweck des Modells, einen bestimmten Ausschnitt des Originals modellieren.

✓ 1c. Welches ist der Zweck der UML?

(Nennen Sie mindestens zwei Aspekte, und erläutern Sie diese in ganzen Sätzen.) (4)

4/4

Der primäre Zweck der UML ist die standardisierte Notation. Dadurch kann in allen Stadien der Softwareentwicklung der Überblick behalten werden und sie vereinfacht die Kommunikation zwischen verschiedenen Personen (-gruppen) innerhalb des Software-Projekts.

Technische Systeme visualisieren.

Kommunikation Auftraggeber, Entwickler

✓ 2. Anwendungsfalldiagramm

(9 Punkte)

Erstellen Sie ein Anwendungsfalldiagramm für ein **Fahrrad-Vermietungssystem** nach folgender Beschreibung:

Für den **Fahrradvermieter** sind die folgenden Anwendungsfälle (Use Cases) definiert:

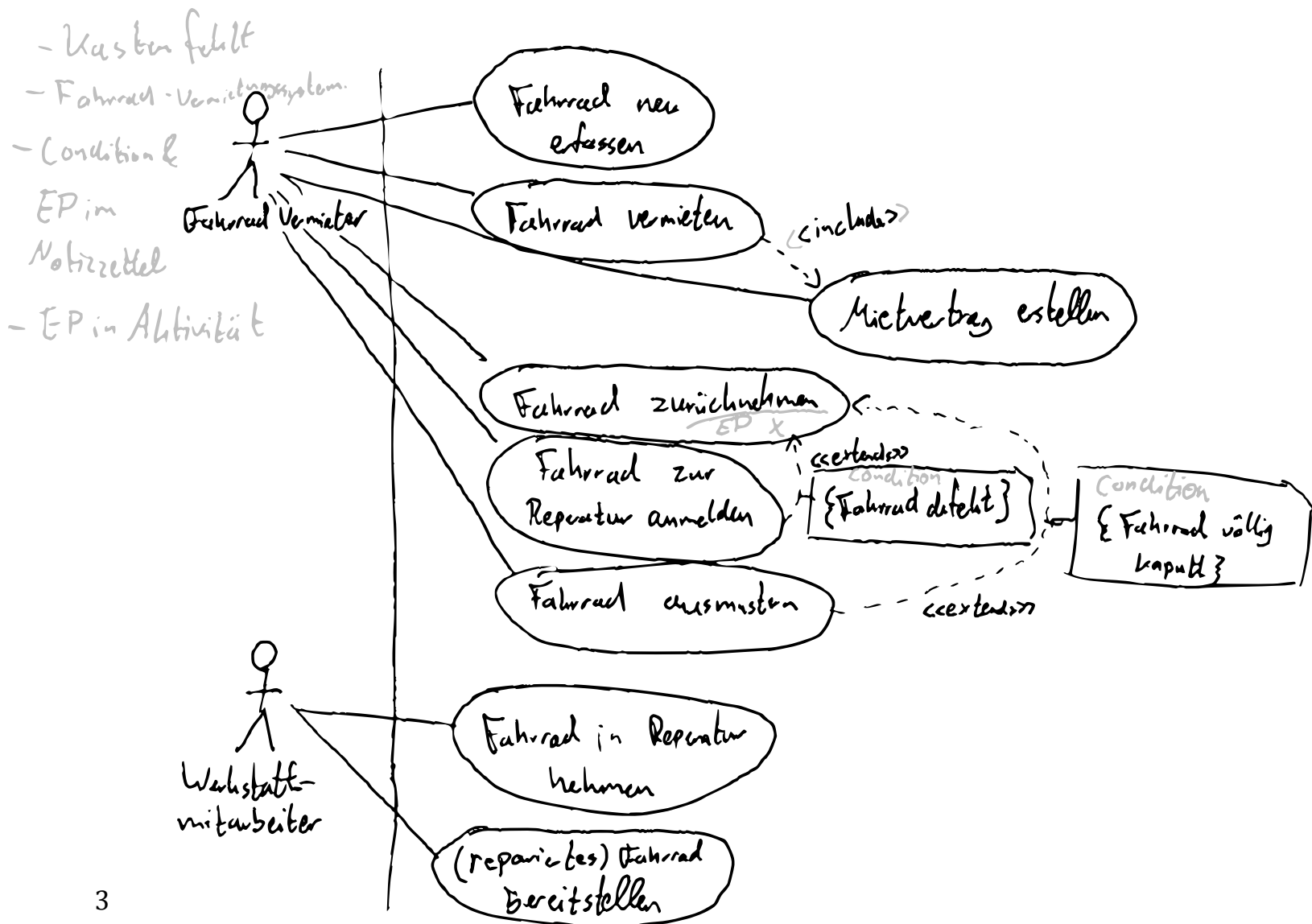
"Fahrrad neu erfassen", "Fahrrad vermieten", "Mietvertrag erstellen", "Fahrrad zurücknehmen", "Fahrrad zur Reparatur anmelden", "Fahrrad ausmustern".

Zudem kann ein **Werkstattmitarbeiter** ein "Fahrrad in Reparatur nehmen" und ein "(repariertes) Fahrrad bereitstellen".

Beachten Sie die folgenden Beziehungen zwischen Anwendungsfällen:

- Das **Vermieten** eines Fahrrads führt **immer automatisch** dazu, dass ein **Mietvertrag erstellt** wird.
- Wenn das Fahrrad beim **Zurücknehmen defekt** ist, meldet der **Fahrradvermieter** das Fahrrad zur **Reparatur** an.
- Wenn das Fahrrad beim Zurücknehmen **völlig kaputt** ist, **mustert** der **Fahrradvermieter** das Fahrrad **aus**.

Zeichnen Sie ein entsprechendes UML-Anwendungsfalldiagramm. Zeichnen Sie bei einer **extend**-Beziehung auch den **Notizzettel mit der Bedingung** ein.



3. Ergänzen Sie in den unten stehenden Satzteilen die passenden UML-Diagrammtypen:

(4 Punkte)

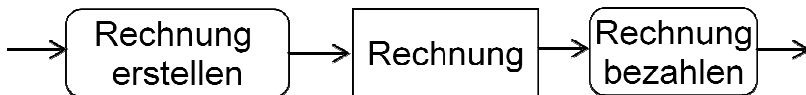
3/4

- a. In Baumdiagramme f. Klassendiagramm können "Ganzes-Teile"-Beziehungen modelliert werden. 0/1
- ✓ b. In Zustandsdiagrammen können Übergänge (Transitionen) durch Ereignisse ausgelöst werden. 1/1
- ✓ c. Optionale und parallele Ablaufmöglichkeiten werden beschrieben in Aktivitätsdiagrammen. 1/1
- ✓ d. Generalisierung und Spezialisierung sind wichtige Konzepte in Anwendungsfelldiagrammen, kommen aber auch in Klassendiagrammen vor. 1/1

Eigentlich falsche Reihenfolge, aber durch Begründung dennoch (2) volle Punktzahl

2/2

✓ 4. Betrachten Sie diesen Diagrammausschnitt:



- ✓ a) Wie bezeichnet man die folgenden Notationselemente?

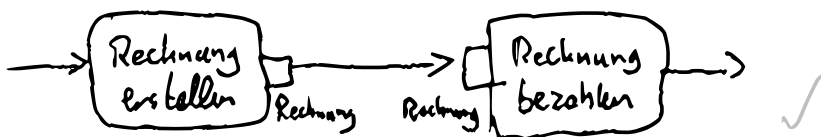
(1) 1/1

Rechnung erstellen : Aktion

Rechnung : Objekt

- ✓ b) Stellen Sie den obigen Diagrammausschnitt mittels der Pin-Notation dar.

(1) 1/1



✓ 5. Modellieren Sie den folgenden Algorithmus als Aktivitätsdiagramm: (11) ^{11/11}

- ✓ a. Zeichnen Sie ein sinnvolles **Aktivitätsdiagramm** mit **"Schwimmbahnen"** (Partitionen) für die **Online-Reservierung** eines **Buches** durch den **Kunden** einer **Bibliothek** mit den unten genannten Aktionen und Bedingungen. (Objekte müssen nicht modelliert werden.)

Nach dem Startknoten beginnt der Ablauf damit, dass der **Kunde ein Buch auswählt**.

Es gibt folgende **Aktionen des Kunden**:

- ✓ **Buch auswählen**, ✓ **"reservieren" anklicken**

und folgende **Aktionen des Systems**:

- ✓ **Buchdetails anzeigen**, ✓ **Verfügbarkeit prüfen**,
✓ **Buch reservieren**, **Buch vorbestellen**

Beachten Sie:

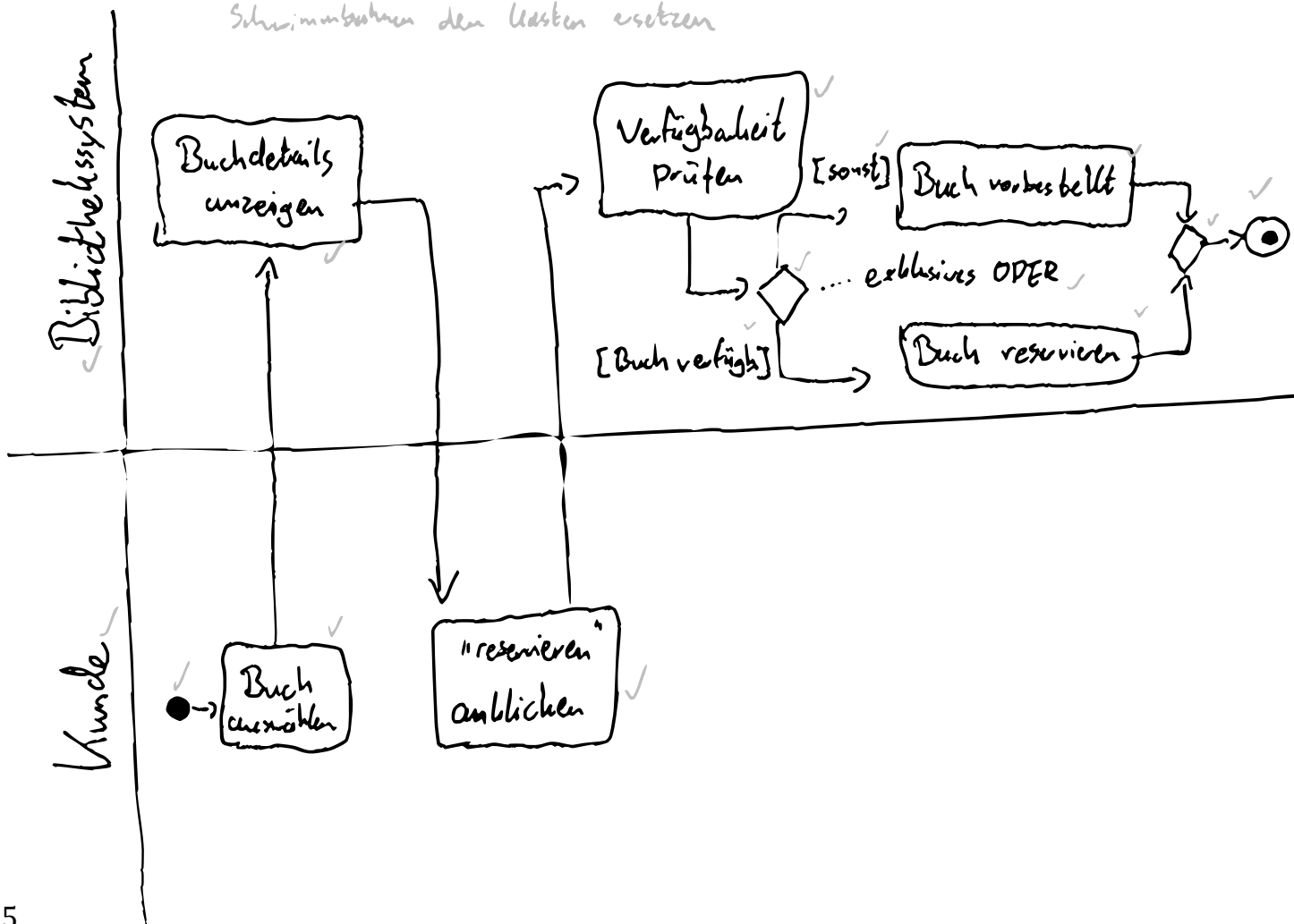
Nach dem Prüfen der Verfügbarkeit gilt die folgende **Bedingung**:

Wenn das Buch **verfügbar** ist, wird es vom System **reserviert**,
sonst wird es vom System **vorgemerkt**.

(10) ^{10/10}

- ✓ b. Markieren Sie, wo in diesem Diagramm ein exklusives „**ODER**“ enthalten ist. (1) ^{1/1}

– Kasten & Aktivitätsname fehlt → aber kein Punktabzug, weil teilweise Schwimmbahnen den Kasten ersetzen



✓ 6. Schleifen

(5 Punkte)

5/5

Schauen Sie sich den folgenden Algorithmus zum Berechnen der Fakultät einer Zahl n an:

✓ Start

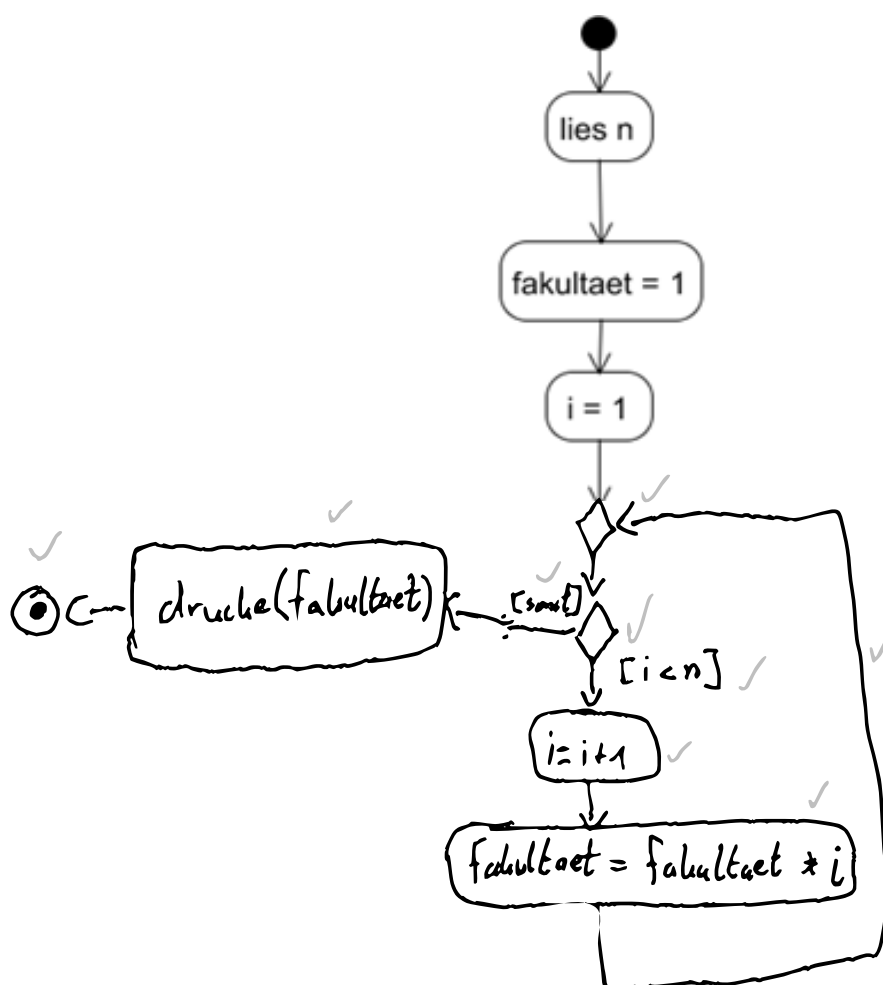
✓ lies n ;
✓ fakultaet = 1;
✓ $i = 1$;

✓ solange $i < n$ {
✓ erhöhe i um 1;
✓ fakultaet = fakultaet * i ;
✓ }

✓ drucke den Wert von fakultaet;

✓ Ende

Erstellen Sie das zugehörige Aktivitätsdiagramm, indem Sie den unten stehenden Beginn des Diagramms entsprechend ergänzen:



7. UML-Klassendiagramm

(22 Punkte)

22/22

Es soll ein Diagramm der fachlich relevanten Klassen einer IT-Anwendung für die Auftragsbearbeitung modelliert werden. Lesen Sie dazu zunächst die folgende Beschreibung:

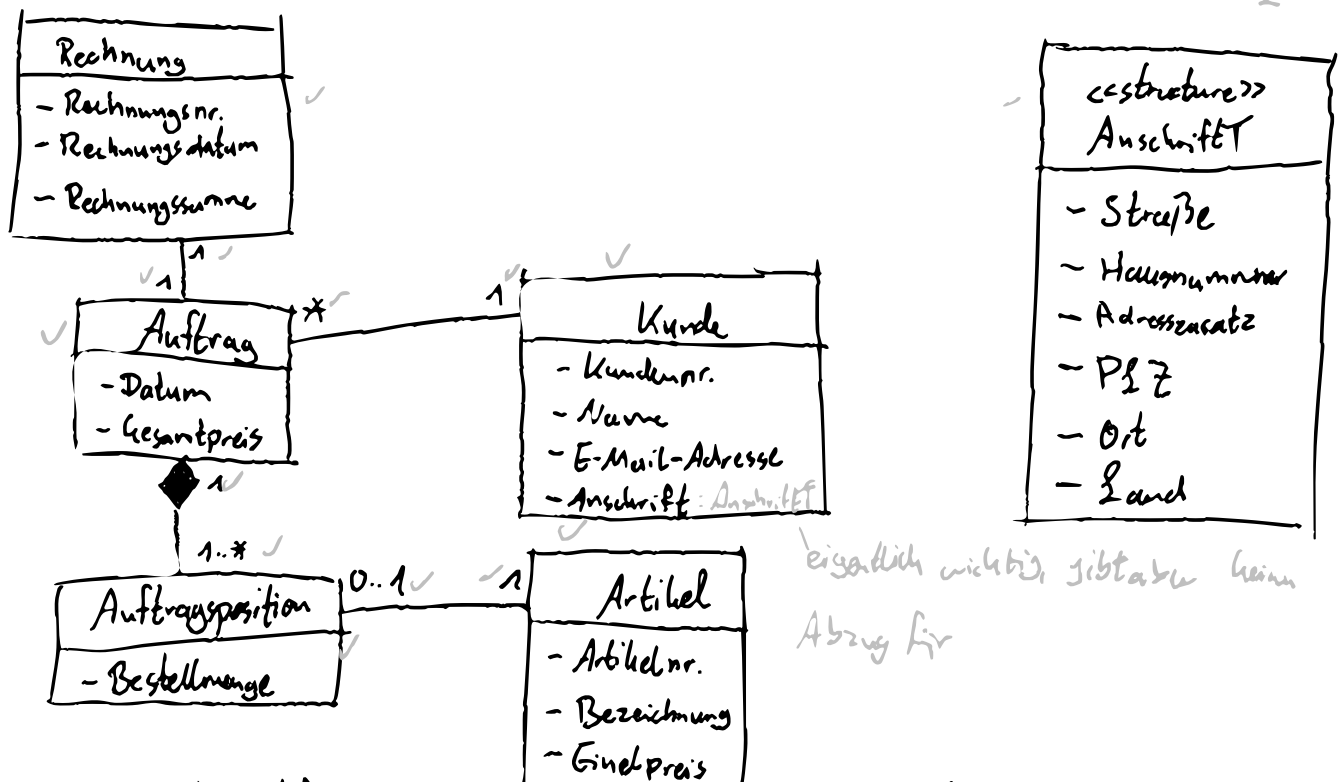
- ✓ Ein Auftrag hat ein Datum und einen Gesamtpreis.
- ✓ Der Auftrag gehört zu genau einem Kunden.
- ✓ Ein Auftrag besteht aus mehreren (aber mindestens einer) Auftragspositionen.
- ✓ Eine Auftragsposition existiert nur als Teil eines Auftrags.
- ✓ Die Auftragsposition enthält Angaben zur jeweiligen Bestellmenge eines Artikels.
- ✓ Jeder Auftragsposition ist genau ein Artikel zugeordnet.
- ✓ Umgekehrt kann ein Artikel nur zu einer Auftragsposition gehören.
- ✓ Ein Artikel hat eine Artikelnummer, eine Bezeichnung und einen Einzelpreis.
- ✓ Kunden haben eine Kundennummer, einen Namen, eine E-Mail-Adresse und eine Anschrift.
- ✓ Zu einem Auftrag wird (genau) eine Rechnung erstellt. Diese hat eine Rechnungsnummer, ein Rechnungsdatum und eine Rechnungssumme.

✓ Modellieren Sie die Klassen mit ihren Attributen, die Sie aufgrund dieser Beschreibung identifizieren können. (Die Datentypen der Attribute müssen nicht angegeben werden.)
Modellieren Sie die Anschrift des Kunden als eigenen Datentyp (elementare Klasse). (12) 12/12

✓ b. Tragen Sie Beziehungen zwischen den Klassen ein, die Sie auf Basis dieser Beschreibung identifizieren können. (4) 4/4

✓ c. Fügen Sie die Multiplizitäten hinzu. (4) 4/4

✓ d. Zwischen welchen Klassen gibt es eine Komposition? Begründen Sie das kurz. (2) 2/2



d) Zwischen den Auftrag und der Auftragsposition existiert eine Komposition.
Ein Auftrag besteht aus Auftragspositionen und die Positionen können nicht ohne einen Auftrag existieren.

✓ 8. Schauen Sie sich das folgende UML-Diagramm an.

(6 Punkte)

6/6

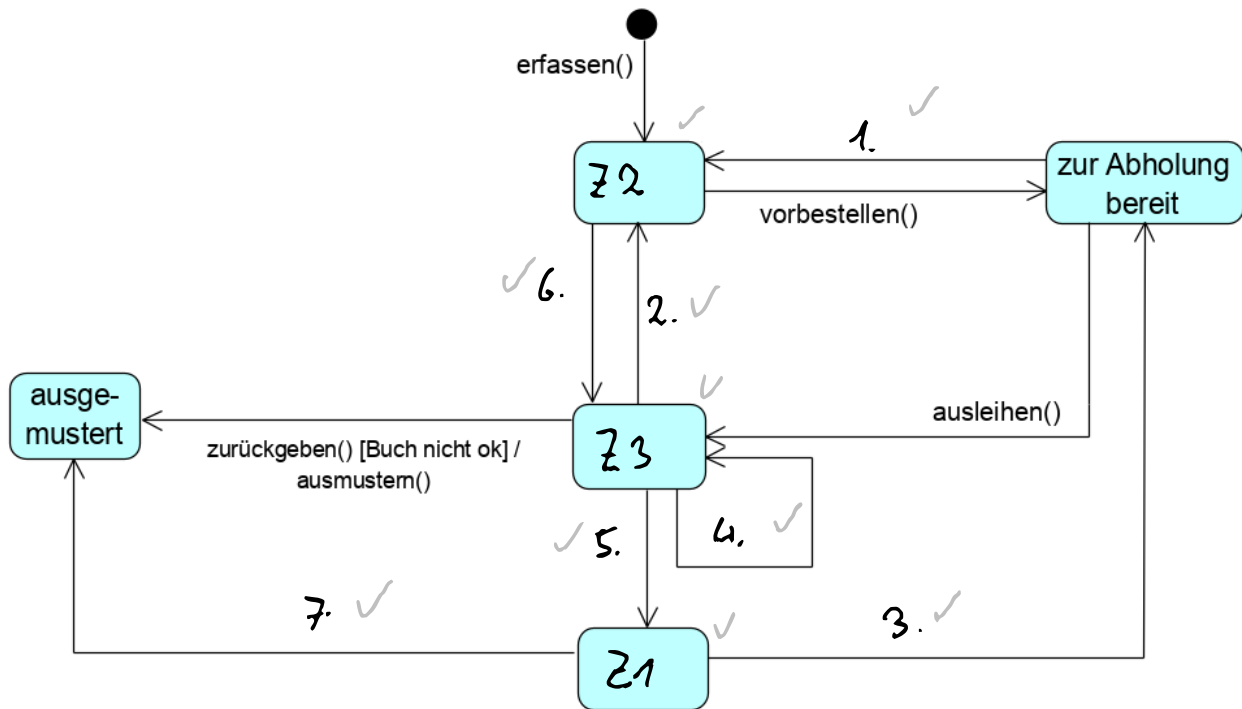
- ✓ a. Um was für einen UML-Diagrammtyp handelt es sich hier?

(1) 1/1

Zustandsdiagramm

- ✓ b. Tragen Sie fehlende Beschriftungen ein bzw. die entsprechenden Ziffern.
Hinweis: Wenn ein zur Abholung bereit gestelltes Buch nach 3 Tagen nicht abgeholt wurde, ist es wieder verfügbar.

(5) 5/5



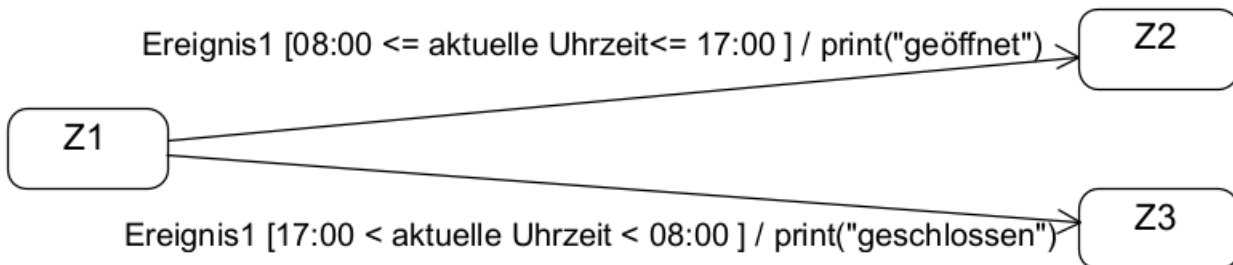
✓ Z1	vorbestellt
✓ Z2	verfügbar
✓ Z3	ausgeliehen

- ✓ 1. after (3 days) / bereitstellen()
 ✓ 2. zurückgeben() [Buch ok] / bereitstellen()
 ✓ 3. zurückgeben() [Buch ok] / abholbereitstellen()
 ✓ 4. verlängern()
 ✓ 5. vorbestellen()
 ✓ 6. ausleihen()
 ✓ 7. zurückgeben() [Buch nicht ok] / ausmustern()

✓ 9. Zustände:

(4 Punkte)

Erläutern Sie in ganzen Sätzen, was bei Eintritt von Ereignis1 passiert, wenn Z1 der aktuelle Zustand ist:



Wenn das Ereignis 1 eintritt wird geprüft, ob die aktuelle Uhrzeit zwischen 08:00 und 17:00 Uhr liegt. Sollte dies der Fall sein, wird die Aktion `print("geöffnet")` ausgeführt und der Zustand Z2 tritt ein.

Sollte der Zeitpunkt, wenn Ereignis 1 eintritt, außerhalb des definierten Zeitfensters liegen, wird die Aktion `print("geschlossen")` ausgeführt und der Zustand Z3 tritt ein.

→ Zweite Bedingung wird auch geprüft, nicht direkt benannt!

Gesamt 67,5 Punkte → 1,3
92,5%