

## Erstklausur

# Softwaretechnik II

Wintersemester 2021/22

Prof. Dr. Ralf H. Reussner

04.04.2022

**Name:** \_\_\_\_\_

**Matrikelnummer:** \_\_\_\_\_

Zur Klausur sind keine Hilfsmittel und kein eigenes Papier zugelassen. Die Bearbeitungszeit beträgt 90 Minuten. Die Klausur ist vollständig und geheftet abzugeben. Verwenden Sie ausschließlich dokumentenechte Stifte. Mit Bleistift oder roter Farbe geschriebene Angaben werden nicht bewertet. Lösen Sie die Aufgaben in leserlicher Schrift. Unlesbare Lösungen können naturgemäß nicht positiv bewertet werden.

<b>Aufgabe</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b><math>\Sigma</math></b>
Maximal	10	16	20	12	16	16	90
K1							
K2							
K3							

## Aufgabe 1: Wissensfragen (10 Punkte)

- a) Nennen Sie stichpunktartig zwei Hauptcharakteristiken von *Software as a Service* Cloud-Anwendungen aus der Vorlesung. (2 Punkte)

- b) Nennen Sie die vier Bestandteile eines standardisierten Containerabbilds (*Container Image*). (2 Punkte)

- c) Nennen Sie drei der fünf Dimensionen der Verlässlichkeit (*Dimensions of Dependability*) aus der Vorlesung Software-Qualität. (1,5 Punkte)

Name:

Matrikelnummer:

---

- d) Handelt es sich bei dem im folgenden beschriebenen Echtzeitsystem um ein Monitoring- (*monitoring*) oder Kontrollsystem (*control system*)? Begründen Sie Ihre Antwort kurz. (1 Punkt)

*Der Landeassistent einer Mondlandefähre überwacht, wenn aktiviert, die Fallgeschwindigkeit im Landeanflug und zündet, sobald eine Fallgeschwindigkeit von 1 m/s überschritten wird, selbstständig kurzzeitig die Bremstriebwerke.*

- e) Nennen Sie die drei Echtzeitentwurfsmuster aus der Vorlesung, die keinen *Fail-Safe*-Zustand benötigen, um die Zuverlässigkeit zu erhöhen. (1,5 Punkte)

- f) Nennen Sie vier der zehn Prinzipien zur Entwicklung sicherer Software aus der Vorlesung. (2 Punkte)

## Aufgabe 2: Anforderungserhebung (16 Punkte)

Im Folgenden betrachten wir die Entwicklung eines Softwaresystems für ein Computerfachgeschäft (*computer retail store*), abgekürzt CRS:

*Dieses Softwaresystem soll zum einen ein Online-Geschäft (online shop) für Ihre Kunden bereitstellen und zum anderen von den Angestellten als Kassensystem (accounting), Lagerverwaltung (stock management) und zur Bestellungsabwicklung (order processing) genutzt werden können. Da das Computerfachgeschäft seinen Kunden zusätzlich die Reparatur eigener Geräte anbietet, muss das Softwaresystem auch die Abwicklung von Reparaturen in der Werkstatt (repair workshop) unterstützen.*

Während der Anforderungserhebung wurden für das Online-Geschäft unter anderem die folgenden Anforderungen identifiziert:

---

Nr.	Anforderung
-----	-------------

---

- |      |   |
|------|---|
| /R1/ | Der Softwareentwicklungsprozess muss den ISO Standard 9001 für die Entwicklung, Wartung und Qualitätssicherung erfüllen.  |
| /R2/ | Die Anzahl Klicks, die ein Kunde im Online-Geschäft benötigt, um ein Produkt zu kaufen, muss minimal sein.  |
| /R3/ | Klickt ein Kunde auf Registrieren, erhebt das Online-Geschäft zuerst die Kundendaten und legt dann ein nicht freigeschaltetes Kundenkonto im System an. Abschließend wird eine Bestätigungsmail an den Kunden versandt. |
| /R4/ | Die von einem Kunden erhobenen Daten sind eine eindeutige Email-Adresse, der Vor- und Nachname, ein Passwort sowie ein Anschrift.   |
| /R5/ | Zur Erhöhung der Datensicherheit darf das Online-Geschäft keine Passwörter im Klartext speichern.   |
| /R6/ | Das Online-Geschäft muss alle Vorgaben der Datenschutz-Grundverordnung (DSGVO) erfüllen.  |
| /R7/ | Sobald der Link einer Bestätigungsmail aufgerufen wird, schaltet das Online-Geschäft das zugehörige Kundenkonto automatisch frei.   |
| /R8/ | Das Online-Geschäft muss bei 100.000 gleichzeitigen Kunden eine Antwortzeit von unter 1s für jede Seitenanfrage haben.  |
-

Name:

Matrikelnummer:

- a) Nennen Sie vier weitere Stakeholder, die neben *Kunden* und *Angestellten* für die Entwicklung des Online-Geschäfts relevant sind. (2 Punkte)

- b) Klassifizieren Sie die oben stehenden Anforderungen nach der in der Vorlesung behandelten Klassifikation von Glinz. Ordnen Sie hierzu die acht Anforderungen des Online-Geschäfts den in der Vorlesung genannten Anforderungsarten (*kind: functional, quality, constraint*), deren Unterkategorie (*subkind: functions, data, performance, security, availability, legal, ...*) sowie deren Repräsentation (*representation: operational, quantitative, qualitative, declarative*) zu. (12 Punkte)

Nr.	Anforderung	Art	Unterkategorie	Repräsentation
/R1/	ISO 9001			
/R2/	Minimale Anzahl Klicks			
/R3/	Kunde registrieren			
/R4/	Kundendaten			
/R5/	Keine Passwörter im Klartext			
/R6/	DSGVO			
/R7/	Bestätigungsmail			
/R8/	100.000 Kunden			

Name:

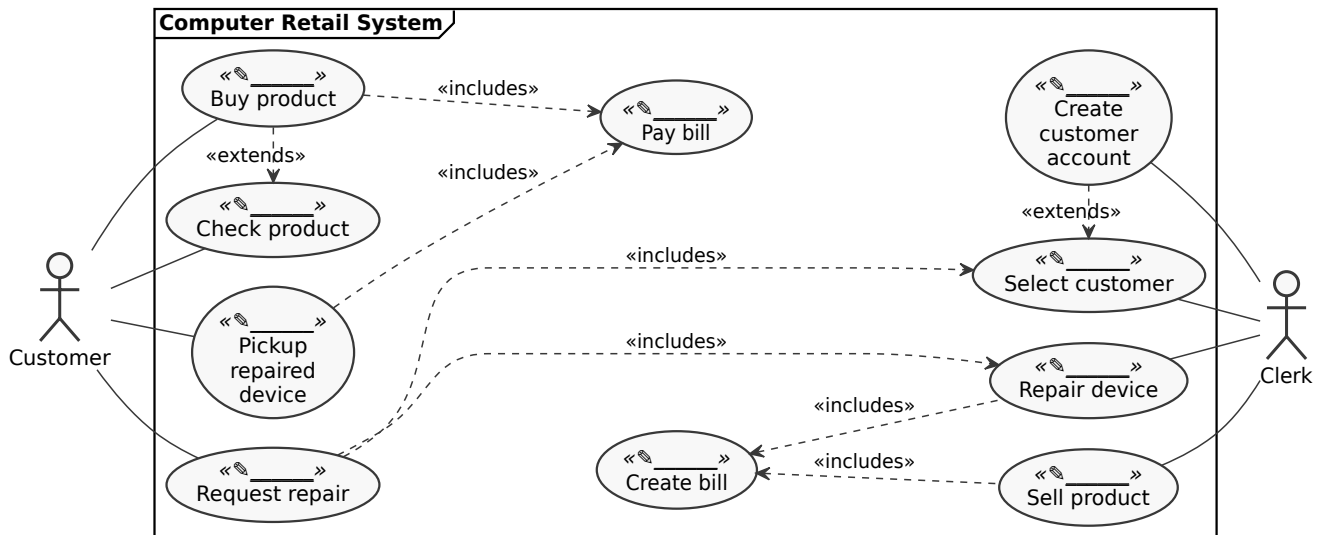
Matrikelnummer:

---


- c) Einige der oben stehenden Anforderungen verstoßen gegen die Empfehlungen für gut geschriebene Anforderungen (*basic writing recommendations*) aus der Vorlesung. Finden Sie **zwei** Anforderungen die gegen **unterschiedliche** Richtlinien verstoßen. Nennen Sie jeweils die Nummer der Anforderung und Richtlinie, gegen die verstoßen wurde. (2 Punkte)

## Aufgabe 3: Anwendungsfallbeschreibung (20 Punkte)

Diese Aufgabe betrachtet die Anwendungsfälle der Kunden (*customers*), Angestellten (*clerks*) und Ingenieure (*engineers*) des Computerfachgeschäfts. Diese sind im folgenden Anwendungsfalldiagramm zusammengefasst und in der nachfolgenden Tabelle kurz erläutert.



Anwendungsfall	Beschreibung
<i>Check product</i>	Betrachten von Produkten durch einen Kunden.
<i>Buy product</i>	Kaufen von Produkten durch einen Kunden.
<i>Sell product</i>	Verkaufen von Produkten durch einen Angestellten.
<i>Create bill</i>	Erstellen einer Rechnung für Produkte und Leistungen.
<i>Pay bill</i>	Begleichen des Rechnungsbetrags durch den Kunden.
<i>Request repair</i>	Stellen einer Reperaturanfrage durch einen Kunden.
<i>Select customer</i>	Auswahl eines Kunden in der Kundendatenbank.
<i>Create customer account</i>	Erstellen eines neuen Kundenkontoobjekts in der Kundendatenbank.
<i>Repair device</i>	Reparatur eines vom Kunden abgegebenen Gerätes.
<i>Pickup repaired device</i>	Abholen des reparierten Gerätes durch den Kunden.

- a) Klassifizieren Sie jeden Anwendungsfall im obigen Anwendungsfalldiagramm (use case diagram) entweder als Nutzerziel (*user goal*) mit U, als Teilfunktion (*subfunction*) mit S oder als Operation (*operation*) mit O. Tragen Sie hierzu die entsprechende Klassifikation in die mit „“ markierten Bereiche der Anwendungsfälle ein. (5 Punkte)

Da diese Anwendungsfälle nicht detailliert genug sind, ist es Ihre Aufgabe ein weiteres Anwendungsfalldiagramm zu erstellen. Hierzu charakterisiert die folgende Beschreibung die Aufgaben die in der angeschlossenen Reparaturwerkstatt erfüllt werden:

*Für die folgende Beschreibung wird angenommen, dass Angestellte und Ingenieure am CRS angemeldet sind. In der Reparaturwerkstatt (repair workshop) können Angestellte (clerk) Reparaturanfragen entgegennehmen (take repair request). Hierzu wählt der Angestellte zuerst das Kundenkonto im CRS aus (select customer). Daraufhin erstellt er für diesen einen Reparaturauftrag (create repair order). Während der Erstellung kann der Angestellte über eine Schaltfläche einen Garantiebeleg zum Auftrag hinzufügen (add warrenty to repair order), falls dieser vorgelegt wurde. Nach Erstellung des Reparaturauftrags weist der Angestellte den Auftrag einem Ingenieur zu (assign to engineer). Falls für die Reparatur ein bestimmter Ingenieur notwendig ist, kann sich der Angestellte in diesem Schritt eine Liste aller verfügbaren Ingenieure anzeigen lassen (list available engineers). Nach der Zuweisung des Ingenieurs ermittelt der Angestellte die voraussichtliche Reparaturzeit (estimate time to repair). Reparaturen werden ausschließlich von Ingenieuren (engineer) durchgeführt (repair device). Falls hierfür Ersatzteile verwendet werden müssen, kann der Ingenieur deren Kosten über eine Schaltfläche zusätzlich zur Rechnung hinzufügen (add replacement parts to bill). Am Ende der Reparatur erstellt der Ingenieur eine Rechnung (create bill). Im CRS werden alle Ingenieure als spezielle Angestellte betrachtet, die zusätzlich Reparaturen (sowie zugehörige Teilaufgaben) durchführen können.*

- b) Erstellen Sie nun ein Anwendungsfalldiagramm (*use case diagram*) genau so, dass es die Anforderung im obigen Text beschreibt. Zeichnen Sie nur Anwendungsfälle und Aktoren ein, die im Text referenziert werden. Achten Sie darauf, an den geeigneten Stellen Beziehungen zwischen den Anwendungsfällen (*use cases*) und Aktoren (*actors*) zu modellieren. (8 Punkte)



Name:

Matrikelnummer:

---

- c) Verwenden Sie das textuelle Anwendungsfallbeschreibungsschema *Fully-Dressed* zur vollständigen Spezifikation des Erfolgsfalls der Aufnahme einer Reparaturanfrage (*take repair request*). Nutzen Sie hierfür die unten gegebene Schablone. Nennen Sie zwei Stakeholder und geben Sie mindestens eine Erweiterung an. (7 Punkte)

---

**Name d. Anwendungsfalls**  
(*Use Case Name*)

**Reparaturanfrage aufnehmen**  
(*take repair request*)

---

**Zielebene**  
(*Goal Level*)

**Primäraktor(en)**  
(*Primary Actor(s)*)

**Stakeholder(s)**

**Vorbedingungen**  
(*Preconditions*)

**Nachbedingungen**  
(*Postconditions*)


**Primäres Erfolgsszenario**  
(*Main Success Scenario*)

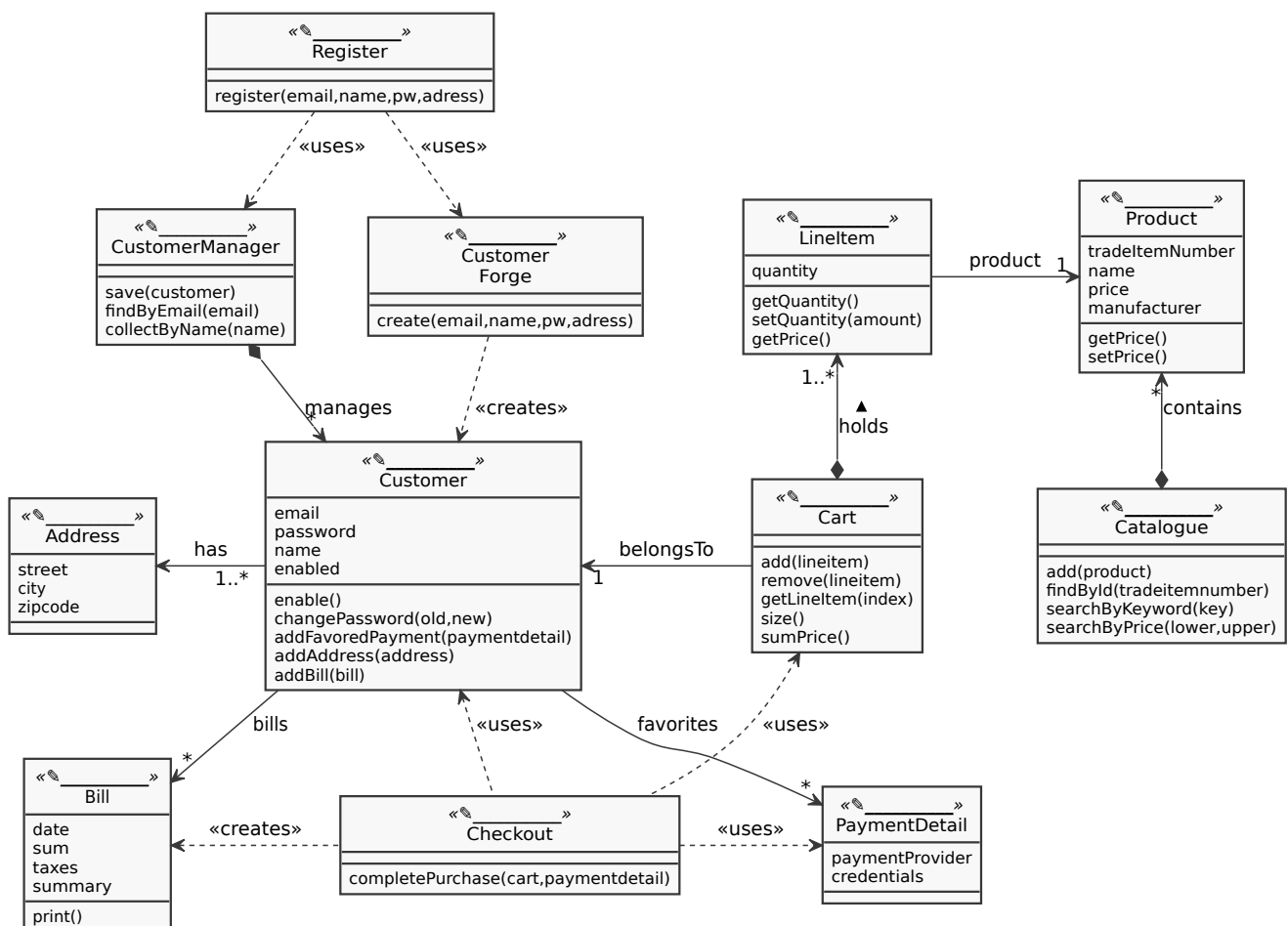
**Erweiterung (Weiterer Ablauf)**  
(*Extension; Additional Flow*)

---

## Aufgabe 4: Domänen-getriebener Entwurf (12 Punkte)

Der folgende Aufgabenkomplex widmet sich der Anwendung des Domänen-getriebenen Entwurfs für das Online-Geschäfts (*online shop*) des Computerfachgeschäfts. Dieses erlaubt es Kunden nach Produkten im Produktkatalog (*catalogue*) zu suchen, ausgewählte Produkte in einen Einkaufswagen (*cart*) zu legen und die Produkte im Einkaufswagen an der Kasse zu bezahlen (*checkout*).

- a) Durch die Domänenanalyse des Online-Geschäfts ist das folgende Klassendiagramm entstanden. Klassifizieren Sie jedes Domänenkonzept entsprechend der Bausteine (building blocks) des Domänen-getriebenen Entwurfs entweder als ValueObject mit V, Entity mit E, Service mit S, Factory mit F oder Repository mit R. Tragen Sie hierzu die Klassifikation in die mit „“ markierten Bereiche der Domänenklassen ein. (6 Punkte)



- b) Begründen Sie Ihre Entscheidung für die Klassifikation der Domänenklasse `Address` in Stichpunkten. (1 Punkt)

- c) Entwerfen Sie ein Aggregat für den Warenkorb (`Cart`), welches sicherstellt, dass der Bestellwert nie 3000€ übersteigen kann. Zeichnen Sie hierfür ein UML-Klassendiagramm, welches ein Paket (*package*) für das Aggregat sowie mindestens die Klassen `Cart` und `LineItem` enthält. Spezifizieren Sie für die Wurzel (*root*) des Aggregates alle neuen oder geänderten Methoden, um Bestelleinträge (`LineItem`) zu erstellen, zu bearbeiten und zu löschen. Markieren Sie alle Klassen, die nicht außerhalb des Aggregates sichtbar sein sollen, mit dem Prefix #. (5 Punkte)

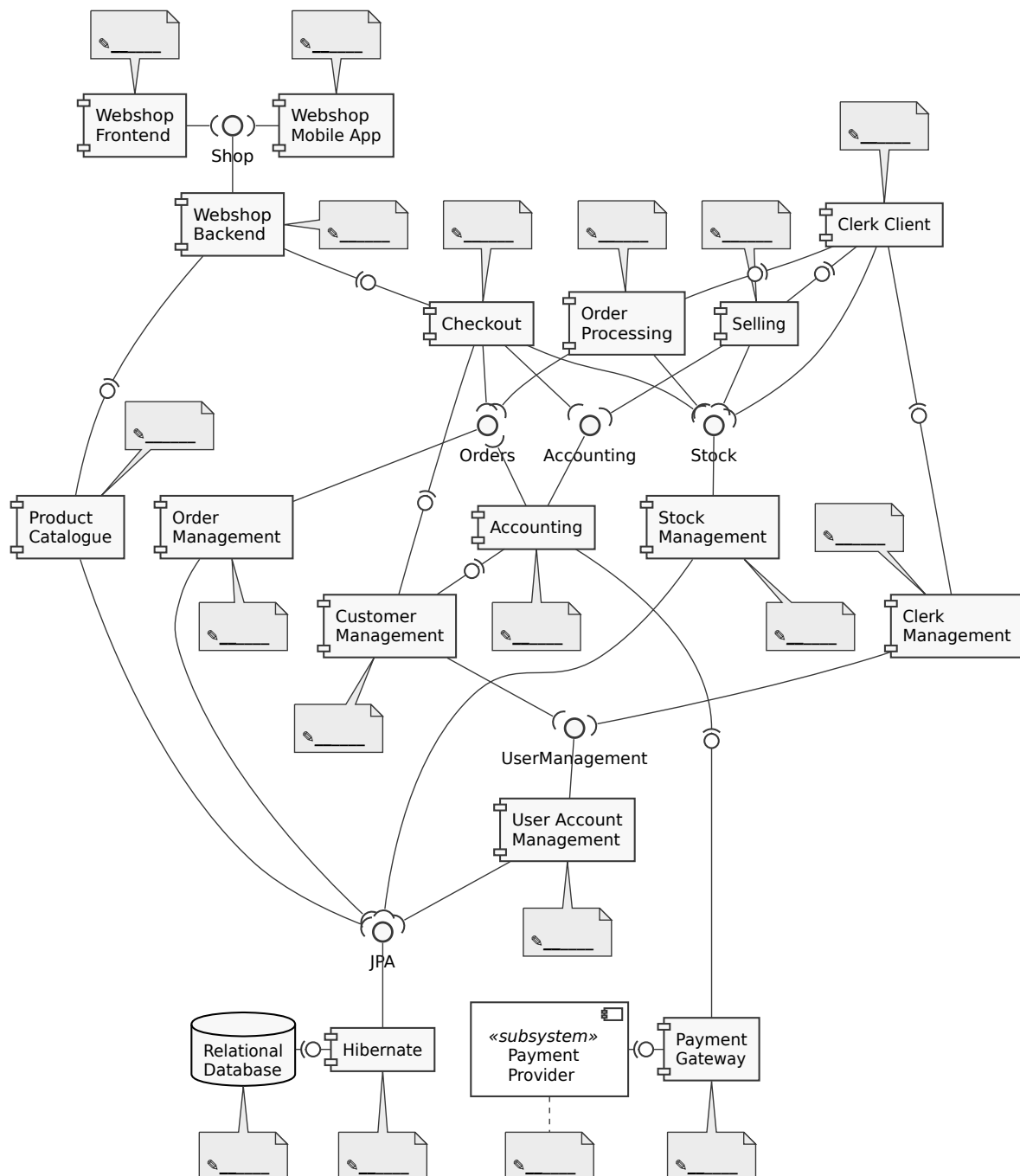
**Hinweis:** Datentypen (wie zum Beispiel `int`, `long`, `Date`, `String`, ...) müssen nicht angegeben werden. Beschriften Sie jede Assoziation mit mindestens einem Rollennamen oder Assoziationsnamen sowie mindestens einer Kardinalität.

## Aufgabe 5: Saubere Unternehmensarchitekturen (16 Punkte)

- a) Tragen Sie in die Tabelle die vier Schichten der *sauberen* Architektur (*clean architecture*) so ein, dass sie zur angegebenen Richtung der Abhängigkeiten passen (*dependency rule*). (2 Punkte)

Schicht	Abhängigkeitsregel	Schichtenname
1.	↓	
2.	↓	
3.	↓	
4.	↓	

- b) Das folgende Komponentendiagramm stellt die Softwarearchitektur des Softwaresystems des Computerfachgeschäfts dar, welches das Online-Geschäft (*online shop*), den Direktverkauf (*selling*), die Buchhaltung (*accounting*), die Lagerverwaltung (*stock management*) sowie die Bestellungsabwicklung (*order processing*) unterstützt. Ordnen Sie nun jede Komponente einer der Schichten *sauberer* Architekturen zu, indem Sie die Nummer der Schicht in die mit „📎“ markierten Bereiche eintragen. (9 Punkte)



Der Quelltext in Listing 1 stellt einen Ausschnitt der Implementierung der Kundenverwaltung (CustomerManager) im Online-Geschäft dar. Genauer werden hier sowohl angemeldete Kunden (Customer) wie auch unangemeldete Seitenbesucher (Anonymous) verwaltet. Die hierfür entwickelte Anwendung enthält jedoch einige Verstöße gegen Praktiken des sauberen Programmierens (Clean-Code-Prinzipien und Coding Conventions aus der Vorlesung).

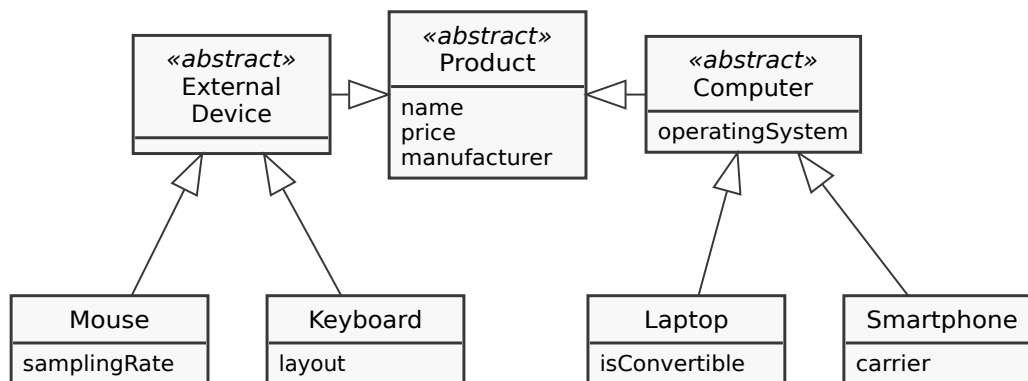
- c) Identifizieren Sie **fünf** Zeilen im Quelltextbeispiel (nächste Seite), die **unterschiedliche** Arten von Verstößen gegen die Praktiken des sauberen Programmierens (Clean-Code-Prinzipien und Coding Conventions aus der Vorlesung) enthalten. Benennen Sie jeweils die **Art** des Verstoßes mit der zugehörigen **Zeilennummer**. (5 Punkte)

```
1 public class Customer{
2     final String email;
3     private String password, name;
4
5     public Customer(String email, String password; String name){
6         if (email==null || email.isBlank()) throw new IllegalArgumentException();
7         this.email = email;
8         this.password = password;
9         this.name = name;
10    }
11    public boolean changePassword(String oldPassword, String newPassword){
12        //if (oldPassword.equals(password) && !newPassword.isBlank())
13        password = newPassword;
14        return true;
15    }
16    public String getName(){ return name; }
17    public String getPassword(){ return password; }
18 }
19 /* Special customer for all users who are not yet logged in */
20 public class Anonymous extends Customer {
21     public Anonymous() { super("anonym@playtech.com", "", "Anonymous"); }
22     public boolean changePassword(){ return false; }
23 }
24
25 public class CustomerManager {
26     private HashMap<String, Customer> customers = new HashMap<>();
27
28     public void save(Customer customer) {
29         if (customer!=null && !customers.containsKey(customer.email))
30             customers.put(customer.email, customer);
31     }
32     public Customer findBy(String email) {
33         for (var customer : customers.values()) if (customer.email.equals(email))
34             return customer;
35         return null;
36     }
37     public List<Customer> collectByName(String name) {
38         // Filter customers by name and collect result as list
39         return customers.values().stream().filter(customer -> customer.getName().equals(name))
40             .collect(Collectors.toList());
41     }
42     public boolean authenticate(String email, String password) {
43         var customer = findBy(email);
44         return customer!=null && !(customer instanceof Anonymous)
45             && customer.getPassword().equals(password);
46     }
47     /* ... */
48 }
```

Listing 1: CustomerManager-Klasse mit Bad Smells

## Aufgabe 6: Entwurfsmuster für Unternehmenssoftware (16 Punkte)

Das Domänenmodell des Computerfachgeschäfts enthält unter anderem die folgende Klassenhierarchie für die unterschiedlichen Produkte die Verkauft werden – Rechner (*computer*) und externe Geräte (*external device*). Um diese Klassenhierarchie in einer Datenbank abzubilden, sollen Sie in dieser Aufgabe die unterschiedlichen Objekt-Relationalen Strukturmuster (*object-relational structural patterns*), welche in der Vorlesung behandelt wurden, anwenden.



- a) Erstellen Sie ein Datenbankschema nach dem Strukturmuster *Concrete Table Inheritance*. Erstellen Sie hierzu ein UML-Klassendiagramm der nötigen Tabellen (dargestellt als UML-Klasse mit dem <<table>> Stereotyp) und ihrer Attribute. Führen Sie, falls nötig, neue Attribute ein, um *Primärschlüssel* und *Vererbung* darzustellen. (7 Punkte)

**Hinweis:** Datentypen (wie zum Beispiel int, long, Date, String, ...) müssen nicht angegeben werden.



Name:

Matrikelnummer:

---

- b) Erstellen Sie ein Datenbankschema nach dem Strukturmuster *Class Table Inheritance*. Erstellen Sie hierzu ein UML-Klassendiagramm der nötigen Tabellen (dargestellt als UML-Klasse mit dem `<<table>>` Stereotyp) und ihrer Attribute. Führen Sie, falls nötig, neue Attribute ein, um die *Primärschlüssel* und *Vererbung* darzustellen. (8 Punkte)

**Hinweis:** Datentypen (wie zum Beispiel `int`, `long`, `Date`, `String`, ...) müssen nicht angegeben werden.

Name:

Matrikelnummer:

---

- c) Geben Sie stichpunktartig einen Vorteil und einen Nachteil der *Concrete Table Inheritance* gegenüber der *Class Table Inheritance* an. (1 Punkt)

Name:

Matrikelnummer:

---

Name:

Matrikelnummer:

---