1 Theoriefragen

1.1 Wieviele Testfälle werden mindestens benötigt (sofern möglich), um bei folgendem Codestück eine: 4 P

a) Anweisungsüberdeckung
Es werden mindestens 2 Testfälle benötigt. ☒ nicht möglich

b) Einfache Bedingungsüberdeckung
Es werden mindestens ____ Testfälle benötigt. ☒ nicht möglich

zu erreichen, oder ist dies nicht möglich? Definieren Sie zwei Testfälle und vermerken Sie, ob diese für a), b) oder beide benötigt werden.

```
read x:
read y:
if x ≥ 50 or y ≤ 100 then
    print "Hello"
else
    print x;
    if x < 50 and y == 100 then
        // empty
    else
        print y;
    end
end
print "World";
```

Testfall 1:
x = 40 y = 70
=> x ≯ 50 ✗ y ≯ 100 ✓ => "Hello" wird ausgegeben ... und Am Ende "World"

Testfall 2:
x = 40 y = 101
=> x ≯ 50 ✗ y ≯ 100 ✗ => 40 (x-Wert) wird ausgegeben and dann "World" (y-invertiert)

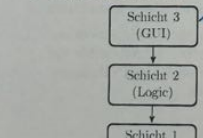
Kommt nie vor, weil vorher in ersten If-Anweisung

Testfall 3: x = 40 y = 100 ✗ => zweiter if-Anweisung kommt nie vor => nicht möglich

Kann auch der Fall werden, dass y gleich 100 wäre.

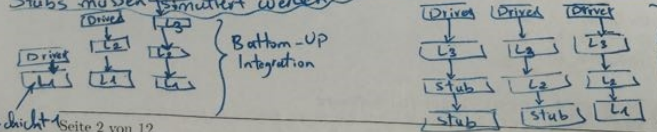
1.2 Erweitern Sie die untenstehende Skizze entsprechend der Vorlesungsfolien und erläutern Sie die beiden Varianten des Horizontalen Integrationstestens anhand der gegebenen Applikationsstruktur. 6 P

Horizontales Integrationstest: Integration Schicht für Schicht. Ein iterative Schichtorientierte Integrationsform. Mit folgenden Varianten:



1) Botton-UP: Hier arbeitet man ausgehend von den hardwarenahen Schichten nach oben vor.
Vorteil: Frühe Integration risikobehafteter Teilen
Nachteil: Hier müssen Driver simuliert werden. Die Kunde sieht lange Keinen Sichtbaren Fortschritt.

2) Top-Down: Die Komponente werden ausgehend von der höher legende Schichten nach unten integriert (Das heißt z.B. zuerst GUI, dann Bussiness Logic dann Data access Layer)
Vorteil: höher legende Schichte sind früher sichtbar Nachteil: Stubs müssen simuliert werden



1.3 Definieren Sie die die Begiffe "Vision" und "Scope" im Bezug auf Softwareprojekte. worin unterscheiden sich diese? 6 P

5

Vision: ~~Ist~~ sind die definierte Projektziele

Scope: ~~Scope~~ Alle Anforderungen die für die Erfüllung der Projektziele gebraut sind.

1.4 Was ist der Unterschied zwischen stabilen und volitalen Anforderungen? Beschreiben Sie drei der in der Vorlesung vorgestellten Methoden zu Ermittlung von Anforderungen. 7 P

S13 EBA1

Stabile Anforderungen sind diejenige die kaum oder nicht geändert werden. Sie ~~können~~ können mit einfachem Design modeliert werden.

Volitale Anforderungen: Diese Anforderungen werden sehr oft geändert, somit ~~brauchen~~ brauchen eine adaptives Design. und detailierbar

Funktionale Anforderungen: Was mit dem Projekt erreichen wird

BSP:
- Kunde muss sich einloggen können
- " " vom Online-Shop Produkt einkaufen können

Nicht funktionale Anforderungen: Wie das Projekt etwas durchführt (hinsichtlich der Qualität, Warbarkeit...)

BSP:
- Speicherung von Passwörten nur mit Verschlüsselung
- Abfrag eines Online Produkt muss nicht länger als zwei Sekunde dauern.

1.5 Stellen Sie den "Strategischer Entwurf" und den "Taktischer Entwurf" anhand von drei Unterschieden gegenüber. 6 P

1.6 Was versteht man unter Testisolation? In welcher Testphase spielt Testisolation die wichtigste Rolle und warum? 6 P

1.7 Nennen und erklären Sie drei der vier in der Vorlesung vorgestellten Vorgangsbeziehungen anhand jeweils eines Beispiels. 6 P

0

1.8 Wozu dienen Design Patterns? Nennen Sie drei Design Patterns der Kategorie "Behavioral Patterns", erklären Sie eines davon anhand eines Beispiels. 7 P

0

1.9 Bestimmen Sie die Äquivalenzklassen im folgenden Beispiel. Erläutern und erstellen Sie eine Grenzwertanalyse für das Beispiel. 6 P

Ein Eissalon führt eine digitale Kundenkarte ein. Wenn Sie die Kundenkarte verwenden, erhalten Sie nach jedem Einkauf ab € 3.00 einen Bonuspunkt und nach jedem Einkauf ab €5.00 zwei Bonuspunkte. Sobald sie 12 Bonuspunkte gesammelt haben, erhalten Sie bei Ihrem nächsten Einkauf einen Rabatt von € 2.00 und die Bonuspunkte werden abgezogen. Sie erhalten nie gleichzeitig Rabatt und Bonuspunkt.

3

Äquivalenzklassen:
$-\infty$ bis 0
0 bis 3
3 bis 5
5 bis $+\infty$

-Grenzwertanalyse ist ein Spezialfall von Äquivalenzklassen, das besagt in den Grenzen der Äquivalenzklassen können Fehler auftreten, und diese Werte der Grenzen werden analysiert.

1.10 Kreuzen Sie alle Aussagen an, die die Prinzipien des agilen Manifests wiederspiegeln. 6 P

4

Aussage	Prinzip des agilen Manifests	
Selbstorganisierte Teams führen zu den besten Resultaten.	ja ☐	nein ☒
Es soll kontinuierlich funktionstüchtige Software ausgeliefert werden.	ja ☒	nein ☐
Testgetriebene Entwicklung führt zu besserer Software.	ja ☒	nein ☐
Changerequests sind zu begrüßen solange sie am Anfang der Entwicklung kommen.	ja ☐	nein ☒
Auch größere Teams können agil arbeiten.	ja ☐	nein ☒
Eine hohe Aufmerksamkeit auf technische Qualität und gutes Design führen zu einer hohen Agilität.	ja ☒	nein ☐

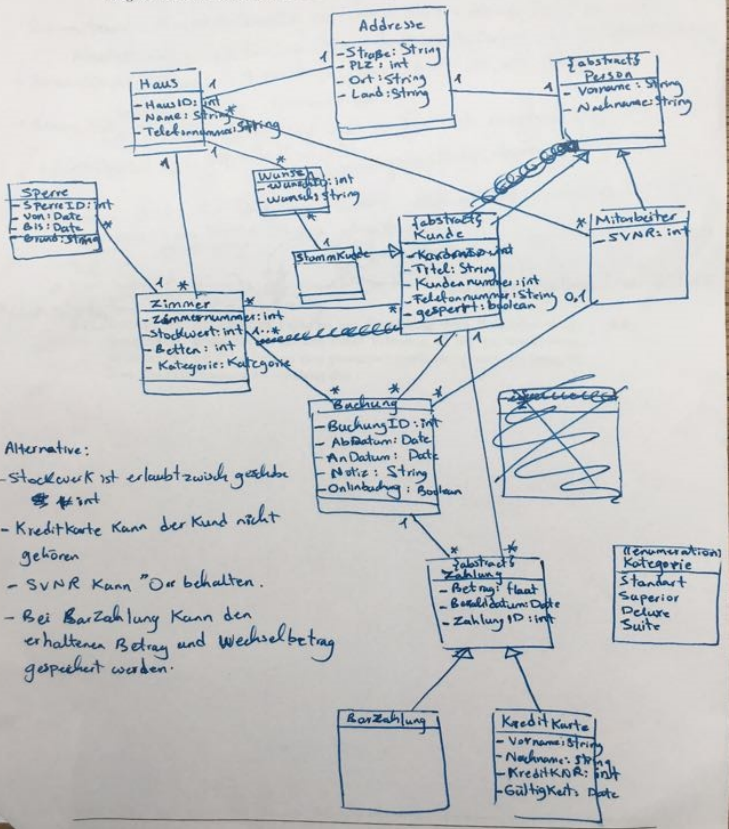
2 Kreativfragen

Gemeinsame Angabe für 2.1, 2.2, 2.3 und 2.4:

Der Besitzer einer kleinen Hotellkette möchte seine Buchungen künftig auch über eine eigene Onlineplattform anbieten. Der Betreiber beauftragt ein mittelständisches Unternehmen mit der Umsetzung. Sie sind der Projektleiter des fünfköpfigen Teams, dass für dieses Projekt verantwortlich ist. Der Betreiber der Hotellkette hat bereits folgende Liste an Funktionalitäten zusammengestellt:

- Häuser: Ein Haus stellt eine „Filiale" der Hotelkette dar. Jedes Haus ist mit Name, Strasse, PLZ, Ort, Land sowie einer oder mehreren Telefonnummern erfasst.
- Kundenstammdaten: Kunden werden mit Titel, Vornamen, Nachnamen, Strasse, PLZ, Ort, Land und einer oder mehreren Telefonnummern erfasst. Eine eindeutige Identifikation geschieht über eine Kundennummer. Es ist notwendig einzelne Kunden zu sperren wenn ein Kunde eine Reservierung nicht in Anspruch genommen beziehungsweise nicht bezahlt hat. Gesperrte Kunden können keine Reservierungen mehr durchführen.
- Stammkundenmanagement: Jeder Kunde kann als Stammkunde ausgezeichnet werden. Wann welcher Kunde als Stammkunde geführt wird, obliegt dem jeweiligen Hotelpersonal, allerdings soll ersichtlich sein, welcher Mitarbeiter den Kunden zu einem Stammkunden gemacht hat. Zu jedem Stammkunden können individuelle Wünsche notiert werden, welche hausspezifisch sind *(z.B.: Kunde XY bevorzugt hofseitiges Zimmer in den höheren Etagen; Kunde Z möchte keine Bananen im Obstkorb; ...)*.
- Erfassen der Zimmer: Zu jedem Haus müssen die Zimmer erfasst werden. Jedes Zimmer gehört einer Zimmerkategorie (Standard, Superior, Deluxe, Suite) an. Weiters wird zu jedem Zimmer die Zimmernummer, das Stockwerk und die Anzahl der Betten erfasst. Sollte ein Zimmer nicht buchbar sein *(z.B. wegen einer defekten sanitären Anlage)*, kann es gesperrt werden. Während des Buchungsprozesses wird geprüft ob das Zimmer gesperrt ist. Im Falle der Buchung eines gesperrten Zimmers wird der Buchungsprozess abgebrochen und der User erhält eine Fehlermeldung. Es soll auch zu einem späteren Zeitpunkt immer nachverfolgbar sein, wann ein Zimmer gesperrt war, aus welchem Grund und von welchem Mitarbeiter.
- Erfassen der Buchungen: Eine Buchung kann ein oder mehrere Zimmer umfassen. Zu jeder Buchung ist ein buchender Kunde zu erfassen, sowie eventuell weitere mitreisende Personen die ebenfalls Kunden sind. Die Anazhl der Betten muss größer oder gleich der Anzahl der Reisenden sein. Jede Buchung enthält zusätzlich Anreisedatum, Abreisedatum sowie eine optionale Buchungsnotiz. Jede Buchung wird einem konkreten Mitarbeiter zugeordnet der die Buchung aufgenommen hat, außer es handelt sich um eine Onlinebuchung.
- Abwicklung der Verrechnung: Wurde eine Buchung bezahlt, so sind zu dieser Buchung die Daten der Zahlung zu speichern. Die Zahlung kann als Barzahlung oder Kreditkartenzahlung durchgeführt werden. Jede Zahlung enthält den bezahlten Betrag und das Bezahldatum. Wird eine Zahlung nicht von dem buchenden Kunden durchgeführt, so muss dies ebenfalls erfasst werden können. Wird eine Kreditkartenzahlung durchgeführt, so sind zusätzlich Vorname, Nachname, Kreditkartennummer und Gültigkeit zu erfassen. Sollte ein Kunde eine Reservierung ohne Stornierung nicht in Anspruch nehmen beziehungsweise nicht bezahlen dann wird dem buchenden Kunden eine Rechnung/Mahnung über den Betrag geschickt. Rechnungen/Mahnungen werden im Abstand von 14 Tagen verschickt. Sollte der Kunde die Rechnung nach der dritten Mahnung nicht bezahlen wird er gesperrt und die Rechnungen wird als Verlust aus dem System ausgebucht.
- Mitarbeiter: Zu jedem Haus werden die Mitarbeiter erfasst. Ein Mitarbeiter ist charakterisiert durch Vorname, Nachname, Strasse, PLZ, Ort, und SVNR.

2.1 Die auszuführende Funktionalität wurde durch den Auftraggeber grob skizziert. Sie sind beauftragt. das Datenmodell für dieses Vorhaben zu entwerfen. Erstellen Sie ein UML Klassendiagramm. welches die Entitäten aus Domänensicht darstellt. Notieren Sie dabei keinerlei Methoden und achten Sie auf die korrekte Syntax sowie die Einhaltung der "Best Practice". Sollten Sie auf dabei auf Widersprüche und Inkonsistenzen in Bezug auf die Angabe stoßen. notieren Sie alternative Lösungsvorschläge für den Kunden. 20 P



Alternative:

- Stockwerk ist erlaubt zwischen gebäude # int
- Kreditkarte kann der Kund nicht gehören
- SVNR kann "0" behalten.
- Bei BarZahlung kann den erhaltenen Betrag und Wechselbetrag gespeichert werden.

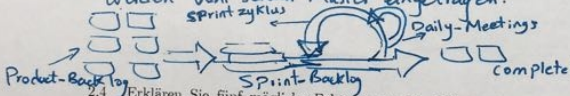
2.2 Erstellen Sie zur Veranschaulichung des Buchungs-/Verrechnungsprozesses ein Zustandsdiagramm (StateChart) welches den Prozess der Buchung *(siehe: Erfassen der Buchungen)* oder der Verrechnung *(siehe: Abwicklung der Verrechnung)* zeigt. Sollten Sie auf dabei auf Widersprüche und Inkonsistenzen in Bezug auf die Angabe stoßen. notieren Sie alternative Lösungsvorschläge für den Kunden. 6 P

Achten Sie auf eine korrekte. UML-Konforme Syntax. (Bedingungen. ...)

2.3 Ihre Firma hat sich für einen Scrum Prozess entschieden. beschreiben Sie seine Komponenten skizzieren und erklären Sie seinen Ablauf. Nennen Sie ein alternatives Prozessmodell. und beschreiben Sie den Hauptunterschied kurz. 8 P

Scrum wichtige Rolle und Elemente:

- Product owner: Erstellt gemeinsam mit der Kunde den Product-Backlog (List aller Anforderungen)
- Scrum_team: 5 - 10 Entwickler mit einem Scrum Master (sorgt für gutes Arbeitsklima) erstellen von Sprint-Backlog (Teilmeng vom Product Backlog
- Scrum-Sprint: Dauer 2-4 Woechen, jedes Sprint ist ein funktionsfähiger Softwareteil
- Scrum Daily-Meeting: Jede sagt was er bis dato geschafft hat bzw. heut schaffen will.
- Sprint Review Meeting: Die Verbeserungsvorschläge werden diskutiert.
- Burn down chart: Ein Diagramm, in dem alle aktuelle Aufwandabschätzu en werden vom Scrum Master eingetragen.



2.4 Erklären Sie fünf mögliche Faktoren zur Ermittlung eines Passendes Vorgehensmodells. Ermitteln Sie anhand dieser Faktoren. ob für das beschriebene Vorhaben der gewählte Prozess eine passende Entscheidung war und legen Sie ihren Entscheidungsprozess schlüssig dar. 6 P