UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR SOFTWARE ENGINEERING
AND PROGRAMMING LANGUAGES

isp

Software Engineering Wintersemester 2018/19
Prof. Dr. Martin Leucker, Torben Scheffel 25.02.2019

Klausur Software Engineering

Aufgaben und Punkte

Die Bearbeitungszeit der Klausur umfasst 90 Minuten. Es gibt 10 Aufgaben mit insgesamt 100 zu erreichenden Punkten.
Notieren Sie Ihre Lösungen wenn möglich direkt auf dem Aufgabenblatt. Sollte der Platz nicht ausreichen, verwenden Sie zusätzliche Blätter, die Ihnen gestellt werden. Benutzen Sie jede Seite der zusätzlichen Blätter nur für genau eine Aufgabe und notieren Sie Ihren Namen und Ihre Matrikelnummer am oberen Rand des Blattes.
Rechts oben auf jeder Seite der Klausur stehen die Punkte für eine gesamte Aufgabe. Die in Klammern gesetzten Zahlen dahinter geben die Punkte für die einzelnen Teilaufgaben an, von links nach rechts, jeweils beginnend mit Teilaufgabe a).
Wenn Sie in einer Aufgabe Lösungen ankreuzen müssen, so erhalten Sie für jedes richtig gesetzte Kreuz Punkte und für jedes falsch gesetzte Kreuz werden Ihnen Punkte abgezogen. Wenn Sie kein Kreuz setzen bekommen Sie weder Punkte, noch werden Ihnen Punkte abgezogen. Die genauen Punktzahlen stehen dabei an jeder Aufgabe, bei der Sie etwas ankreuzen müssen, dabei. Sie können in jedem Aufgabenteil aber nicht weniger als 0 Punkte bekommen.

Persönliche Daten

Notieren Sie im Folgenden Ihre persönlichen Daten. Notieren Sie Ihren Namen und Ihre Matrikelnummer außerdem auf jedem weiteren Blatt der Klausur am oberen Rand sowie auf jedem zusätzlichen Zettel, den Sie benutzen, wie vorher erläutert.

Vorname: _____

Nachname: _____

Geburtsdatum: _____

Matrikelnummer: _____

Studiengang: _____

Viel Erfolg!

Aufgabe 1: Softwareentwicklungsprozess **8 Punkte (2/4/2)**

a) Nennen Sie zwei Besonderheiten, die das Spiralmodell im Vergleich zu generellen, prototyporientieren Vorgehensmodellen, besitzt.

b) Beschreiben Sie den Entwicklungsprozess bei Scrum. Gehen Sie dabei nicht auf die beteiligten Personen ein, sondern beschreiben Sie nur, was in welcher Reihenfolge stattfindet.

c) Entscheiden Sie, ob folgende Aussagen wahr oder falsch sind. Es gibt $+0.5$ Punkte für korrekte Kreuze und -0.5 Punkte für falsche.

	Wahr	Falsch
Das V-Modell basiert auf dem Wasserfallmodell.	☐	☐
Das Lebenszyklusmodell ist ein prototyporientiertes Vorgehensmodell.	☐	☐
Das zu nutzende Vorgehensmodell wird in der Entwurfsphase ausgewählt.	☐	☐
Extreme Programming ist eine inkrementelle Methode.	☐	☐

Aufgabe 2: Management **7 Punkte (5/2)**

a) Beschreiben Sie kurz, was ein Legacy-System ist. Nennen Sie dabei auch drei typische Eigenschaften von Legacy-Systemen.

b) Entscheiden Sie, ob folgende Aussagen wahr oder falsch sind. Es gibt $+0.5$ Punkte für korrekte Kreuze und -0.5 Punkte für falsche.

	Wahr	Falsch
GPL-lizensierter Source-Code kann in proprietären Produkten verwendet werden.	☐	☐
Re-factoring und Reverse Engineering sind das gleiche.	☐	☐
Das Vier-Ohren-Modell wird zum Planen von Besprechungen genutzt.	☐	☐
Walkthrough ist eine manuelle Prüfmethode.	☐	☐

Aufgabe 3: Anforderungsphase **12.5 Punkte (4/2.5/6)**

Ein Kunde möchte von Ihrer Firma eine Software entwickeln lassen und hat nun einen Termin vereinbart, um Ihnen mehr über seine Wünsche und Vorstellungen zu erzählen. Gehen Sie davon aus, Ihre Firma wird den Auftrag annehmen.

a) Beschreiben Sie die Vorgänge in der Anforderungsphase sowie die Unterlagen, die dabei entstehen, ab dem Zeitpunkt, an dem Sie nun zu dem Kunden fahren um seine Wünsche und Vorstellungen aufzunehmen, in chronologischer Reihenfolge.

b) Beschreiben Sie die Vorgänge in der Spezifikationsphase sowie die Unterlagen, die dabei entstehen, nachdem die Anforderungsphase aus Teilaufgabe a) abgeschlossen ist, in chronologischer Reihenfolge.

c) Der Kunde wünscht sich eine Büchereiverwaltungssoftware. Für diese soll es drei unterschiedliche Nutzertypen geben: Büchereinutzer, Büchereiverwalter und Administratoren. Büchereinutzer können Bücher ausleihen und Bücher zurückgeben. Büchereiverwalter können auch Bücher ausleihen und zurückgeben sowie zusätzlich neue Nutzer anlegen und Strafen verhängen, wenn Bücher nicht rechtzeitig zurückgegeben werden. Administratoren können nur das System warten. Alle Nutzertypen müssen sich anmelden, entweder über ein Passwort oder über ihren Fingerabdruck, um etwas in dem System zu machen.
Zeichnen sie ein UML-Anwendungsfalldiagramm für die beschriebene Anwendung.

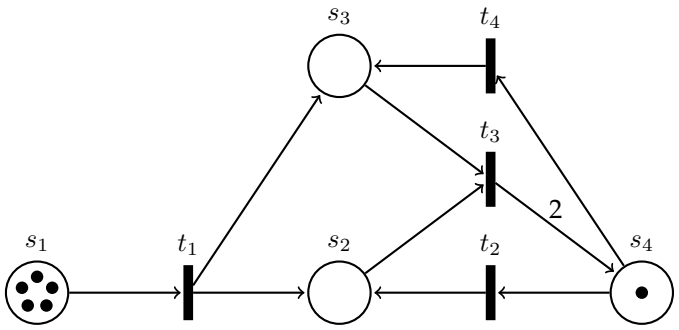
Aufgabe 4: Entwurfs- und Implementierungsphase **7 Punkte (4/3)**

a) Erläutern Sie den Unterschied zwischen UML-Klassendiagrammen und UML-Objektdiagrammen. Erklären Sie auch, wofür welche Diagrammart eingesetzt wird.

b) Nennen Sie drei Arten von Tests und erläutern Sie diese kurz. Eine davon soll die Art von Tests sein, die mit JUnit durchgeführt werden können. Geben Sie an, welche der von ihnen genannten Arten mit JUnit durchgeführt werden kann.

Aufgabe 5: Petri-Netze **7.5 Punkte (2.5/2.5/2.5)**

Es sei folgendes Petri-Netz gegeben:



a) Ist das Petri-Netz lebendig? Begründen Sie die Antwort.

b) Besitzt das Petri-Netz eine Verklemmung? Begründen Sie die Antwort.

c) Ist der Erreichbarkeitsgraph des Petri-Netzes endlich groß? Begründen Sie die Antwort.

Aufgabe 6: Gantt-Diagramm **11 Punkte (9/2)**

Gegeben ist folgendes Software-Projekt zur Entwicklung eines Navigationssystems. Die Vorgangsdauer wird in Tagen angegeben und die Modellierung soll mit Tag 1 beginnen. Ihnen stehen zwei Teams zur Verfügung. Ein Team braucht für eine Aufgabe die angegebene Dauer in Tagen. Eine Aufgabe kann nicht schneller erledigt werden, wenn mehrere Teams daran arbeiten würden.

- Zum Projektstart wird zuerst die **Anwendungsdomäne** erkundet (Dauer: 3 Tage).
- Danach werden in beliebiger Reihenfolge **Routingalgorithmen** analysiert (Dauer: 5 Tage) und **Kartenmaterial** zusammengesucht (Dauer: 2 Tage). Ist das Kartenmaterial zusammengesucht, kann unabhängig von den Routingalgorithmen das Kartenmaterial in eine **Datenbank** getan werden (Dauer: 4 Tage).
- Der Projektleiter hat nach Tag 9 den ersten Meilenstein **M1** angesetzt, da dies ein guter Zeitpunkt ist, um den Stand des Projektes zu evaluieren. Das heißt, die vorher genannten Aufgaben sollen bis zum Meilenstein **M1** nach Tag 9 fertiggestellt sein.
- Nach Meilenstein **M1** wird in beliebiger Reihenfolge mit der **Implementierung der Routingalgorithmen** (Dauer: 2 Tage), der **Dokumentation** (Dauer: 4 Tage) und der **GUI-Entwicklung** (Dauer: 5 Tage) begonnen. Ist die **Implementierung der Routingalgorithmen** abgeschlossen, werden danach dessen **Tests** (Dauer: 4 Tage) durchgeführt. Ist die GUI-Entwicklung abgeschlossen, wird unabhängig vom Rest noch eine **Kundenevaluation** der GUI durchgeführt (Dauer: 2 Tage).
- Zum Abschluss wird dann noch ein Meilenstein **M2** nach Tag 18 festgelegt, zu dem alles fertig sein soll.

a) Entwickeln Sie ein Gantt-Diagramm, dass die oben genannten Schranken einhält. Ihnen stehen zwei Teams zur Verfügung, die parallel Arbeitspakete abarbeiten können. Markieren Sie, was von welchem Team bearbeitet wurde.

Nutzen Sie für die Lösung die Tabelle auf der nächsten Seite!

b) Beantworten Sie danach noch die Frage: Könnte das Projekt mit einem dritten Team früher fertiggestellt werden? Begründen Sie ihre Antwort kurz.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Aufgabe 7: UML-Zustandsdiagramm **10 Punkte**

In dieser Aufgabe soll ein UML-Zustandsdiagramm für eine vereinfachte Flugzeugsteuerung für einen Simulator erstellt werden. Die Flugzeugsteuerung beginnt immer im Zustand *Stehend*. Des Weiteren gilt folgendes:

- Steht das Flugzeug, können entweder die Räder aktiviert werden, wodurch es im Zustand *Rollend* ist oder Schub gegeben werden, unter der Bedingung, dass es auf der Startbahn ist, um zum Zustand *Startend* zu wechseln.
- Rollt das Flugzeug, kann es jederzeit zum Stehen gebracht werden, indem die Räder deaktiviert werden.
- Startet das Flugzeug, fährt es damit so lange fort, bis es in der Luft ist und wechselt dann in den Zustand *Fliegend*. Außerdem wird während des gesamten Starts das Anschnallensymbol angezeigt.
- Fliegt das Flugzeug, werden zuerst auf allen Bildschirmen die *Sicherheitshinweise* angezeigt. Sind die Sicherheitshinweise durch, wird automatisch das *Entertainmentsystem* für die Fluggäste aktiviert.
- Fliegt das Flugzeug, kann außerdem jederzeit das Anschnallensymbol durch das Drücken eines Knopfes vom Piloten aktiviert und deaktiviert werden, wobei es zu Beginn aktiviert ist.
- Gibt es einen Fehler in der Steuerung, wenn das Flugzeug fliegt, wechselt das Flugzeug in einen Fehlerzustand. Ist der Fehler behoben, wechselt es zurück in den Zustand *Fliegend*. Wenn dies passiert, erhält das Anschnallensymbol den Status, den es vor dem Fehler hatte. Waren die Sicherheitshinweise vor dem Fehler bereits durch, wird direkt das Entertainmentsystem aktiviert. Ansonsten werden nochmal die Sicherheitshinweise angezeigt.
- Fliegt das Flugzeug, kann der Schub verringert werden und das Flugzeug wechselt in den Zustang *Landend*.
- Landet das Flugzeug, fährt es damit so lange fort, bis es gelandet ist und wechselt dann in den Zustand *Rollend*. Außerdem wird während der gesamten Landung das Anschnallensymbol angezeigt.

Aus Platzgründen sollten Sie die nächste, komplett freie Seite für die Lösung benutzen!

Aufgabe 8: Algebraische Spezifikation **12 Punkte (12)**

Im Folgenden soll ein abstrakter Datentyp `Trans` für ein Transitionssystem angegeben werden, wobei jeder Knoten durch eine ID aus der Sorte `Nat` identifiziert werden kann. Dabei können Transitionssysteme aus dem leeren Transitionssystem `empty` durch das Anwenden der Operation `addT` erzeugt werden. `addT` fügt dabei eine gerichtete Transition von einem Knoten zu einem anderen ein. Knoten werden beim Einfügen von Transitionen automatisch mit erzeugt. Außerdem gibt es noch drei weitere Operationen:

- die Operation `exists`, die für eine natürliche Zahl prüft, ob ein Knoten mit der Zahl als ID schon in einer Transition in einem Transitionssystem benutzt wird,
- die Operation `sum`, die die IDs aller Knoten aufsummiert, die in den Transitionen benutzt werden (dabei sollen auch IDs mehrfach benutzter Knoten mehrfach gezählt werden), und
- die Operation `swap`, die alle Transitionen umdreht.

Geben Sie eine algebraische Spezifikation für `Trans` an. Dabei dürfen die aus der Vorlesung bekannten Spezifikationen für `Nat` und `Bool` benutzt werden.

Aufgabe 9: Lineare Temporallogik **13 Punkte (2/1/1/6/3)**

Sei im Folgenden $\text{AP} = \{a, b, c\}$ und $\Sigma = 2^{\text{AP}}$.

a) Sei $\neg a \wedge \mathcal{F}\,\mathcal{G}(a \wedge \mathcal{X}\,b)$ eine LTL-Formel.

Geben Sie einen unendlichen Lauf über Σ an, der die Formel erfüllt.

Geben Sie einen unendlichen Lauf über Σ an, der die Formel nicht erfüllt.

b) Wie viele unendliche Läufe über Σ erfüllen die LTL-Formel $\mathcal{G}(a\,\mathcal{U}\,b)$?

c) Wie viele unendliche Läufe über Σ erfüllen die LTL-Formel $(\mathcal{X}\,\mathcal{G}(\neg a \wedge b \wedge \neg c)) \vee (\neg a \wedge \neg b \wedge c \wedge \mathcal{X}\,\mathcal{G}(a \wedge \neg b \wedge \neg c))$?

d) Entscheiden Sie, ob folgende Aussagen wahr oder falsch sind. Es gibt $+1$ Punkt für korrekte Kreuze und -1 Punkt für falsche.

	Wahr	Falsch
$a \wedge b \wedge (c\,\mathcal{U}\,\mathcal{R}\,b) \wedge \mathcal{G}\,\mathcal{F}\,a$ ist eine syntaktisch korrekte LTL-Formel.	☐	☐
$\mathcal{F}(a\,\mathcal{U}\,a)$ ist semantisch äquivalent zu *true*.	☐	☐
Jeder Lauf, der $a\,\mathcal{U}\,b$ erfüllt, erfüllt auch $(\mathcal{F}\,b) \vee (\mathcal{G}\,a)$.	☐	☐
Jeder Lauf, der $a\,\mathcal{U}(b\,\mathcal{U}\,c)$ nicht erfüllt, erfüllt $(\mathcal{F}(\neg a \wedge \neg b \wedge \neg c)) \vee \mathcal{G}(b \wedge \neg\,\mathcal{X}\,c)$.	☐	☐
Der Lauf $\{b, c\}\{c\}(\{a, b\})^\omega$ erfüllt $(\mathcal{G}\,\mathcal{F}\,c) \wedge \mathcal{X}\,\mathcal{X}\,\mathcal{G}\,a$.	☐	☐
Der Lauf $\{a, c\}\{b, c\}\{a, c\}\{b, c\}\{\}(\{a\}\{a, b, c\})^\omega$ erfüllt $(\mathcal{X}(c\,\mathcal{U}\,b))\,\mathcal{U}(a\,\mathcal{U}\,\neg c)$.	☐	☐

e) Zeichnen Sie für die LTL-Formel $((\mathcal{G}\,a)\,\mathcal{U}(\mathcal{F}\,b)) \wedge \neg b \wedge (\mathcal{X}\,\mathcal{X}\,c) \wedge \mathcal{X}\,\neg c$ ein Transitionssystem, bei dem **alle** unendlichen Läufe die Formel erfüllen.

Aufgabe 10: Design-Pattern **12 Punkte (2/3/7)**

a) Erläutern Sie kurz, wozu Design-Pattern generell gedacht sind. Was stellt ein Design-Pattern dar und wofür wird es genutzt?

b) Welche Nachteile sehen Sie in der folgenden Implementierung der Befehle mit denen Tasten in einem Texteditor belegt werden können? Dabei gibt es Einzelbefehle (z.B. create und open), die im Texteditor nur einen Befehl ausführen und zusammengesetzte Befehle, die im Texteditor mehrere Befehle ausführen können (z.B. clear). Die Map und die Befehle, die für einen Wert aus der Map ausgeführt werden, soll der Benutzer dabei selber nach seinen Wünschen zusammenstellen können.

```
Map<Key, String> m = new HashMap<Key, String>();

public execute(Key keyPressed) {
  String c = m.get(keyPressed);
  if (c.equals("create")) {
    editor.create();
  } else if (c.equals("open")) {
    editor.open();
  } else if (c.equals("clear")) {
    editor.selectAll();
    editor.delete();
  } else if ...
  ...
}
```

c) Verbessern Sie den gegebenen Entwurf mit Hilfe von Design-Pattern. Dabei sollen zusammengesetzte Befehle beliebig aus Einzelbefehlen und anderen zusammengesetzten Befehlen erzeugt werden können. Zeichnen Sie ein UML-Klassendiagramm, das ihren Vorschlag für einen besseren Entwurf darstellt.

Korrektur

Diese Seite wird von den Korrektoren ausgefüllt.

Aufgabe	Erreichte Punkte	Mögliche Punkte
1 Softwareentwicklungsprozess		8
2 Management		7
3 Anforderungsphase		12.5
4 Entwurfs- und Implementierungsphase		7
5 Petri-Netze		7.5
6 Gantt-Diagramm		11
7 UML-Zustandsdiagramm		10
8 Algebraische Spezifikation		12
9 Lineare Temporallogik		13
10 Design-Pattern		12
Gesamtpunktzahl		100

Note: _____