



Klausur Software Engineering

Aufgaben und Punkte

Die Bearbeitungszeit der Klausur umfasst 90 Minuten. Es gibt neun 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. Sie können in jedem Aufgabenteil nicht weniger als 0 Punkte erhalten.

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 jeden zusätzlichen Zettel, den Sie benutzen, wie vorher erläutert.

Vorname: _____

Nachname: _____

Geburtsdatum: _____

Matrikelnummer: _____

Studiengang: _____

Viel Erfolg!

Name:

Matrikelnummer:

Aufgabe 1: Anforderungsfestlegung

13 Punkte (4/2/3/4)

- a) Anforderungen unterteilen sich in funktionale und nicht-funktionale Anforderungen. Erläutern sie kurz den Unterschied zwischen den beiden Arten.

- b) Nennen Sie vier Eigenschaften guter Anforderungen.

- c) Welche Gruppierungsmöglichkeiten existieren für die ermittelten Anforderungen?

- d) Beschreiben Sie kurz das allgemeine Vorgehen bei der Anforderungsfestlegung.

Name:

Matrikelnummer:

Aufgabe 2: Planung und Aufwandschätzung

9 Punkte (3/2/1/3)

Im Folgenden wird ein System zur Buchungsverwaltung in einer Pension beschrieben. Die Anwendung soll folgende Funktionalitäten ermöglichen.

- Einen neuen Kunden inklusive Kontaktadresse anlegen. (Datenbestand, Eingabe)
- Eine neue Buchung mit ihren relevanten Merkmalen anlegen. (Datenbestand, Eingabe)
- Detailansicht für einzelne Kunden anzeigen. (Abfrage)
- Detailansicht für einzelne Buchungen anzeigen. (Abfrage)
- Natürlich wird auch eine Übersicht über die aktuellen und kommenden Buchungen benötigt. (Abfrage)
- Terminverlegung einer Buchung (Ausgabe)
- Stornierung einer Buchung unter Beachtung der Stornierungsregelungen (Ausgabe)

- a) Bestimmen Sie den Gesamtwert der gewichteten Function-Points (FP_1). Verwenden Sie dabei das Näherungsverfahren Rapid. Dieses ordnet einer Eingabe 4 fp, einer Ausgabe 5 fp, einer Abfrage 4 fp, einem Datenbestand 7 fp und externen Referenzdaten 5 fp zu. **Machen Sie ihren Rechenweg kenntlich**

- b) Eine erfahrenere Kollegin hat entschieden, dass die Einflussfaktoren 1 bis 10 mit geringer Bedeutung von 2 bewertet werden und die Faktoren 11 bis 14 mit einer Bedeutung von 5.

Berechnen Sie den Einflussfaktoren-Faktor FP_2 . **Machen Sie ihren Rechenweg kenntlich**

- c) Setzen Sie die Werte in die Formel zur Berechnung des FP-Werts ein. **Machen Sie ihren Rechenweg kenntlich**

- d) Erläutern Sie kurz den Unterschied zwischen Referenzdaten und Datenbeständen.

Name:

Matrikelnummer:

Aufgabe 3: Netzpläne

11 Punkte (6/2/2/1)

Im Folgenden soll die Entwicklung des Systems zur Buchungsverwaltung in einer Pension mithilfe eines Netzplans geplant werden. Für die Planung wird angenommen, dass Team und Ressourcen ausreichend groß bemessen sind.

Die Vorgangsdauer wird in Tagen angegeben und die Modellierung soll mit Tag 1 beginnen. In einer ersten Analyse wurde folgendes Vorgehen als Basis für die Entwicklung gewählt:

- Zum Projektstart (an Tag 1) wird die Datenmodellierung für Kunden (1 Tag) und Buchungen (3 Tage) definiert und entsprechende Datenbanken konfiguriert.
- Jeweils aufbauend auf der entsprechenden Datenmodellierung werden die Eingabemasken für das Anlegen von Kunden (3 Tage) und Buchungen (7 Tage) erstellt.
- In Abhängigkeit von den jeweiligen Datenmodellierungen wird parallel zu den Eingabemas-
ken bereits die Detailansichten für einzelne Kunden (2 Tage) und Buchungen (5 Tage) imple-
mentiert.
- Ausgehend von der Detailseite für Buchungen wird auch die Übersichtsseite für Buchungen
erstellt (5 Tage).

Da nach einem inkrementellen Vorgehensmodell gearbeitet werden soll, wurde eine erste Demo über die Kundenverwaltung vereinbart. Dafür ist ein Milestone *Kundenverwaltung* nach Abschluss aller Kunden bezogenen Aufgaben einzuplanen.

Nach Abschluss aller Vorgänge ist ein ausreichender Funktionsumfang erreicht und ein erstes Re-
lease kann erfolgen, diesem ist der *Release1*-Meilenstein zuzuordnen.

- a) Stellen Sie die gegebenen Aufgaben in einem Netzplan dar. Die (gerichteten) Kanten entspre-
chenden dabei den Abhängigkeiten.
- b) Rechnen Sie vorwärts: Berechnen Sie, wann die Vorgänge frühestens begonnen werden kön-
nen und annotieren Sie diese Werte im Netzplan.
- c) Rechnen Sie rückwärts: Berechnen Sie ausgehend vom Meilenstein *Release1*, wann die Vor-
gänge spätestens begonnen werden können (ohne dass sich das Projektende verändert) und
annotieren Sie diese Werte im Netzplan.
- d) Bestimmen Sie sämtliche Pufferzeiten und geben Sie einen kritischen Pfad an.

Verwenden Sie folgende Knotendarstellung:

Vorgang	
frühester Start	spätester Start
Dauer	Puffer

Benutzen Sie aus Platzgründen die nächste Seite im Querformat für die Bearbeitung.

Name:

Matrikelnummer:

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for a drawing or a detailed answer.

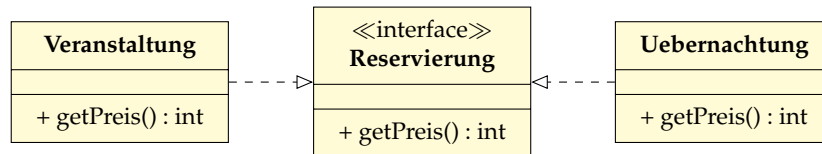
Name:

Matrikelnummer:

Aufgabe 4: Design-Pattern und Klassendiagramme

15 Punkte (4/4/7)

Für das bereits bekannte Buchungssystem existiert aus einem früheren Projekt bereits die folgende Implementierung:



- a) Neben der Buchung von Übernachtungen und Veranstaltungsräumen soll es auch möglich sein, Stadtrundgang und Kutschfahrten zu buchen. Da die Klassen jedoch von anderen Modulen des Systems genutzt werden und die Abhängigkeiten wie immer unbekannt sind, sollen die Klassen in ihrer Struktur nicht verändert werden. So dass eine Übernachtung mit Stadtrundgang und Kutschfahrt sich für die restlichen Module nicht von einer Veranstaltung ohne Extras unterscheidet. Gleichzeitig muss ihr Verhalten jedoch für die neuen Anforderungen erweitert werden. Ihr Kollege schlägt die Erweiterung um entsprechende Klassen

- UebernachtungMitStadtrundgang,
- UebernachtungMitKutschfahrt,
- ...,
- VeranstaltungMitStadtrundgangUndKutschfahrt vor.

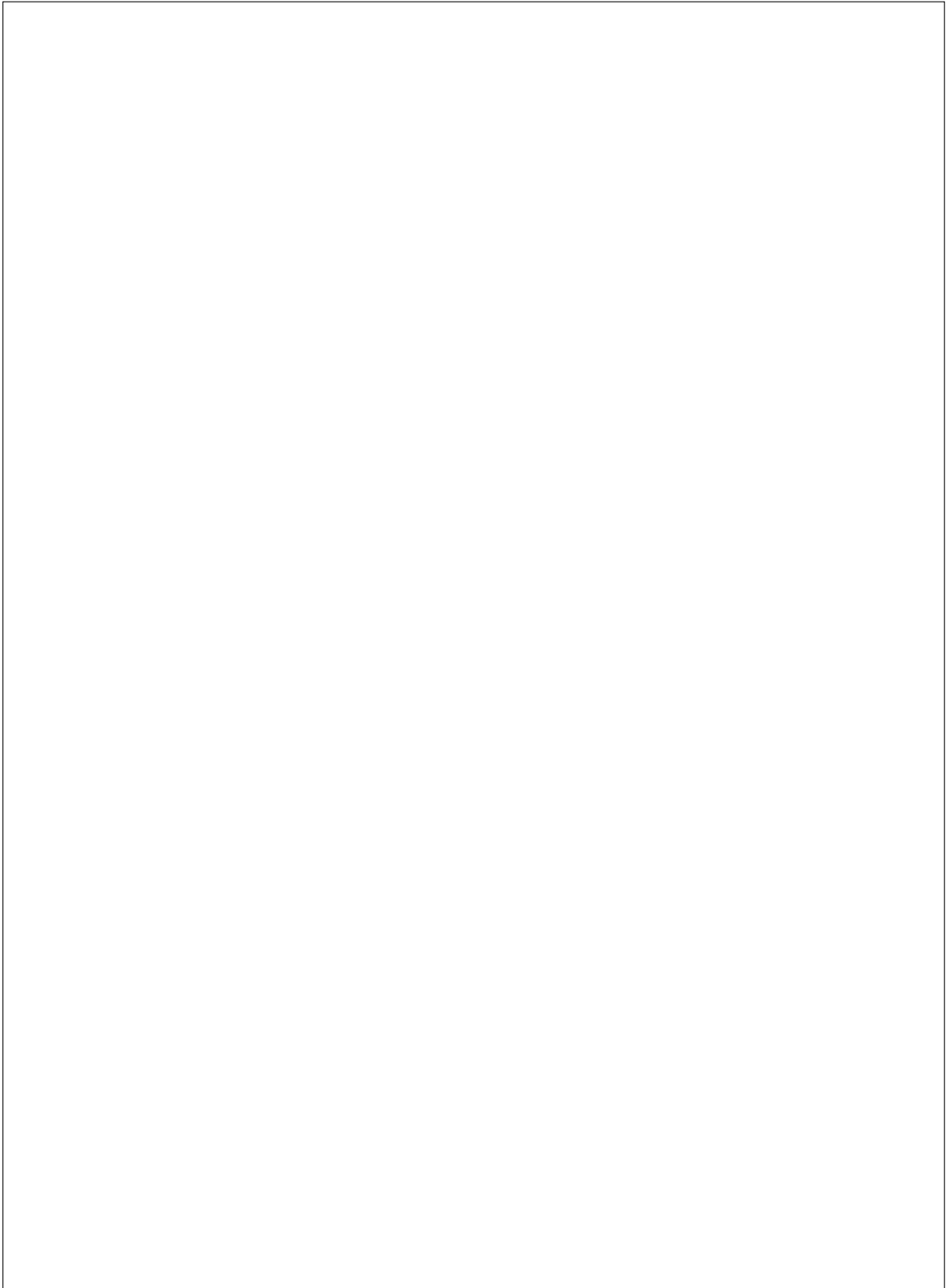
Welche Probleme sehen Sie dabei?

- b) Statt der Erweiterung durch Unterklassenbildung gibt es noch andere Möglichkeiten die Buchung von Extras im System abzubilden. Eine davon ist die Verwendung des Decorator-Patterns. Welche Vorteile hat dies gegenüber der Unterklassenbildung?

- c) Stellen Sie die Realisierung mithilfe des Decorator-Patterns in einem Klassendiagramm dar. Dabei werden die Reservierungen durch eine Decorator Klasse umschlossen, die die Erweiterung des Funktionsumfangs übernimmt.
Benutzen Sie aus Platzgründen die nächste Seite für die Bearbeitung.

Name:

Matrikelnummer:

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for a drawing or a detailed answer.

Name:

Matrikelnummer:

Aufgabe 5: Syntaxdiagramme

10 Punkte

Übertragen Sie die folgender Beschreibung einer *angepassten* FEN in Syntaxdiagramme. Geben Sie nur die Diagramme für **FEN**, **Reihe**, **Rochade** und **enPassant** an.

Die Forsyth-Edwards-Notation (FEN) ist eine Kurznotation, mit der jede beliebige Brettstellung im Schach niedergeschrieben werden kann.

Die FEN ist in 6 Gruppen aufgeteilt. Im Folgenden sei folgende abgewandelte Beschreibung der FEN in EBNF gegeben:

```
FEN := Figurenstellung AmZug Rochade enPassant Halbzuege Zugnummer
```

Die 1. Gruppe beschreibt die Positionen der Figuren auf dem Brett. Mögliche Zeichen sind:

```
Figurenstellung := Reihe "/" Reihe "/" Reihe "/" Reihe "/" Reihe "/" ↔  
                ↔ Reihe "/" Reihe  
Reihe := ( Figur | Leerfelder ) { Figur | Leerfelder }
```

Jede Reihe in einer FEN besteht aus Figuren und/oder Leerfeldern.

```
Figur := "p" | "r" | "n" | "b" | "q" | "k" | "P" | "R" | "N" | "B" | "Q" | "K"  
Leerfelder := "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8"
```

Die 2. Gruppe gibt den Spieler an, der in dieser Stellung am Zug ist. Mögliche Zeichenketten sind (in EBNF):

```
AmZug := "w" | "b"
```

Die 3. Gruppe gibt die noch erlaubten Rochaden an. Mögliche Zeichenketten sind:

```
Rochade := "K" ["Q"] ["k"] ["q"] | "Q" ["k"] ["q"] | "k" ["q"] | "q" | "-"
```

Die 4. Gruppe gibt einen möglichen en passant Zug an. Mögliche Zeichenketten sind:

```
enPassant := "-" | ( Spaltenname ("3"|"6") )  
Spaltenname := "a"|"b"|"c"|"d"|"e"|"f"|"g"|"h"
```

In der 5. Gruppe wird die Anzahl der Halbzüge seit dem letzten Bauernzug bzw. des Schlagens einer Figur angegeben. Dieser Wert ist wichtig, um die 50-Züge-Remisregel zu überwachen.

```
Halbzuege := "0" | positiveGanzeZahl
```

In der 6. Gruppe wird die Nummer des nächsten Zuges angegeben. In der Ausgangsstellung ist der Wert 1. Nach jedem Zug von Schwarz wird um den Wert 1 erhöht.

```
Zugnummer := positiveGanzeZahl
```

Benutzen Sie aus Platzgründen die nächste Seite für die Bearbeitung.

Name:

Matrikelnummer:

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for a drawing or a detailed answer.

Name:

Matrikelnummer:

Aufgabe 6: Software-Entwicklungsprozesse

8 Punkte (5/3)

- a) Erläutern Sie kurz wie sich das Prototypenorientierte Vorgehensmodell von Modellen wie dem Lebenszyklus-, Wasserfall- und V-Modell unterscheidet. Gehen Sie auch auf Vor- und Nachteile ein, die sich durch die Verwendung des Prototypenorientierten Modells ergeben.

- b) Erläutern Sie kurz den Unterschied zwischen nicht-inkrementellen und inkrementellen Vorgehensmodellen und welche Vorteile durch die Verwendung inkrementeller Modelle entstehen.

Name:

Matrikelnummer:

Aufgabe 7: Management

9 Punkte (3/3/3)

- a) Erläutern Sie kurz die Unterschiede zwischen konstruktiver und destruktiver Fehlerkultur.

- b) Nennen Sie drei Risiken die existieren, wenn die Menge der dringenden Änderungen (Ad-Hocs) die Menge der geplanten Änderungen übersteigt.

- c) Nennen Sie drei Vorteile des Re-Engineerings gegenüber der Neuentwicklung eines Softwareprodukts.

Name:

Matrikelnummer:

Aufgabe 8: Zustandsdiagramme

13 Punkte

In dieser Aufgabe soll ein System zur Verwaltung von Alarmsignalen in technischen Prozessen spezifiziert werden. Ein Signal kann deaktiviert sein (`SignalOff`), aktiviert sein (`SignalReady`) und in Betrieb sein (`SignalActive`).

Die zugehörige Alarmbedingung kann aktuell erfüllt sein (`ConditionActive`) oder aktuell nicht erfüllt (`ConditionInactive`). Zudem kann das System selbst ein- und ausgeschaltet sein (`SystemOn`, `SystemOff`). Das System verwaltet die Zustände des Alarmsignals und der Bedingung unabhängig voneinander.

Solange das Alarmsystem ausgeschaltet ist, muss das Signal deaktiviert und die Bedingung im Zustand „nicht erfüllt“ sein.

Sobald das Alarmsystem eingeschaltet wird, beginnt das Signal im Zustand `SignalOff` und die Bedingung im Zustand `ConditionInactive`.

Allerdings kann die Bedingung nun eintreten und wieder aufgelöst werden. Immer wenn die Bedingung eintritt soll das Event `conditionHigh` ausgelöst werden. Ebenso soll im entgegengesetzten Fall das Event `conditionLow` ausgelöst werden.

Außerdem kann nun das Signal aktiviert und wieder deaktiviert werden, dabei ist zu gewährleisten, dass das Signal immer deaktiviert werden kann. Ist das Signal aktiviert, so muss es bei eintreten des Events `conditionHigh` in Betrieb übergehen bis das Event `conditionLow` auftritt. Solange sich das Signal in Betrieb befindet, muss eine Leuchte zum blinken gebracht werden.

Modellieren Sie die Systembeschreibung durch ein Zustandsdiagramm.

Benutzen Sie aus Platzgründen die nächste Seite für die Bearbeitung.

Name:

Matrikelnummer:

Name:

Matrikelnummer:

Aufgabe 9: Lineare Temporallogik

12 Punkte (2/5/5)

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

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

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

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

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

	Wahr	Falsch
$\mathcal{G}(a \rightarrow \mathcal{G}(b \rightarrow (c \wedge \neg d \wedge \mathcal{X}(\neg d \mathcal{U} a))))$ ist syntaktisch korrekt	<input type="checkbox"/>	<input type="checkbox"/>
Der Lauf $\{a\}\{c\}\{b\}\{b\}\{a, c\}^\omega$ erfüllt $\mathcal{G} \neg c \vee \mathcal{F}(c \wedge (\neg a \mathcal{U}(b \vee \mathcal{G} \neg a)))$	<input type="checkbox"/>	<input type="checkbox"/>
Jeder Lauf der $(\mathcal{F} b) \rightarrow (a \mathcal{U} b)$ erfüllt, erfüllt auch $\mathcal{G}(b \rightarrow \mathcal{G} a)$	<input type="checkbox"/>	<input type="checkbox"/>
$(a \mathcal{U} b) \wedge \mathcal{X} \mathcal{G} b$ ist semantisch äquivalent zu a	<input type="checkbox"/>	<input type="checkbox"/>
Ein Lauf, der $(\mathcal{G} a) \mathcal{U} c$ erfüllt, erfüllt auch $\mathcal{F} \neg c$	<input type="checkbox"/>	<input type="checkbox"/>

- c) Zeichnen Sie für die folgenden Formeln jeweils ein Transitionssystem, für das gilt, dass **alle** unendlichen Läufe, die Formel erfüllen.

$$\varphi_1 = \mathcal{G} \mathcal{F}(a \wedge \mathcal{X}(b \wedge \mathcal{X}(c \wedge \neg a))) \quad (1)$$

$$\varphi_2 = \mathcal{G}((a \rightarrow \mathcal{X} \neg a) \wedge (\neg a \rightarrow \mathcal{X} a) \wedge (b \rightarrow \mathcal{X} \mathcal{X} \neg b) \wedge (\neg b \rightarrow \mathcal{X} \mathcal{X} b)) \quad (2)$$

Korrektur

Diese Seite wird von den Korrektoren ausgefüllt.

Aufgabe	Erreichte Punkte	Mögliche Punkte
1 Anforderungsfestlegung		13
2 Planung und Aufwandsschätzung		9
3 Netzpläne		11
4 Design-Pattern und Klassendiagramme		15
5 Syntaxdiagramme		10
6 Software-Entwicklungsprozesse		8
7 Management		9
8 Zustandsdiagramme		13
9 Lineare Temporallogik		12
Gesamtpunktzahl		100

Note: _____