

Nachklausur

## Softwaretechnik II

JProf. Dr. Oliver Hummel

Sommersemester 2013

11. April 2013

*Die Bearbeitungszeit beträgt 90 Minuten.*

Den Aufkleber mit den persönlichen  
Daten bitte hier anbringen!

### Bearbeitungshinweise:

Die Klausur ist vollständig und geheftet abzugeben.

Schreiben Sie nicht mit Bleistift oder roter Farbe, da diese nicht gewertet werden können.

Zur Klausur sind kein eigenes Papier und außer einem nicht programmierbaren Taschenrechner keine Hilfsmittel zugelassen.

Bitte prägen Sie sich Ihre Klausurnummer ein, da wir Ihr Klausurergebnis im Anschluss an die Korrektur im geschützten Bereich der Vorlesungshomepage unter dieser Nummer veröffentlichen. (Diesen Abschnitt bitte streichen, falls nicht gewünscht.)

Es ist garantiert, dass die Klausur mit 45 Punkten bestanden wird und eine 1,0 ab 85 Punkten erreicht werden kann.

Aufgabe	1 (14)	2 (14)	3 (10)	4 (15)	5 (11)	6 (14)	7 (12)	Gesamt (90)
Punkte	a) b) c)	a) b)	a) b) c) d)	a) b) c) d)	a) b) c)	a) b) c)	a) b) c)	- - - -
$\Sigma$								

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

**Aufgabe 1: Software-Requirements (14 Punkte)**

a) Nennen und erklären Sie stichwortartig die 4 Prioritätsstufen des MoSCoW-Modells für Software-Anforderungen. (4 Punkte)

b) Identifizieren Sie das korrekte „Use Case Goal Level“ für die im Folgenden genannten Aktivitäten. (4 Punkte)

Hinweis: Geben Sie eine stichwortartige Begründung für Ihre Entscheidung, falls aus Ihrer Sicht mehrere Möglichkeiten in Frage kommen sollten.

1. Validieren einer in einem Geldautomaten eingegebenen Geheimnummer.

-> \_\_\_\_\_

2. Aufnahme, Bearbeitung und Abschluss eines Schadensfalls in einem Versicherungssystem durch verschiedene Sachbearbeiter.

-> \_\_\_\_\_

3. Auffinden eines Kundendatensatzes in einem Versicherungssystem durch einen Sachbearbeiter.

-> \_\_\_\_\_

4. Auswählen und Reservieren eines Mietwagens über eine Online-Plattform.

-> \_\_\_\_\_

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

1. c) In einem agilen Softwareentwicklungsprojekt soll für eine öffentliche Bibliothek ein webbasiertes Ausleihsystem entwickelt werden, das es den Benutzern erlaubt, Bücher auszuwählen, die ihnen nach Hause gesendet werden. Dazu wurden die folgenden User Stories identifiziert.

ID	User Story	Frequenz	Value
US1	Ich als Kunde möchte den Bibliotheksbestand durchsuchen und die Verfügbarkeit überprüfen können.	Stündlich	Mittel
US2	Ich als Kunde möchte Bücher auswählen und bestellen können, damit sie mir geliefert werden können.	Täglich	Hoch
US3	Ich als Kunde möchte „Reviews“ zu Büchern verfassen und lesen können.	Wöchentlich	Gering
US4	Ich als Kunde möchte eine Übersicht meiner gerade entliehenen Bücher anzeigen können, damit ich weiß, bis wann ich sie wieder zurückgeben muss.	Täglich	Mittel
US5	Ich als Bibliothekar möchte Bücher, die von Kunden wieder zurückgegeben wurden, im System vermerken können.	Täglich	Hoch
US6	Ich als Bibliothekar möchte das Ausleihverhalten der Kunden anonymisiert analysieren können, um andere Kunden besser beraten zu können.	Stündlich	Gering
US7	Ich als Bibliothekar möchte eine Liste mit Kunden anzeigen können, die ihre Bücher regelmäßig zu spät zurückgeben.	Täglich	Mittel
US8	Ich als Bibliothekar möchte eine Liste mit Büchern anzeigen können, die an die Kunden verschickt werden müssen.	Täglich	Hoch

Erstellen Sie nun aus den gegebenen User Stories eine Story Map (nach Patton) und identifizieren Sie darin sowohl die Geschäftsprozesse als auch mögliche Releases unter der Annahme, dass pro Release drei User Stories implementiert werden können. (6 Punkte)

## Aufgabe 2: Software-Entwurf (14 Punkte)

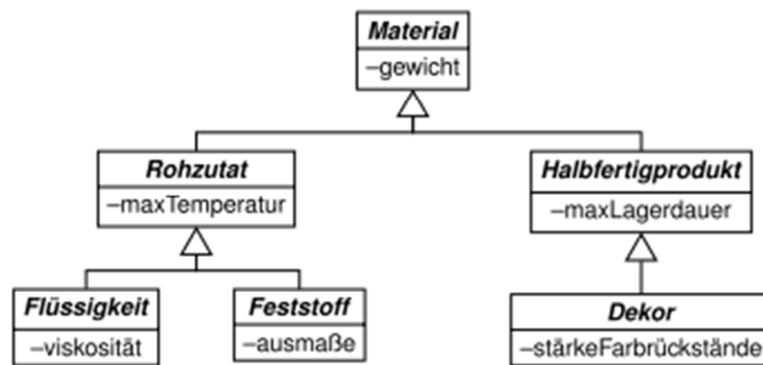
a) Erstellen Sie einen vollständigen graphischen Architekturentwurf für das im Folgenden beschriebene Szenario und erweitern Sie diesen mit den unten angegebenen und ggf. weiteren zentralen Klassen. Berücksichtigen Sie dabei auch angegebene Funktionalitäten und geben Sie für die Domänenobjekte jeweils wenigstens ein sinnvolles Attribut (sprechende Namen!) an. Nutzen Sie die in den Veranstaltungen vorgestellte Kombination eines UML-2-Paketdiagramms und eines Klassendiagramms und visualisieren Sie Abhängigkeiten entsprechend. (8 Punkte)

Szenario: Ein Softwarehersteller möchte ein Verkaufssystem für den Einzelhandel erstellen. Beim Design des Systems sollen eine **mehrschichtige logische Architektur** und das Model-View-Controller-Muster zum Einsatz kommen. Ferner sollen auch folgende Klassen des Domänenmodells Verwendung finden:

- Product, Sale und Cart

Änderungen an Objektinstanzen dieser Klassen werden durch einen ActionLogger gespeichert. Alle Eingaben sollen in einer SaleUI gemacht werden, welche die Funktionalität bietet, einen neuen Kauf zu beginnen, sowie Produkte zu einem bestehenden Verkauf hinzuzufügen und danach eine Übersicht des Warenkorbs anzuzeigen.

2. b) Das Domänenmodell eines Warenwirtschaftssystems enthält die folgenden Entitäten.



**Weitere Hintergrundinformationen (nicht entscheidend für die Bearbeitung):** Es wird zwischen Rohzutaten, Halbfertigprodukten und Produkten unterschieden. Material und Rohzutat sind abstrakte Konzepte. Rohzutaten sind entweder Flüssigkeiten mit Viskosität (Maß für die Zähigkeit) oder Feststoffe (mit Angabe zum Ausmaß). Halbfertigprodukte können Dekor (mit Angaben zur Stärke der Farbrückstände) spezialisiert werden, um den Produktionsprozess zu optimieren.

Die Persistierung in einer relationalen Datenbank soll mit Hilfe eines Data Mapper geschehen. Nennen Sie die drei in der Vorlesung vorgestellten Möglichkeiten, um diese Vererbungshierarchie in der Datenbank abzubilden und geben Sie jeweils an, welche DB-Tabelle(n) angefragt werden müssen, um ein Objekt vom Typ Feststoff zu erhalten. (6 Punkte)

**Aufgabe 3: Security und moderne Entwicklungspraktiken (10 Punkte)**

a) Erklären Sie kurz, wieso eine HTTPS-Verbindung für einen Online-Shop auch dann obligatorisch ist, wenn nach der Authentifizierung des Benutzers keine kritischen Daten, wie z.B. Passwörter u.ä. übertragen mehr übertragen werden. (2 Punkte)

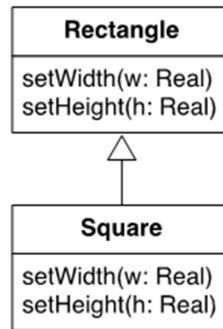
b) Erklären Sie stichwortartig, wie Sie im folgenden EJB-Code die Ausführung von Schadcode verhindern würden (engl. sanitize), ohne query.setParameter zu verwenden. (2 Punkte)

`Query query = em.createQuery("select r from Restaurant r where r.name = '" + name + "'");`

c) Erklären Sie kurz den Begriff Refactoring und die Bedeutung von Testfällen beim Refactoring. (2 Punkte)

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

3. d) Erklären Sie stichwortartig das Substitution Principle nach Liskov (LSP) und auch, was es für Vor- und Nachbedingungen (Precondition & Postcondition) von Operationen bedeutet. Begründen Sie ferner, ob das LSP für das folgende Vererbungsbeispiel erfüllt ist. (4 Punkte)

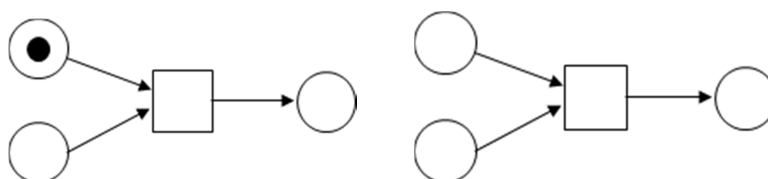
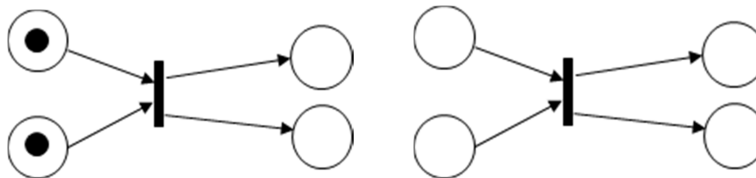


**Aufgabe 4: Echtzeit-Systeme (16 Punkte)**

a) Skizzieren und erklären Sie kurz die Komponenten des Watchdog-Patterns und seine Funktionsweise. (4 Punkte)

Hinweis: Es ist nicht notwendig, das Zustandsdiagramm des Patterns anzugeben.

b) Im Folgenden sehen Sie drei einfache Petri-Netze. Geben Sie jeweils auf der rechten Seite an, wie das entsprechende Petri-Netz von links nach dem nächsten Taktschritt aussehen wird. (3 Punkte)





Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

4. c + d) Es seien die folgenden vier Prozesse gegeben:

	ArrivalTime	Duration	Deadline
<b>P1</b>	<b>1</b>	<b>2</b>	<b>4</b>
<b>P2</b>	<b>3</b>	<b>3</b>	<b>9</b>
<b>P3</b>	<b>3</b>	<b>4</b>	<b>11</b>
<b>P4</b>	<b>4</b>	<b>4</b>	<b>12</b>

Hinweise: Die Prozesse kommen zu Beginn des in ArrivalTime angegebenen Zeitslots an und sollten bis zum Ende des in Deadline angegebenen Zeitslots beendet sein. Sollte dies nicht der Fall sein, führen Sie das Scheduling bitte trotzdem zu Ende.

c) Ordnen Sie den Prozessen die Prozessorzeit nach dem Earliest-Deadline-First-Verfahren zu. (3 Punkte)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<b>P1</b>														
<b>P2</b>														
<b>P3</b>														
<b>P4</b>														

d) Teilen Sie nun die Prozessorzeit bitte nach dem Least-Laxity-Verfahren ein. (6 Punkte)

Hinweis: Die grau hinterlegten Zellen können Sie zum Notieren der jeweiligen Laxity nutzen, zum Erreichen der vollen Punktzahl ist dies aber nicht erforderlich.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<b>P1</b>														
Lax.														
<b>P2</b>														
Lax.														
<b>P3</b>														
Lax.														
<b>P4</b>														
Lax.														

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

**Aufgabe 5: Usability und UI-Design (11 Punkte)**

a) Nennen Sie vier der in der Vorlesung vorgestellten „Gestalt-Laws“ und erläutern Sie ihre Bedeutung stichwortartig. (4 Punkte)

b) Geben Sie zu den im Folgenden genannten vier verschiedenen UI-Elementtypen jeweils an, welche Elemente aus den in der Vorlesung vorgestellten Analyse- und Design-Modellen als Grundlage dafür verwendet werden können. (4 Punkte)

*Buttons:* \_\_\_\_\_

*Bildschirme/Fenster:* \_\_\_\_\_

*Eingabefelder:* \_\_\_\_\_

*Beschriftungen (Labels):* \_\_\_\_\_

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

5. c) Analysieren Sie die folgende UI zum Ändern der PIN eines Online-Kontos und nennen Sie mindestens drei Fehler bzw. Ungereimtheiten, die ihre Benutzbarkeit beeinträchtigen. (3 Punkte)

**Your IPIN should**

- ▶ Be a minimum of 8 characters and a maximum of 16
- ▶ Be alphanumeric – combination of alphabets and numbers
- ▶ Not contain blank spaces
- ▶ Not contain 2 similar characters consecutively
- ▶ Not contain special characters like ", <, >, /, \, & and %
- ▶ Not be the same as one of your last 6 IPiNs

Example of Correct PIN : abc12345

Example of Incorrect PIN : ab<3&%5



Enter IPIN of your choice using Virtual Keyboard ([Why Virtual Keyboard?](#))

.....

Re-enter selected IPIN

.....

**Aufgabe 6: Zuverlässigkeit (14 Punkte)**

a) Die Zuverlässigkeit von technischen Systemen kann durch Redundanz relativ einfach erhöht werden. Erklären Sie, welche Unterschiede es dabei zwischen Hardware- und Software-Redundanz zu beachten gilt und wie Software-Redundanz evtl. sinnvoll umgesetzt werden kann. (4 Punkte)

**Hinweis:** Hardware bezeichnet hier nicht alleine Computer-Hardware, sondern auch komplexe Systemteile, wie bspw. Flugzeugtriebwerke.

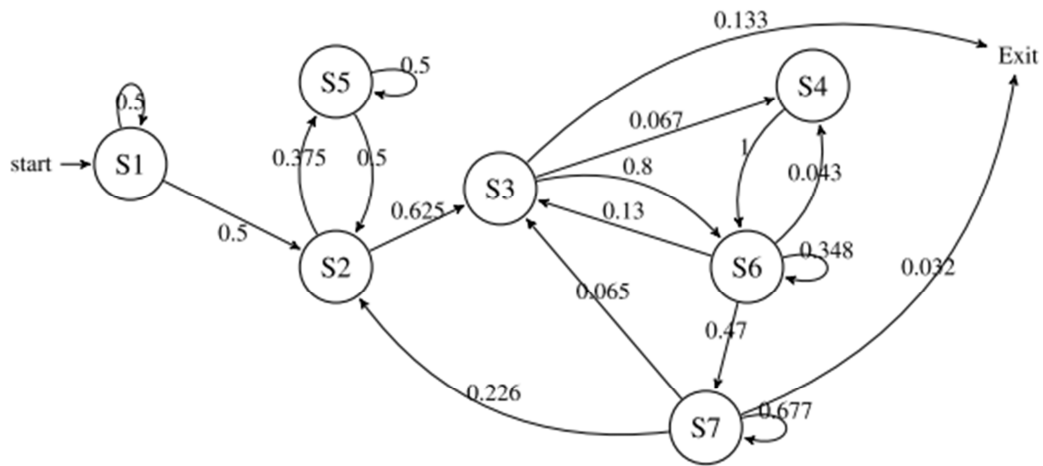
b) Berechnen Sie die Anzahl der notwendigen statistischen Testfälle, um eine Zuverlässigkeit von 99,9 Prozent bei einer maximalen Irrtumswahrscheinlichkeit von 5 Prozent annehmen zu können. (5 Punkte)

Hinweis: Eine Gleichung der Form  $\log a^x = \log b$  lässt sich zu  $x \log a = \log b$  umformen. Bitte geben Sie alle verwendeten Formeln und Zwischenschritte an, da auch diese bepunktet werden.

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

6. c) Erstellen Sie eine rekurrente Übergangsmatrix  $T$  (State Transition Matrix) für die folgende Markov-Kette. (5 Punkte)

Hinweis: S1 soll dazu dem aus der Vorlesung bekannten „Enter“-Knoten entsprechen.



Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

**Aufgabe 7: Software-Projektmanagement (11 Punkte)**

a) Skizzieren Sie den in der Vorlesung vorgestellten „Cone of Uncertainty“ und erklären Sie ihn kurz. (4 Punkte)

b) Nennen und erklären Sie stichwortartig die drei verschiedenen Arten von Aufgaben (Tasks), die nach Brooks' „Mythical Man Month“ in Softwareprojekten vorkommen können und geben Sie zu jedem ein passendes Beispiel mit Bezug zu Softwareprojekten. (4 Punkte)

Name: \_\_\_\_\_ Matrikelnummer: \_\_\_\_\_

c) Beschreiben Sie kurz den Inhalt von Brooks bekanntem Statement „No Silver Bullet“ und geben Sie anschließend eine kurze Begründung dafür an. (4 Punkte)