

Klausur

Softwaretechnik

Prüfer: Prof. Dr. Malte Weiß

Vorname	Nachname	Matrikelnummer
---------	----------	----------------

Bewertung:

1	2	3	4	5	6	Gesamt	Note
12	20	20	15	18	12	100	

Zum Ablauf der Prüfung

- **Zeit für die Lösung:** 4 Stunden
- **Beginn:** 8:00 Uhr, **Ende:** 12:00 Uhr am 17.03.2021
- **Lösung wo hochladen:** in E-Learning im [Kursraum der Vorlesung](#)
- **Lösung wie hochladen (Form):** Zip-Datei im Format
swt-exam-VORNAME-NACHNAME-MATRIKELNUMMER.zip.
Beispiel: Student *Mathias Bauer* mit Matrikelnummer *10001234* gibt eine Zip-Datei *swt-exam-Mathias-Bauer-10001234.zip* ab.
- **Ordnerstruktur:** Jede Aufgabe liegt in einem entsprechenden Unterordner in der Zip-Datei. Siehe aufgabenspezifische Konventionen.
- **Nicht erlaubte Hilfsmittel/Hilfe:** Kommunikation digital oder persönlich, direkt oder indirekt mit anderen Menschen (**keine** SMS, WWW-Kommunikation, Telefon, E-Mail, Chat, etc.)
- **Kompilierbarkeit:** Lösungen *müssen* kompilieren, ansonsten werden 50 % der Aufgabenpunkte abgezogen.
- **Fragen:** Für Fragen während der Klausurzeit nutzen Sie bitte das Forum im E-Learning-Bereich der Veranstaltung. Preisgabe von Lösungen oder Ansätzen führt zu Nichtbestehen der Veranstaltung inkl. Fehlversuch.
- **Eidesstattliche Erklärung:** Die Klausur gilt als **nicht angetreten**, wenn Sie bei Beginn der Klausur nicht die vorgeschriebene eidesstattliche Versicherung unterschrieben haben.

Aufgabe 1: Anforderungsanalyse (12 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A1 gespeichert werden.

Stellen Sie sich das folgende Produkt vor:

Der *Milkynator™* ist ein Haushaltsgerät, das es erlaubt, auf Knopfdruck Babymilch aus Milchpulver herzustellen. Einfach Fläschchen unter der Düse platzieren und Knopf drücken. Schon wird die Milch in die Flasche gefüllt und kann direkt verfüttert werden.

Nennen Sie die 3 **wichtigsten** funktionalen (FA) und die 3 **wichtigsten** nicht-funktionalen Anforderungen (NFA). Beachten Sie folgendes:

- Es geht um die wichtigsten Anforderungen. Generische Anforderungen (z.B. "Das Gerät muss die Farben der Corporate Identity verwenden") geben 0 Punkte. Die Anforderungen sollten sich speziell auf das Produkt beziehen.
- Für die nicht-funktionalen Anforderungen sollten Sie die Anforderungen von Babymilch recherchieren.
- Achten Sie auf eine qualitativ hochwertige Formulierung der Anforderungen, wie Sie es in der Vorlesung gelernt haben.

Aufgabe 2: Use-Case Diagramm (20 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A2 gespeichert werden.

Lesen Sie aufmerksam das folgende Szenario:

"Ein Gast möchte zu Mittag essen. Hierfür geht er zu seinem Döner-Laden des Vertrauens. Der Koch weist darauf hin, dass die Zubereitung der Dönertasche Zeit benötigt. In der Zwischenzeit geht der Gast zur Bank, um Bargeld für den Bezahlvorgang der Dönertasche abzuheben.

Nach Fertigstellung des Döners bezahlt der Kunde. Wenn der Koch den Döner in unter 5 Minuten zubereitet hat, gibt ihm der Gast zusätzlich ein Trinkgeld.

In letzter Zeit haben sich Vorfälle gemehrt, in denen *sehr hungrige* Gäste hinter die Theke des Döner-Ladens gesprungen sind, um sich selbst eine Dönertasche herzustellen. Dann erscheint aber ein Alarmton und der Koch wirft den Gast aus dem Laden."

Erstellen Sie für das obige Szenario ein Use-Case Diagramm. Extrahieren Sie dafür alle Akteure und Use-Cases. Denken Sie auch daran, bei Bedarf include- und extend-Beziehungen zu verwenden.

Aufgabe 3: Aktivitätsdiagramm (20 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A3 gespeichert werden.

Lesen Sie folgendes Kochrezept für ein Nudelgericht:

"Setzen Sie zunächst Wasser auf und kochen Sie 250g Nudeln.

Während die Nudeln kochen, braten Sie zunächst 3 Frühlingszwiebeln und 2 Knoblauchzehen mit 2 Teelöffeln Butter an. Anschließend braten Sie 400 g Rindfleisch an. Sobald das Rindfleisch durchgebraten ist, werden 2 Esslöffel Zucker hinzugefügt. Idealerweise brauner Zucker – steht dieser nicht zur Verfügung, wird normaler Zucker genutzt. Am Ende werden 2 Esslöffel Sojasoße hinzugefügt.

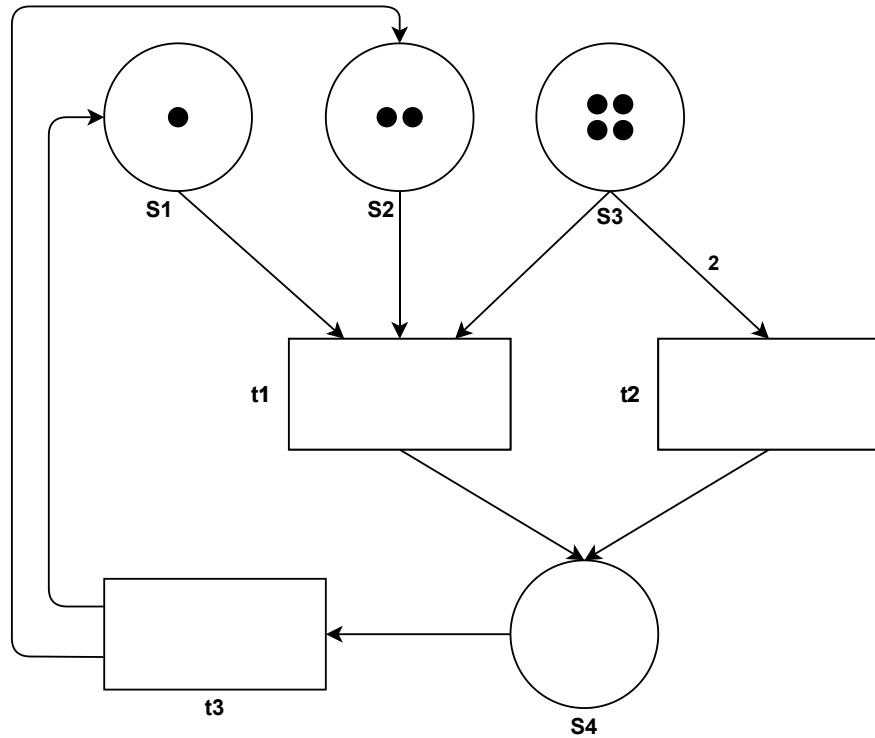
Sind die Nudeln *al dente* und das Fleischgericht fertig, wird beides zusammengemischt und abgeschmeckt. Schmeckt das Gericht furchtbar, wird es unter gleichmäßigem Rühren in den Abfalleimer gekippt. Ansonsten wird es serviert."

Erstellen Sie ein Aktivitätsdiagramm, das die Aktivität des Kochens repräsentiert.

Aufgabe 4: Petri-Netze (15 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A4 gespeichert werden.

- a) Erstellen Sie einen Erreichbarkeitsgraphen für das abgebildete Petri-Netz.



(12 Punkte)

- b) Ist das Petri-Netz aus Aufgabenteil a) stark lebendig? Begründen Sie Ihre Antwort.
(3 Punkte)

Aufgabe 5: Pattern implementieren (18 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A5 gespeichert werden.

Betrachten Sie die beigefügte Java-Klasse **Book** im Package **de.hrw.st.exam**.

- a) Implementieren Sie für die Klasse das **Builder**-Pattern in Form einer Klasse **BookBuilder**. Schreiben Sie ein Hauptprogramm, das das Pattern demonstriert.
(6 Punkte)

- b) Implementieren Sie für die Klasse das **Observer**-Pattern. Es soll ermöglichen, sich über Änderungen an Attributen von Bücher-Objekten benachrichtigen zu lassen. Schreiben Sie ein Hauptprogramm, das das Pattern demonstriert.
(12 Punkte)

Aufgabe 6: Unit-Tests (12 Punkte)

Die Lösung der Aufgabe muss im ZIP-Archiv im Ordner A6 gespeichert werden.

- a) Implementieren Sie einen Unit-Test für die Hilfsmethode **getDigitSum** in der Klasse **MathAnalyser**. Der Unit-Test sollte Anweisungsüberdeckung erreichen.
(5 Punkte)

- b) Implementieren Sie einen Unit-Test für die Hilfsmethode **getWordCount** in der Klasse **TextAnalyser**. Der Unit-Test sollte Anweisungsüberdeckung erreichen.
(7 Punkte)