
KLAUSUR
ZUR VORLESUNG
“SOFTWARE ENGINEERING”
SOMMERSEMESTER 2005
26.7.2005
PROF. DR. RALF MÖLLER

Name: _____

Matrikelnummer: _____

Studiengang: _____

Unterschrift: _____

Bitte beachten:

1. **Klausur nicht öffnen bis zur Erlaubniserteilung durch die Klausuraufsicht!**
2. **Bitte Personalausweis als auch Studentenausweis auf den Tisch legen.**
3. **Prüfen Sie, ob Ihre Matrikelnummer auf der ausgehändigten Liste aufgeführt ist. Wenn nicht, so**
 - (a) **Füllen Sie einen Vorbehalt (Proviso) aus**, der Ihnen von der Klausuraufsicht ausgehändigt wird. **Sie dürfen die Klausur erst beginnen, nachdem Sie den Vorbehalt ausgefüllt und abgegeben haben.**
 - (b) **Füllen Sie auch den “Formzettel zur Nachmeldung” aus.** Lassen Sie das Formular vom Studiensekretariat abstempeln und unterzeichnen und bringen Sie diesen Zettel *persönlich* in das STS-Sekretariat (Harburger Schloßstr. 20, 2nd floor, Frau Hantschmann); bringen Sie auch ihren Personalausweis mit.
4. **Bearbeitungszeit: 90 Minuten. Zusätzliche Hilfsmittel sind nicht erlaubt.**
5. Das Symbol “☺” gibt Hinweise zur Bearbeitungszeit.
6. Es befindet sich ausreichend Platz für Ihre Antworten auf den Aufgabenzetteln.
7. Wenn Sie *zusätzliches Papier* von der Klausuraufsicht erhalten, schreiben Sie auch Ihren Namen, Matrikelnummer sowie eine Seitennummer auf die Zettel.
8. Wenn Sie den Prüfungsraum verlassen müssen, **so melden Sie sich geräuschlos und informieren Sie die Klausureinsicht. Verlassen Sie Ihren Platz nicht unaufgefordert.**

1. In vielen Anwendungsprojekten stehen schon aus vorherigen Projekten Lizenzen für bewährte Altsysteme zur Verfügung. In dieser Situation sollte sich doch eigentlich eine starke Vereinfachung für die Erstellung von Software in einem Nachfolgeprojekt ergeben.
Nennen Sie drei Argumente, die belegen, daß diese nicht der Fall ist.

⌚
6
P
6

2. Belegen Sie, warum im Einsatz befindliche Software einer ständigen “Wartung” bedarf, wo doch eigentlich Software nicht “altert”. Nennen Sie drei Argumente.

⌚
6
P
6

3. Nennen Sie drei wesentliche Prinzipien des Extreme Programming.

⌚
3
P
3

4. (a) Erläutern Sie die zentralen Ideen des V-Modelles.

⌚
8
P
8

- (b) Grenzen Sie Validation und Verifikation voneinander ab.

5. (a) Erläutern Sie den fundamentalen Unterschied der Function-Point-basierten Aufwandsschätzmethode zu der LOC-basierten Aufwandsschätzmethode nach Boehm.

⌚
10
P
10

- (b) Schildern Sie zwei wesentliche Nachteile der Function-Point-Methode.

6. (UML-Strukturdiagramme) Erläutern Sie, warum es für das Software Engineering bedeutsam ist, in einem UML-Klassendiagramm bzgl. der Attribute, die für die Instanzen einer Klasse definiert sein sollen, die Wertebereiche anzugeben (also die "Typen" der Attribute zu spezifizieren). Ergibt sich durch Angabe der Typen nicht eigentlich eine unnötige Inflexibilität?

⌚
2
P
2

7. (a) Erläutern Sie den Zusammenhang von UML-Use-Case-Diagrammen und dem Lasten- bzw. Pflichtenheft.

⌚
10
P
10

- (b) Stellen Sie dar, in welcher Weise Use-Case-Diagramme für das Vorgehen im Software Engineering bedeutsam sind.

8. Für die Erstellung einer Software zur Unterstützung der Kundenbetreuung nehmen wir an, es stehen folgende Fragmente im Pflichtenheft:

Ein Kundendienstmitarbeiter bearbeitet ein Problem bei einem Kunden. Jeder Kundendienstmitarbeiter ist für die Betreuung von mindestens fünf bis maximal zehn Kunden zuständig. Ein Kundendienstmitarbeiter führt eine Operation "Reparatur" bei einem Kunden aus. Je nach Art des vom Kunden genannten Problems, kann die Reparatur unterschiedliche Form annehmen. Weiterhin kann sich ein Problem in verschiedenen Teilprobleme zerlegen.

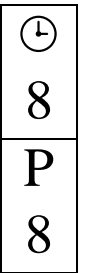
Erstellen Sie ein UML-Klassendiagramm, in dem die in dem obigen Abschnitt skizzierten Zusammenhänge erfasst werden.

⌚
10
P
10

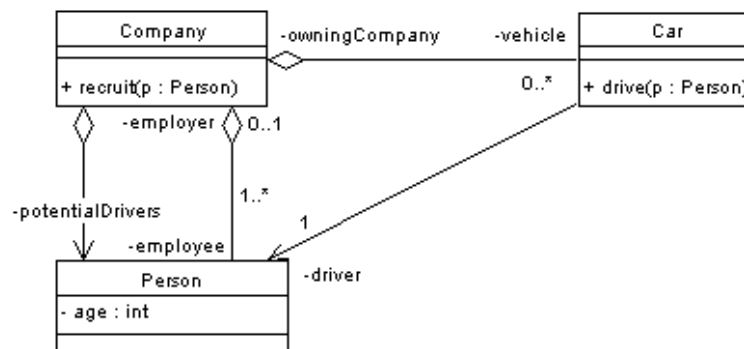
9. Für eine Warenhauskette soll in einem neuen Projekt eine Individualsoftware für den Verkauf erstellt werden. Im Pflichtenheft ist die Interaktion zwischen dem Verkauf und dem elektronischen Warenabrechnungssystem skizziert. Erstellen Sie aus dem nachfolgendem Szenario ein UML-Sequenzdiagramm.

Der Kunde übergibt dem Verkäufer einen Warenkorb mit Waren, die abgerechnet werden sollen. Der Verkäufer startet den Bezahlprozess. Er gibt die Artikelnummern der ersten Ware in das Warenabrechnungssystem ein. Das System gibt daraufhin die Warenbezeichnung, den Preis und die Zwischensumme an den aktuellen Rechnungsbezahlprozess aus. Dieser Vorgang wird für alle Artikel wiederholt. Am Ende wird der Prozess durch den Verkäufer beendet. Anschließend teilt das Warenwirtschaftssystem dem Verkäufer Gesamtsumme und MWSt-Anteil mit. Der Verkäufer gibt diese Information an den Kunden weiter, der anschließend bezahlt. Der Verkäufer vermerkt die Bezahlung im Warenwirtschaftssystem, woraufhin ein Beleg erstellt und dem Kunden ausgehändigt wird.

Erstellen Sie ein UML-Sequenzdiagramm, dass diese Szenario bestmöglich wiedergibt.



10. Erläutern Sie die Einschränkungen, die sich durch die OCL-Spezifikationen in folgendem Klassendiagramm ergeben.



⌚
4
P
4

Car

inv: owningCompany = driver.employer
 inv: driver.age >= 18

Company

inv: potentialDrivers = employees->select(age >= 18)
 inv: vehicles->forAll(driver.age >= 18)

Company::recruit(p : Person)

pre : -- none
 post: (employees.size = employees @pre.size+1) &&
 (employees.includes(p))

Car::drive(p : Person)

pre : (driver == null) && (p.age >= 18)
 post: driver = p

11. Bahnübergang

Ein Bahnübergang ist eine niveaugleiche Kreuzung zwischen einer Strasse und zwei Gleisen (vereinfachte Annahme). Züge können nur rechts fahren. Jedes Gleis hat in Fahrtrichtung vor dem Bahnübergang einen Sensor, der das Nähern eines Zuges signalisiert (siehe Bild). Jedes Gleis hat in Fahrtrichtung hinter dem Bahnübergang einen Sensor, der signalisiert, dass ein Zug den kritischen Bereich vollständig verlassen hat.

Auf jeder Spur der Strasse befindet sich in Fahrtrichtung vor der Schranke eine Ampel. Immer wenn sich mindestens ein Zug im kritischen Bereich aufhält, muss die Ampel rot leuchten. Sobald die Schranken geöffnet wurden, erlöschen die Ampeln.

Zeichnen Sie das Zustandsdiagramm (auch Statechart genannt) für eine Ampel.

