
Koziolk
Softwaretechnik 2

Dauer: 90 min.

Lösung: keine

Bestanden mit: ? P.

Bemerkung: Hauptklausur

Aufgabe 1: Verschiedenes (16 Punkte)

- Nennen Sie die vier Phasen des *Unified Process (UP)* und beschreiben Sie diese stichpunktartig. (6 Punkte)
- Ordnen Sie die folgende Produkthanforderung gemäß der Facetten *Repräsentation* und *zu Grunde liegendes Anliegen (kind)* ein: „Das System soll auf einer 64-Bit Prozessorarchitektur lauffähig sein, die den x86-Befehlssatz implementiert“. (1 Punkt)
- Welche Fehler finden Sie in den nachfolgend definierten Software-Anforderungen? Benennen Sie jeweils einen Verstoß gegen eine der Regeln für das Schreiben von natürlichsprachigen Anforderungen und schreiben Sie die Anforderung in korrigierter Form auf. (3 Punkte)
 - „Nachdem Nutzernamen und Passwort validiert wurden, soll der Nutzer zu seinem Konto weitergeleitet werden.“
 - „Das System soll dem Nutzer innerhalb von 100ms die Antwort in Form einer Tabelle präsentieren.“
- Beschreiben Sie kurz den Unterschied zwischen Verifikation und Validierung. (2 Punkte)
- Beschreiben Sie drei Formen des Testens, zum Beispiel aus der Test-Pyramide („Testing Pyramid“). Nennen und begründen Sie außerdem kurz, welche dieser Formen sich der *Continuous Integration (CI)* zuordnen lassen. (2.5 Punkte)
- Nennen Sie drei Gründe für das Ersetzen von *Depended-on Components (DOC)* bei Tests. (1.5 Punkte)

Aufgabe 2: Use-Cases (15 Punkte)

Folgende Beschreibung charakterisiert die zugrundeliegende Funktionalität eines Online-Blumenversands:

In einem Online-Blumenversand kann ein Kunde zunächst die verschiedenen Artikel betrachten. Bevor der Kunde Artikel in den Warenkorb legen kann, muss er einen Blumenhändler auswählen. Dabei gibt der Kunde zunächst die gewünschte Lieferadresse an. Dann muss der Kunde aus einer Liste an Blumenhändlern einen präferierten Händler auswählen. Die Liste wird generiert, indem alle Blumenhändler ermittelt werden, bei denen die Lieferadresse des Kunden im Liefergebiet liegt. Daraufhin kann der Kunde die Artikel, die ihm gefallen und die er kaufen möchte, in den Warenkorb legen. Dazu wählt der Kunde die Stückzahl und die gewünschte Variante des Artikels aus und legt den Artikel dann in den Warenkorb. Sollte der Artikel nicht beim ausgewählten Händler verfügbar sein, wird der Kunde darauf hingewiesen. Der Kunde kann dann für diesen Artikel einen alternativen Händler auswählen, der den Artikel führt und der ebenfalls an die Adresse des Kunden liefert. Nachdem der Kunde alle gewünschten Artikel dem Warenkorb hinzugefügt hat, schließt er, die Bestellung ab. Dafür muss der Kunde die Lieferadresse bestätigen und gegebenenfalls eine abweichende Rechnungsadresse hinzufügen, bevor er die Zahlung abwickeln kann. Bei der Zahlung kann er aus den verschiedenen Bezahlmethoden (Vorkasse, Lastschrift oder Kreditkarte) auswählen und wird dann entsprechend der ausgewählten Bezahlmethode weitergeleitet. Nach dem Bezahlen ist der Bestellvorgang abgeschlossen und der Kunde erhält noch eine Übersicht zur Bestellung.

- a) In der Vorlesung haben Sie Shaping-Heuristiken zur Identifikation guter Anwendungsfälle kennengelernt. Benennen und beschreiben Sie diese kurz und geben Sie jeweils ein Beispiel aus dem Beschreibungstext an, der den Test nicht besteht. Begründen Sie dies anhand der Shaping-Heuristiken. (4.5 Punkte)
- b) Verwenden Sie das textuelle Anwendungsfallbeschreibungsschema *Fully-Dressed* zur vollständigen Spezifikation des Anwendungsfalls „Blumenstrauß bestellen“. Nutzen Sie hierfür die unten gegebene Schablone. Geben Sie auch mindestens eine Erweiterung an. (7.5 Punkte)

Name d. Anwendungsfalls (<i>Use Case Name</i>)	
Geltungsbereich (<i>Scope</i>)	
Primäraktor(en) (<i>Primary Actor(s)</i>)	
Stakeholder(s)	
Vorbedingungen (<i>Preconditions</i>)	
Nachbedingung (<i>Postconditions</i>)	
Primäres Erfolgsszenario (<i>Main Success Scenario</i>)	
Erweiterung (Alternativer Ablauf) (<i>Extension; Alternative Flow</i>)	

- c) Nennen Sie die vier *Use Case Goal Levels* nach Cockburn. Ordnen Sie den Anwendungsfall aus Teilaufgabe b) einer der Stufen zu. Begründen Sie dabei kurz ihre Entscheidung. (3 Punkte)

Aufgabe 3: Modellierung (16 Punkte)

- Sie möchten zur Entwurfszeit für Ihre komponentenbasierte Softwarearchitektur Vorhersagen zu Qualitätseigenschaften machen. Was sind die vier Einflussfaktoren auf die Performanz einer Komponente, wie Sie sie im Zusammenhang mit dem *Palladio Component Model (PCM)* kennengelernt haben? (2 Punkte)
- Erstellen Sie das Klassendiagramm eines *Metamodells* für die Domäne des im Folgenden beschriebenen Online-Blumenversands. Berücksichtigen Sie Modellklassen, Beziehungen, Rollen, sowie Multiplizitäten. (8 Punkte)

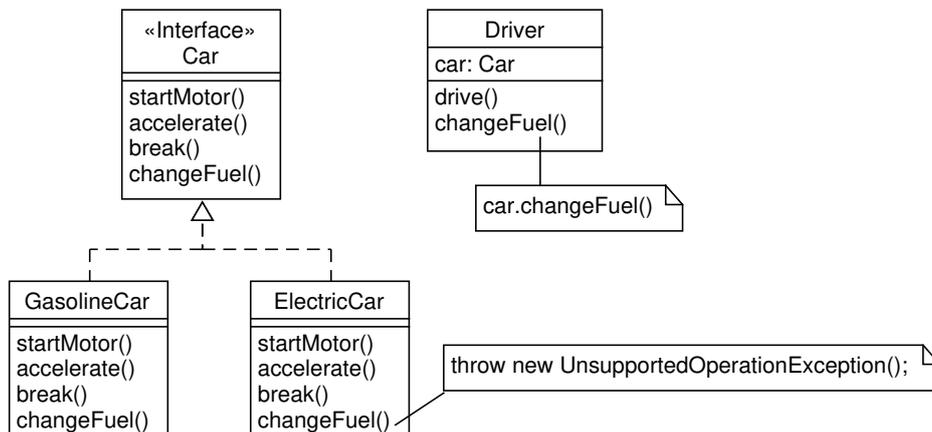
Beim Online-Blumenversand gibt es zwei Arten von Konten. Zum einen Geschäftskonten und zum anderen Kundenkonten. Bei Kundenkonten werden Name und Adresse hinterlegt, während Geschäftskonten neben dem Namen noch ein Liefergebiet haben. Die Konten beinhalten außerdem jeweils eine Zahlungsadresse und ein Anmeldedatum. Ein Konto kann genutzt werden, um Artikel in den aktuell beim Konto hinterlegten Warenkorb zu legen. Im Warenkorb wird neben den Artikeln auch stets der aktuelle Gesamtbetrag gespeichert. Die angebotenen Artikel haben einen Preis und können in drei Arten unterschieden werden: Gartenpflanzen mit einer Pflanzzeit, Blumensträuße mit einer Anzahl an enthaltenen Blumen und Pflanzbedarf mit einer Artikelnummer. Wenn der Kunde mit seinen Artikeln im Warenkorb zufrieden ist, gibt er die Bestellung auf. Bestellungen werden dabei immer zwei Konten zugeordnet, dem Konto des Kunden und zusätzlich dem Konto eines ausgewählten Geschäfts mit passendem Liefergebiet. Bei der Bestellung wird neben dem Verweis auf den Warenkorb auch stets das Bestelldatum gespeichert. Über eine Erneut-Bestellen-Funktion können Kunden eine vergangene Bestellung erneut ordern. Dafür wird eine neue Bestellung angelegt, die auf den selben Warenkorb wie die erneut gewünschte Bestellung verweist.

Hinweis: Datentypen (wie int, long, Date,...) müssen **nicht** angegeben werden.

- Warum ist Ihre Lösung von Teilaufgabe b) ein Modell? Benutzen Sie die Definition eines Modells von Stachowiak und setzen Sie diese in Bezug zu den Attributen Ihrer Lösung. (3 Punkte)
- Ihr Vorgesetzter möchte, dass Sie sich entscheiden, ob als Enterprise Application Pattern die *class table inheritance* oder die *concrete table inheritance* für die Persistierung der Artikel aus Teilaufgabe b) eingesetzt werden soll. Sie wissen, dass Performanz eine wichtige Qualitätsanforderung darstellt. Außerdem soll in der nächsten Iteration ein Refactoring stattfinden. Beschreiben sie zunächst kurz den Unterschied zwischen den beiden Mustern und diskutieren Sie dann kurz, welches der beiden Muster in diesem Fall besser geeignet ist. (3 Punkte)

Aufgabe 4: Software-Design & Refactoring (11 Punkte)

- Welche Verstöße gegen die aus der Vorlesung bekannten Prinzipien von gutem OO-Entwurf sind im folgenden Beispiel enthalten? Nennen Sie zwei der Prinzipien, gegen die ein Verstoß vorliegt, und erklären Sie diese Prinzipien stichwortartig anhand des Beispiels. Wie können diese Verstöße jeweils behoben werden? (8 Punkte)



- Nennen Sie drei Einschränkungen, die gegen ein Refactoring sprechen könnten. (3 Punkte)

Aufgabe 5: Scheduling (9 Punkte)

Es seien die folgenden vier Prozesse gegeben:

	Arrival Time	Duration	Deadline
P1	2	3	6
P2	1	3	11
P3	4	4	10
P4	6	2	9

Hinweis: Die Prozesse kommen zu Beginn des in *Arrival Time* angegebenen Zeitslots an und sollen bis zum Ende des in *Deadline* angegebenen Zeitslots beendet sein. Führen Sie das Scheduling auch dann fort, wenn die Deadline nicht eingehalten werden kann.

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

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

- b) Teilen Sie nun die Prozessorzeit nach dem prä-emptiven Least-Laxity-Scheduling-Verfahren ein. (6 Punkte)

Hinweise: Bei gleicher Laxity verteilen Sie die Prozessorzeit so, dass möglichst jeder Prozess zum Zuge kommt (Round-Robin). Die Zeilen *Lax* sind zum Notieren der Laxity vorgesehen.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
P1														
Lax														
P2														
Lax														
P3														
Lax														
P4														
Lax														

Aufgabe 6: Cloud Computing & Architecture (12 Punkte)

- Beschreiben Sie das Problem der Bereitstellung von Ressourcen und wie Cloud Computing dieses Problem zu lösen versucht. Vergleichen Sie dazu auch die Kosten von traditionellen Lösungen mit den Kosten beim Einsatz von Cloud Computing. (3 Punkte)
- Was ist die grundlegende Idee des MapReduce-Programmiermodells? Erläutern Sie diese Idee anhand der Wortzählfunktionalität aus der Vorlesung, bei der die Vorkommen der verschiedenen Worte in einem Text gezählt werden. (3 Punkte)
- Nennen Sie die aus der Vorlesung bekannten drei Modelle der Dienstleistungserbringung (*Service Delivery Models*) nach NIST und geben Sie jeweils ein Beispiel für das jeweilige Modell an. (3 Punkte)
- Nennen Sie drei Architekturprinzipien von Cloud Architektur aus der Vorlesung, z.B. aus den *Amazon S3 Design Principles*, und beschreiben Sie diese jeweils kurz. (3 Punkte)

Aufgabe 7: Sicherheit (11 Punkte)

- a) Warum sollte Software-Sicherheit in Betracht gezogen werden? (2.5 Punkte)
- b) Für Ihre Webapplikation möchten Sie eine Login-Funktionalität bereitstellen und wollen dafür die Benutzeranmeldeinformationen bestehend aus Nutzernamen und Passwort speichern. Nennen Sie zwei Sicherheitsmechanismen, um das Anmeldesystem abzusichern und die Anmeldedaten zu schützen. (2 Punkte)
- c) Sie haben bei einer Sicherheitsevaluation ihrer Software festgestellt, dass ihre Applikation anfällig für *SQL Injections* ist. Was sind *SQL Injections* und wie könnte man dieses Problem beheben? (2 Punkte)
- d) Welches potentielles Sicherheitsproblem ist in folgendem Code enthalten? Beschreiben Sie außerdem, welche Folgen das Sicherheitsproblem hat. (2 Punkte)

```
#include<stdio.h>
#include <string.h>

int main(int argc, char const *argv[]) {
    int var = 0;
    char secret[100];
    printf("Please insert your secret to be stored securely: ");
    gets(secret);
    printf("Secret stored, never to be seen again!\n");

    if (var != 0) {
        printf("Your secret: %s", secret);
    }

    return 0;
}
```

- e) Beschreiben Sie kurz das Problem von Pseudo-Zufälligkeit und erläutern Sie kurz eine Möglichkeit, dieses Problem zu minimieren. (2.5 Punkte)