

24. August 2018

Klausur

Professionelle Softwareentwicklung
Sommersemester 2018

Nachname: _____ Vorname: _____

Matrikelnummer: _____

Unterschrift:

Zugelassene Hilfsmittel: Eine handschriftlich beschriebene A4 Seite. Matrikelnummer und Name muss auf der Seite stehen

Sollten Sie zusätzliche Blätter benötigen, bekommen Sie diese von uns. Sie dürfen keine eigenen Blätter verwenden!

Diese Klausur enthält 6 nummerierte Seiten. Prüfen Sie bitte zuerst, ob alle Seiten vorhanden sind.

Diesen Teil bitte nicht ausfüllen:

[illegible]

Aufgabe 1

[10 Punkte]

Geben Sie ein Beispiel für einen Fall, in dem das Liskov'sche Substitutionsprinzip (LSP) verletzt ist. Erläutern Sie, an welcher Stelle und warum das LSP verletzt wird. Sie dürfen Code schreiben, müssen das aber nicht.

Aufgabe 2

[10 Punkte]

Angenommen wir haben eine Klasse ‘User’ und eine Methode ‘login’. Auf welche Bestandteile darf die Methode zugreifen, wenn das Gesetz von Demeter eingehalten werden soll? Schreiben Sie ja neben die Aussage, wenn diese korrekt ist oder nein, falls die Aussage nicht korrekt ist.

- A. Auf Methoden in ‘User’
- B. Auf Methoden aller Subklassen von ‘User’
- C. Auf alle Attribute von ‘User’, die ‘private’ sind
- D. Auf alle Attribute von ‘User’, die ‘public’ sind

Bewertung: Volle Punktzahl erhalten Sie, wenn alle Aussage korrekt beantwortet wurden. Bei einem Fehler bekommen Sie die Hälfte der Punkte. Bei zwei oder mehr Fehlern bekommen Sie keine Punkte. Nicht beantwortete Aussagen werden als Fehler gewertet.

Aufgabe 3

[10 Punkte]

Folgende Gradle Datei bricht beim Aufruf von `gradle build` mit einem Compiler-Fehler ab, was ist die Ursache und wie kann das Problem repariert werden?

```
apply plugin: 'java'
```

```
repositories {  
    jcenter()  
}
```

Ein Ausschnitt aus der Fehlermeldung ist

```
Putzplan.java:5: error: package org.apache.pdfbox.pdmodel does not exist  
import org.apache.pdfbox.pdmodel.PDDocument;  
                                ^
```

Aufgabe 4

[10 Punkte]

Welche Konsequenzen entstehen aus Verletzungen des Single Responsibility Prinzips?

Aufgabe 5

[10 Punkte]

Warum sollten Tests nach Möglichkeit unabhängig voneinander sein?

Aufgabe 6

[10 Punkte]

Angenommen die Klasse zur Kommunikation mit der Kundendatenbank ‘DerbyCustomerRepository’ unserer Marketinganwendung würden wir von einem Drittanbieter beziehen. Wir könnten diese dann natürlich nicht einfach so umschreiben, dass sie ‘CustomerRepository’ implementiert und auch der Drittanbieter wird das nicht tun. Nennen und erklären Sie zwei **unterschiedliche** Strategien um die ‘doMarketing’ Methode trotzdem zu testen.

Aufgabe 7

[10 Punkte]

Das Interface Segregation Prinzip besagt, dass Interfaces möglichst klein sein sollen. Warum ist es nicht generell sinnvoll grundsätzlich nur Interfaces mit genau einer Methode zu schreiben?

Aufgabe 8

[10 Punkte]

Was bedeutet es, wenn eine Methode idempotent ist? Geben Sie ein Beispiel für eine idempotente Methode, die einen Seiteneffekt hat. Sie können eine eigene (sinnvolle) Methode schreiben oder eine Methode aus der Java Standardbibliothek angeben.

Aufgabe 9

[10 Punkte]

Was wird als querschnittliche Aufgabe (cross-cutting concern) bezeichnet? Nennen Sie ein Beispiel und erläutern Sie, warum ihr Beispiel eine querschnittliche Aufgabe ist.

Aufgabe 10

[10 Punkte]

Angenommen, Sie sollen innerhalb einer Spring Anwendung folgendes Formular verarbeiten:

```
<form action="/employees/add" method="POST">
  <input type="text" name="name" id="namefield">
  <input type="text" name="adresse" id="adressfield">
  <input type="text" name="gehalt" id="gehaltfield">
</form>
```

Die Formularfelder für den Namen und die Adresse sollen in Strings, das Gehalt in einem double Wert weiterverarbeitet werden. Die Controllerklasse hat nur eine `@Controller` Annotation. Schreiben Sie die Methodensignatur und ggf. notwendigen Code sowie die Annotation(en) auf, die benötigt werden, um die Formularfelder in dem SpringMVC Controller zu verarbeiten.