Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Arbeitsgruppe Software Engineering

Prof. Dr. Wilhelm Hasselbring
M.Sc. Christian Zirkelbach

Institut für Informatik

1. Übung zur Vorlesung „Softwaretechnik“

Serie 1

Abgabe am Montag, 15. April 2019 - 10:00

Vorlesungsmaterialien

Die Vorlesungmaterialien findet ihr zum Download unter der iLearn-Beschreibung der Veranstaltung.

Übungsbetrieb

Der Übungsbetrieb wird über das iLearn und Git / GitLab durchgeführt. Dies bedeutet, dass ihr Übungsserien und Korrekturen hier im iLearn erhaltet. Die Übungsserien werden wie gewohnt in 2er Gruppen bearbeitet. Normalerweise habt ihr für die Bearbeitung der Übungsaufgaben ungefähr eine Woche Zeit. Trotzdem solltet ihr immer den Abgabetermin jedes Übungszettels beachten. Abgaben, die zu einem **späteren** als dem jeweils angegebenen Zeitpunkt erfolgen, können leider nicht mehr bewertet werden.

Die ersten Übungstermine finden in der Woche ab dem **15. April 2019** statt.

Anmeldung

Zur Teilnahme an der Vorlesung, den Übungen und der Klausur ist eine **Anmeldung** in der StudiDB notwendig. Bitte meldet euch mit eurem *StudiDB-Account* bei der Veranstaltung *Inf-ST Softwaretechnik Priv.-Doz. Dr. H. Schnoor* an.

Meldet euch zusätzlich mit einem Übungspartner in einer *Kleingruppe* im iLearn an.

Um für das GitLab-System (welches zu einem späteren Zeitpunkt genutzt wird) freigeschaltet zu werden, stellt bitte sicher, dass ihr eine korrekte E-Mail im iLearn eingetragen habt.

Klausur: Zulassung und Bonuspunkte

Um zur Klausur zugelassen zu werden, müsst ihr am Ende des Moduls mindestens 50% der maximalen Punkte erreicht haben. Alternativ habt ihr die Zulassung, wenn ihr bereits mindestens einen Fehlversuch für dieses Modul habt. Übungspunkte, die über die 50% hinausgehen, werden als Bonuspunkte für die Klausur gewertet. Maximal können 20 Bonuspunkte erreicht werden. Die folgende Formel gibt die Bonuspunkte B an, wobei p die erreichte Punktzahl und P die maximal erreichbare Punktzahl repräsentieren: $B = (20 \cdot \frac{(p/P)-0{,}5}{0{,}5})$. Wenn ihr also 50%, 75% oder 100% in den Übungen erreicht habt, erhaltet ihr 0, 10 bzw. 20 Bonuspunkte für die Klausur.

Aufgabe 1 - Qualitätsaspekte in der Softwareentwicklung

3 Punkte

Nennen Sie jeweils drei Eigenschaften der beiden Begriffe **interne Qualität** und **externe Qualität** im Kontext der Softwareentwicklung und erläutern sie diese.

Aufgabe 2 - Integrated Development Environment (IDE): Eclipse 7 Punkte

Die integrierte Entwicklungsumgebung Eclipse bietet viele Funktionen an, um schneller Quellcode zu schreiben und dabei weniger Fehler zu produzieren. Außerdem ermöglicht es Eclipse die eigenen Programme direkt innerhalb der IDE auszuführen, ohne auf die Konsole auszuweichen. Für diese und die nachfolgenden Programmieraufgaben soll **Java in der Version 1.8** verwendet werden. Bitte installieren Sie daher das Java SE Development Kit auf eurem Rechner.

Aufgabenteil a (6 Punkte)

Erstellen Sie mit Eclipse ein Java-Projekt mit dem Namen `S01A02`. Achten Sie hierbei insbesondere darauf, dass als JRE die Ausführungsumgebung `JavaSE-1.8` verwendet wird und keine konkrete (Unter-)version. Erstellen Sie einen Ordner mit dem Namen `lib` und kopieren Sie die Bibliothek commons-lang in der Version 3.8.1 von Apache dort hinein. Fügen Sie die neue Bibliothek zum Classpath eures Projekts hinzu, damit die Bibliothek gefunden und genutzt werden kann. Machen Sie hierzu einen Rechtsklick auf das Projekt und klicken Sie auf `Build Path->Configure Build Path...`. Anschließend wählen Sieden Reiter `Libraries` aus und fügen die Bibliothek mit `Add External Jars...` hinzu. Kopieren Sie anschließend die (eventuell ausgeblendete) Datei `.classpath` als `classpath-abs.txt`.

Erstellen Sie eine `main`-Methode in einer Java-Klasse mit dem Namen `Example` in einem neuen Paket namens `swt`. Implementieren Sie innerhalb der `main`-Methode die Ausgabe aller Kommandozeilenargumente separiert durch Kommas auf die Konsole. Nutzen Sie dazu den Aufruf `StringUtils.join(..)` und die entsprechende JavaDoc-Beschreibung aus der oben angegeben Bibliothek. Führen Sie anschließend euer Projekt aus, indem ihr folgende Schritte durchführt:

- Selekttion der Klasse `Example` links im `Package Explorer` mit einem Rechtsklick
- Klick auf `Run As->Run Configurations...`
- Selektion von `Java Application`
- Klicken Sie links oben auf das Piktogramm eines leeren Blattes mit einem Plus-Symbol (`New launch configuration`)
- Wählen Sie den Reiter `Arguments` in der neuen Launch Configuration aus und geben unter `Program arguments` die folgende Zeilen ein:

  ```
  Hello
  World
  !
  ```

- Drücken Sie schließlich auf `Run`
- Das Programm sollte nun Folgendes ausgeben: `Hello,World,!`

Entfernen Sie jetzt die Bibliothek `commons-lang` aus dem Classpath eures Projekts und fügen Sie erneut, dieses Mal jedoch mit Hilfe von `Add Jars...` , hinzu. Dies erfolgt über einen Rechtsklick auf das Projekt und einen Klick auf `Build Path -> Configure Build Path...`. Anschließend wählen Sie den Reiter `Libraries` aus und fügen die Bibliothek mit `Add Jars...` hinzu. Kopieren die Datei `.classpath` als `classpath-rel.txt`. Führen Sie nun das Programm erneut aus um Sicherzustellen, dass es immer noch funktioniert.

Bauen Sie anschließend ein Java Archive (kurz: jar-Datei) des Projekts, indem Sie das Projekt mit einem Rechtsklick selektieren und `Export...` auswählen. Wählen im neuen Fester `Java -> Runnable JAR file` aus und verwendem Sie die bereits bekannte `Launch Configuration`. Der Name der jar-Datei soll `swt-test.jar` lauten und im Eclipse-Projektverzeichnis erstellt werden. Klicken Sie anschließend auf `Finish`. Wenn Sie nach dem Export nun in einem Konsolenfenster (außerhalb von Eclipse) Folgendes eingeben, sollte die gleiche Ausgabe, die innerhalb von Eclipse ausgegeben wurde, zu sehen sein:

```
java -jar swt-test.jar Hello World !
```

Packen Sie anschließend das Eclipse-Projekt (inkl. die beiden Textdateien und der exportierten Jar) in ein Zip-Archiv mit dem Dateinamen `S01A02.zip` und laden dieses hoch.

Aufgabenteil b (1 Punkt)

Beantworten Sie außerdem die folgenden Fragen:

- Welche Variante ist für ein Projekt mit mehreren Entwicklern von Vorteil? Begründen Sie Ihre Antwort.
- Vergleichen Sie die Inhalte der beiden Dateien `classpath-abs.txt` und `classpath-rel.txt`. Was für Unterschiede sind zu erkennen?

Aufgabe 3 - Java Wiederholung - Objektgleichheit

5 Punkte

Die Java-Klasse `java.lang.Object` ist die höchste Oberklasse (oder auch Wurzel) der Java-Klassenhierarchie. Jede andere Klasse erbt somit unter anderem von `java.lang.Object`, daher auch ihre *nonfinal* Methoden wie z.B. `toString`, `hashCode` und `equals`. Letztere ist entscheidend für den Vergleich von Objekten.

Je nach Design (und Anforderungen) kann die Gleichheit von Objekten durch Objektidentität (`==` Operator) oder *logische Gleichheit* (Wertgleichheit von Attributen) bestimmt werden.

Gegeben ist die folgende Java Klasse:

```
public class Studierender {

  private final int matrikelnummer;

  private final String name;

  private boolean studiertInformatik;

  public Studierender(int matrikelnummer, String name, boolean studiertInformatik) {
    this.matrikelnummer = matrikelnummer;
    this.name = name;
    this.studiertInformatik = studiertInformatik;
  }

  // Getter und Setter ausgeblendet
}
```

Die `matrikelnummer` eines Studierenden ist nicht änderbar und wird einmalig vergeben. Der `Name` und der Status der Einschreibung für Informatik (`studiertInformatik`) kann von einem Studierenden zum Beispiel beim Studierendenservice geändert werden.

Aufgabenteil a (3 Punkte)

Nennen und erläutern Sie drei Eigenschaften, die Implementierungen der `equals` Methode einhalten sollten.

Aufgabenteil a (2 Punkte)

- Überschreiben Sie die `equals` Methode so, dass neben der Objektidentität auch die logische Gleichheit berücksichtigt wird. Überlegen Sie sich dafür zunächst, welche Attribute notwendig sind. Geben Sie Ihre Implementierung der `Studierender` Klasse als Quelltext ab.
- Die Klasse soll ebenfalls eine `Main` Methode enthalten, in der Sie die Korrektheit Ihrer `equals` Methode testen.

Aufgabe 4 - Softwarequalität: Kohäsion und Kopplung

4 Punkte

In der Softwareentwicklung werden verschiedene Metriken verwendet, um die Qualität einer Software zu bestimmen. Zwei solcher Metriken sind die **Kopplung** und die **Kohäsion**. In der folgenden Abbildung sehen Sie sechs Module (äußere Kästen mit Buchstaben), die jeweils eine unterschiedliche Anzahl an Komponenten (innere Quadrate) besitzen. Geben Sie für jedes Modul sowohl dessen Kopplung als auch dessen Kohäsion als Ganzzahl an. Geben Sie weiterhin an, welches Modul die höchste Qualität und welches die niedrigste Qualität bezüglich der Kombination beider Metriken besitzt.