

Hausaufgaben 6

Programmieren II – SS 2021

Erzeugen sie für jede Aufgabe ein eigenes Eclipse-Projekt. Erstellen Sie zum Schluss für alle erzeugten Klassen jeweils die JavaDoc (sie soll die JavaDoc für alle Methoden enthalten).

Exportieren Sie jeweils ihre Aufgaben in eine jar-Datei „AufgabeX.jar“. Die Aufgaben müssen dann jeweils aus der Konsole (ohne IDE) heraus aufrufbar sein. Diese Jar-Datei soll die binaries, den Sourcecode und die JavaDoc enthalten (vgl. Übungen 1).

Schicken sie ihre Lösung als gepackte Zip-Datei namens HA_6_<Matrikelnummer>.zip, welche jeweils eigene jar-Files für die Aufgaben enthält bis zum 20.06.2021 an jost@lehre.dhbw-stuttgart.de

Für jedes jar-File ist eine bat-Datei mitzuliefern, über die das Programm gestartet werden kann.

Aufgabe 1

Schicken Sie bis zum 20.06.2021 einen Vorschlag für Ihre Projekt-Idee. Der Vorschlag soll im Rahmen eines kurzen Prosa-Text die Ideen umschreiben. Für die finale Ablieferung ist dieser Prosa-Text zur fachlichen Grob-Spezifikation auszubauen.

Machen Sie sich Gedanken, welcher geschätzter Arbeitsaufwand für die Realisierung anfällt. Denken Sie dabei an folgende wichtigen Bewertungskriterien:

- a) Ein Prosa-Text der Aufgabenstellung soll vorliegen. Dieser dient als „fachliche Grob-Spezifikation“.
- b) Es muss ein UML-Klassendiagramm für alle Klassen vorliegen. Desweiteren sollen alle Aktoren und Use-Cases in einem Use-Case-Diagramm aufgeführt sein. Der wichtigste Anwendungsfall soll mit einen Aktivitätsdiagramm dargestellt sein. Desweiteren soll für diesen auch das Sequenzdiagramm der beteiligten Klassen vorliegen.
- c) Es sollen die Konzepte für die Anwendungsprogrammierung berücksichtigt werden zur Planung und Schnitt von Abhängigkeiten.
- d) Es sollen fachliche Ausnahmen und Laufzeitfehler sinnvoll eingesetzt werden, sowie Design- und Entwurfsmuster, wie sie in der Vorlesung noch vorgestellt werden, integriert sein.
- e) Die Klassen sollen gemäß den SOLID-Entwurfsprinzipien, wie sie in der Vorlesung noch vorgestellt werden, entworfen und implementiert sein.
- f) Die Entwicklerdokumentation (JavaDoc) enthält Klasseninvarianten für die Klasse, desweiteren sind alle Attribute und Methoden den Java-Konventionen gemäß benannt und dokumentiert. Die Datentypen sind passend gewählt. Die JavaDoc liegt wie in den Hausaufgaben geübt vor.
- g) Die Klassen für die fachliche Logik sollen eine C1-Überdeckung von mind. 60 % haben. Controller-Klassen und GUI-Klassen werden bei der Bewertung der Test-Überdeckung nicht berücksichtigt.
- h) Die Anwendung muss keine anspruchsvolle GUI haben. Es genügt auch eine einfache Ausgabe in die Konsole bzw. Testen der Fachlichkeit durch jUnit-Tests.

Aufgabe 2

Schreiben Sie eine Methode, welche einem Consumer Eingaben von der Eingabekonsole übergibt. Überlegen Sie sich, wie sie erkennen können, ob der Anwender weitere Eingaben in der Konsole machen möchte oder fertig ist. Eine Eingabe kann über mehrere Zeilen hinweg erfolgen (d.h. mehrere <Enter> bzw. <Strg>-Tastendrucke enthalten).

Testen sie diese Methode durch Übergabe geeigneter Consumer.

Aufgabe 3

Implementieren Sie einen Supplier für Studenten, d.h. einen Generator, der Testdaten für eine Studentenliste erzeugt. Für den Wohnort sollen statistisch zufällig verteilt drei Werte übergeben werden (Berlin, Stuttgart, München).

Schreiben sie unter Verwendung geeigneter Lambda-Ausdrücke eine Methode welche aus dem vom zuvor erstellten Supplier generierten Datenstrom die Namen der Studenten zu einem vorgegebenen Ort ausgibt, d.h. es wird der Methode ein Wohnort (Berlin, Stuttgart oder München) übergeben, und diese gibt dann die Namen der Studenten aus dem Stream aus, die in der jeweiligen Stadt wohnen.

Zur Kontrolle soll der vom Supplier erzeugte Datenstrom über eine geeignete Methode zum Ausspähen auf die Konsole geloggt werden.