

OOP2 Abschlussprojekt

1. Überblick

Das Abschlussprojekt gibt dir die Möglichkeit, deine Skills in Bezug auf den im Kurs vermittelten Stoff unter Beweis zu stellen. Entwickle eine Applikation, eine Animation, ein Spiel oder auch etwas Anderes, das deine Kompetenzen aufzeigt.

2. Umfang

- Die Applikation soll eine graphische Schnittstelle haben, die mit JavaFX implementiert wird. Dabei soll es die Möglichkeit geben, dass der Benutzer im Programm etwas eingeben kann und diese Eingabe dann entsprechend verarbeitet wird.
- Der Benutzer der Applikation muss mindestens zwei unterschiedliche Arten von Objekten über die Benutzerschnittstelle pflegen können. (z.B. Mitarbeiter und Abteilungen)
- Die Arbeit soll in 2er Gruppen gemacht werden → Saubere Absprachen & Projektaufteilung. Es muss klar und ersichtlich definiert werden, wer was macht (Pakete, Klassen). So dass auch auf einem Git-Branch gearbeitet werden kann und bei den Commits jeweils der entsprechende User ersichtlich ist.

3. Themen

Im Programm musst du beweisen, dass du die im Kurs gelernten Techniken selbständig umsetzen kannst. Folgende Themen sind dabei mindestens abzudecken:

- a) Erstelle mindestens vier Unit-Tests, die wichtige Teile deiner Applikationslogik prüfen
- b) Aufzählungstyp mit Enum
- c) Setze mindestens drei der folgenden Techniken ein: Vererbung, Casting, Interfaces, abstrakte Klassen oder Generics.
- d) Collections und Sortierung
- e) Gewisse Daten sollen vom Benutzer mit Java Serialisierung gespeichert werden können, dass diese auch nach dem Neustart der Applikation immer noch zur Verfügung stehen.
- f) Setze die Möglichkeiten von Exceptions gezielt ein, um Ausnahmen in deiner Applikation sauber abzuhandeln. Stelle dazu sicher, dass mit Hilfe von Log-Daten ein Fehler detailliert analysiert werden kann.
- g) Das Projekt soll als Maven-Projekt abgegeben werden und über den Build-Prozess von Maven erzeugt werden können. Ebenfalls soll der Abgabe eine bereits kompilierte und direkt ausführbare-Version des Programmes beigelegt werden. Setze auf Pakete, um deine Klassen gemäss eines gewählten Architektur-Ansatzes zu gliedern.

4. Abgabe Szenarien-Beschrieb

Du kannst im Kurs verwendete Problemstellungen bearbeiten. Deine Lösung muss sich aber deutlich von den Kursbeispielen unterscheiden. Ebenfalls sollte sich die Projekt-Arbeit klar von der Arbeit aus dem Kurs OOP1 unterscheiden. Falls du das Projekt aus dem ersten Kursteil weiterentwickeln möchtest ist dies möglich. Du musst dazu den Ist-Zustand und Soll-Zustand im Beschrieb klar abgrenzen und beachten, dass nur die neu eingebrachten Leistungen bewertet werden.

Erstelle einen einfachen Beschrieb zu deinem gewählten Szenario mit den wichtigsten Anforderungen an die Applikation und **stelle dieses Dokument bis zum erwähnten Termin** im entsprechenden Thema als PDF auf Moodle.

5. Arbeit mit Git und GitLab

Setze von Beginn weg auf die Versionsverwaltung Git und stelle sicher, dass du deine Arbeit sauber und schrittweise ins lokale Repository aufnimmst und auch auf das Schul-GitLab pushst. Überlege dir dabei, welche Dateien aus dem Maven-Projekt notwendig sind und welche Dateien du nicht zwingend in das Repository aufnehmen musst.

6. Dokumentation

Lege deinem Programm in der Moodle-Abgabe (siehe nächster Abschnitt) eine Dokumentation bei. In dieser soll der Zweck und die Benutzung der Applikation kurz erklärt werden. Setze auch auf Screenshots, mit welchen du das Endresultat deiner Arbeit präsentieren kannst. Dazu sollte die Dokumentation einen Überblick bieten, wie du die Applikation umgesetzt hast, damit sich ein anderer Programmierer schnell in den Source-Code hineindenken kann.

Beschreibe dabei auch, wie und weshalb du die unter Themen aufgeführten Punkte umgesetzt hast. (Z.B. Für Thema Collections: Weshalb hast du die entsprechende Collection-Art gewählt, wie und wo hast du sie eingesetzt? Usw.)

7. Abgabe Projektarbeit

Stelle sicher, dass du in der Abgabe-Dokumentation beschreibst, wie das der zu bewertende Source-Code auf dem GitLab aufzufinden ist und dass der Benutzer `alain.rohr@hftm.ch` Zugriff auf den Source-Code hat. (Rolle Developer zuweisen)

Erzeuge eine Zip-Datei mit einer ausführbaren Version deiner Applikation, sowie der Dokumentation als PDF. Lade das Archiv bis zum erwähnten Termin im entsprechenden Thema auf Moodle hoch.

8. Bewertung

Die Bewertung der Arbeit erfolgt nach folgenden Kriterien:

(In Klammern werden die Bedingungen für das Erreichen der Maximalnote aufgeführt.)

- Organisation und Übersicht: 10 Punkte
(Die Abgabe deiner Arbeit zeigt, dass du ein sinnvolles Vorgehen für das Erreichen des Resultates gewählt hast. Die Datei- und Inhalts-Struktur der Abgabe ist übersichtlich und klar. Es werden nur die notwendigen Dateien abgeben. Das Java-Projekt ist gut strukturiert. Die Art und Weise des Git-Einsatzes erleichtert die Übersicht und die Wartung des Source-Codes. Das Programm lässt sich auf Anhieb oder wie beschrieben starten. Die Dokumentation ist nachvollziehbar aufgebaut, enthält alle notwendigen Teile und bietet einen sehr guten Überblick.)
- Erzielen und dokumentieren der gewünschten Themen: 20 Punkte
(Die Aufgabestellungen werden umfangreich erfüllt und teilweise übertroffen.)
- Auftritt: 8 Punkte
(Die Benutzerschnittstellen sind sauber umgesetzt, übersichtlich und ansprechend. Das Erscheinungsbild der Dokumentation (Layout, Grafiken, Schriftbild, Umbrüche, Titelbild usw.) ist makellos und attraktiv. Der Text ist gut verständlich und grammatikalisch einwandfrei.)