

Vorlesung Softwaretechnik I · SoSe 2024

Prof. Dr. J. Grabowski · Dr. P. Makedonski · M.Sc. C. Bieber

Aufgabenblatt 9

Allgemeine Informationen

Alle Studierende sollen die Aufgaben, die unter dem Punkt *Vorbereitung auf die Übung* genannt sind, bis zum Übungstermin bearbeitet haben.

Vorbereitung auf die Übung

Für die Übung sollen **alle Studierende** Aufgabe 1 und 2 gelöst haben. Zudem sollen alle Studierende mit dem Prinzip des Refactorings vertraut sein.

Aufgabe 1

Sie wurden ausgewählt das Versionsverwaltungssystem für die Bibliothekssoftware zu initialisieren. Es wurde entschieden, dass Sie Git dafür verwenden sollen. Glücklicherweise hat ein Kollege von ihnen schon ein Template in das Versionsverwaltungssystem hochgeladen.

Führen Sie folgende Schritte aus und dokumentieren Sie Ihre Ergebnisse:

1. Loggen Sie sich beim GitLab der GWDG ein (<https://gitlab.gwdg.de>).
2. Forken Sie das Projekt **bibsoft** (<https://gitlab.gwdg.de/swe/bibsoft>). Dadurch erhalten Sie eine eigene Kopie des Projektes mit der Sie weiterarbeiten können.
3. Führen Sie ein `git clone` aus, sodass Sie das geforkte Projekt auf ihrer Festplatte haben. Siehe: [1].
4. Testen Sie, ob alles funktioniert hat, indem Sie in dem Hauptverzeichnis des Projektes (in dem eine `pom.xml` liegt) das Kommando `mvn package` auf der Kommandozeile ausführen. Hinweis: Dafür müssen sie Maven installiert haben. Sie finden in den Hinweisen einen Link zu der Dokumentation (Siehe: [2]). Weiterhin, achten Sie darauf, dass Sie mindestens Java Version 11 auf ihrem System eingerichtet haben.
5. Führen Sie das erstellte Jar-File mit dem Befehl `java -jar target/media-1.0-SNAPSHOT-jar-with-dependencies.jar` aus. Maven packt alle benötigten Dependencies mittels des `maven-assembly-plugins` in das Jar-File. Dadurch muss der Classpath für das Ausführen nicht extra angepasst werden.

6. Fügen Sie ein neues Medium im *models* Package hinzu. Es soll *Buch* heißen, eine ISBN besitzen und von der *Medium*-Klasse erben. Committed und Pushen Sie ihre Änderungen.
7. Implementieren Sie im *MedienController* die Funktionen *ausleihen* und *ruueckgabe*. Es reicht, wenn Sie in diesen Funktionen nur die Verfügbarkeit des Mediums korrekt setzen. Committed und Pushen Sie ihre Änderungen.
8. Der Kunde hat festgestellt, dass VHS-Kassetten nicht mehr im Medienbestand benötigt werden. Löschen Sie die *VHS*-Klasse und passen Sie die *Main*-Klasse an. Committed und Pushen Sie ihre Änderungen.
9. Es sind doch noch VHS-Kassetten beim Kunden aufgetaucht. Er möchte, dass das System nun doch wieder VHS-Kassetten unterstützt. Benutzen Sie das `git revert` Kommando um auf die vorherige Version zurückzuspringen. Hierzu müssen Sie den Hash des vorherigen Commits herausbekommen. Benutzen Sie dazu das `git log` Kommando. Pushen Sie ihre Änderung.

Aufgabe 2

Ihre Firma möchte auf Gradle als Build-System umsteigen, allerdings nicht das bestehende Projekt gefährden.

1. Erstellen Sie einen neuen Branch im Projekt.
2. Passen Sie die *.gitignore* an, sodass die automatisch generierten Dateien (*.gradle* und *build* Ordner) ignoriert werden und nicht mit committed werden. Committed und Pushen Sie ihre Änderungen.
3. Ändern Sie das Build-System des Projektes von Maven auf Gradle. Committed und Pushen Sie ihre Änderungen.
4. Mergen Sie den Branch zurück auf den Master, wenn Sie mit Gradle ein Jar-File bauen können. Hier reicht es, wenn Sie ein normales Jar-File mittels Gradle erstellen können, d.h. es ist nicht nötig das dieses ausführbar ist und alle Dependencies beinhaltet.

Hinweise: Siehe [3] und [4].

Ablauf der Übung

In der Übung wird das Arbeiten mit Versionskontrollsystemen, Build Systemen und das Prinzip von Refactorings vertieft.

Der Ablauf der Übung gestaltet sich folgendermaßen:

1. Vorstellen der Aufgaben 1 und 2
2. Gruppenbildung (4-6 Studenten)
3. Austeilen der Code-Beispiele und starten der Bearbeitungszeit für Aufgabe 3 und 4
4. Vorstellen der Ergebnisse durch zwei randomisiert ausgewählte Personen (pro Gruppe ca. 6-10 Minuten)

Aufgabe 3

In der Übung wird das Prinzip des Refactorings vertieft. Es werden Code Beispiele ausgeteilt (siehe StudIP Material) an denen ein Refactoring ausgeführt werden soll.

Aufgabe 4

Erstellen Sie anhand der *createMedium* Methode ein Sequenzdiagramm.

Hinweise

In der Online-Übung wird für die Modellierung von Sequenzdiagrammen das Tool www.draw.io genutzt. Hier können Sie, wenn Sie ein neues Diagramm erstellen (Blank) die Elemente (Shapes) für die verschiedenen Diagramme auf der linken Seite unter UML finden.

Hinweise für diese Aufgaben finden Sie in den Vorlesungsfolien zur Implementierung und in den folgenden Quellen:

- 1 Git-Handbuch: <https://git-scm.com/book/de/v1>
- 2 Maven-Handbuch: <http://maven.apache.org/guides/>
- 3 Gradle-Handbuch: <https://gradle.org/guides/>
- 4 Gradle-Java-Tutorial: https://docs.gradle.org/current/userguide/tutorial_java_projects.html