ÜBUNGSZETTEL TDD

AUFGABE 1: THEORIE

Arbeite dich in die Theorie zum Thema Testen bzw. Test-Driven-Development (TDD) ein. Nutze dazu die Materialien aus den verschiedenen „Input“ Abschnitten im Themenbereich „Test-Driven Development“ unseres Moodle-Kurses (https://moodle.tsn.at/course/view.php?id=24763#section-8). Nach dieser Aufgabe musst du folgende Punkte erklären können:

- Testdriven Development
- Red-Green-Refactor
- FIRST-Acronym
- Kent Beck (welche Rolle spielt er in Bezug auf TDD)
- Testarten
    - Unit-Tests (Sociable, Solitary, Mocks)
    - Integrationstests
    - UI-Tests / End-To-End Tests / Systemtests
    - Akzeptanztests
- Testpyramide
- JUNIT (Junit5)
- Mockito (Sinn und Funktionsweise von Mocking-Bibliotheken)

AUFGABE 2: AUSGANGSPROJEKT

Laden Sie sich das gegebene Maven-Ausgangsprojekt („TDD Kino Demo“, siehe Moodle) herunter. Laden Sie es als Maven-Projekt in ihre IDE und schauen Sie sich an, wie das Projekt aufgebaut ist:

- pom.xml (Dependencies, Java-Version etc.)
- Gegebene Domänen-Klassen (Kinosaal, Ticket etc.)
- Gegebene Start-Junit5-Tests in test /java/at.itkolleg/AppTest

Starten Sie den Test AppTest über den grünen Pfeil und versichern Sie sich, dass alles korrekt läuft. Starten Sie auch die App (main-Methode).

AUFGABE 3: EINARBEITUNG IN DEN GEGEBENEN CODE

Arbeiten Sie sich in den gegebenen Code zur Kinoverwaltung ein. Verwenden Sie die gegebenen Klassen KinoSaal, Ticket, Vorstellung, Kinoverwaltung in der App-Klasse (main-Methode), um ein Gefühl für die Funktionsweise des Programms zu bekommen. Führen Sie folgende Punkte durch:

- Kinosäle anlegen
- Vorstellungen anlegen
- Vorstellungen über die Kinoverwaltung einplanen
- Tickets für Vorstellungen ausgeben
- etc.

AUFGABE 4: JUNIT-TESTS FÜR KINOSAAL

Testen Sie alle Methoden der Klasse KinoSaal (Testklasse TestKinoSaal).

AUFGABE 5: JUNIT-TESTS FÜR VORSTELLUNG

Testen Sie alle Methoden der Klasse Vorstellung (Testklasse TestVorstellung).

AUFGABE 6: JUNIT-TESTS FÜR KINOVERWALTUNG

Testen Sie alle Methoden der Klasse KinoVerwaltung (Testklasse TestKinoverwaltung).

AUFGABE 7: JUNIT-TESTS ADVANCED

Falls nicht schon in den vorhergehenden Aufgaben passiert, testen Sie folgende Punkte unter Verwendung der fortgeschrittenen Features von JUNIT 5:

1. Schreiben Sie einen Test, der validiert, dass das Anlegen einer Vorstellung korrekt funktioniert. Der Test sollte eine fachliche Bezeichnung haben und die Assertions sollten bei Validierungsfehler eine Hinweistext liefern.
2. Schreiben Sie einen Test, der validiert, dass das Einplanen mehrerer Vorstellungen korrekt funktioniert. Stellen Sie zudem sicher, dass beim möglichen Auftreten eines Fehlers trotzdem alle Validierungen ausgeführt werden.
3. Schreiben Sie einen Test, der sicherstellt, dass ein Fehler geworfen wird, wenn eine Veranstaltung doppelt eingeplant wird.
4. Schreiben Sie einen parametrisierten Test, der mehrere Ticketkäufe mit unterschiedlichen Parametern überprüft.
5. Schreiben Sie eine dynamische TestFactory die den Ticketkauf mit zufälligen Werten bombardiert. Der Test soll sicherstellen, dass der Ticketkauf entweder funktioniert oder nur einen der definierten Fehlermeldungen (z.B. new IllegalArgumentException("Nicht ausreichend Geld.")) ausgibt. Die Tests müssen reproduzierbar sein.

AUFGABE 8: MOCKITO EINFÜHRUNG

Lesen Sie sich in das Mocking-Framework Mockito ein (Links siehe Moodle im Abschitt „Input zu Mockito“).

Verwenden Sie die wesentlichen Mockito-Möglichkeiten praktisch in kleinen Programmen.

AUFGABE 9: SELENIUM EINFÜHRUNG

Lesen Sie sich in das Browser-Testframework Selenium ein (Links siehe Moodle im Abschitt „Input zu Selenium“).

Verwenden Sie das gegebene Beispiel und das Tutorial „Guide to Selenium with JUnit / TestNG“ um die Möglichkeiten von Selenium praktisch auszuprobieren.

AUFGABE 10: TDD IST DEAD

Der Diskurs über Sinn und Unsinn von TDD in der Praxis ist durchaus kontroversiell. TDD hat offensichtlich nicht nur Vorteile.

Versuch über folgenden Link herauszufinden, welche Argumente Kritiker und Befürworter zum Thema TDD bzw. automatisiertem Testen von Code ins Treffen führen und übertrage deine Erkenntnisse in die eigene Praxis.

https://www.youtube.com/watch?v=YUm4P2b0YPI

https://www.youtube.com/watch?v=vOO3hullcsY

https://martinfowler.com/articles/is-tdd-dead/

Und noch eine Frage: wie kann eine KI beim automatischen Testen von Software hilfreich sein? Recherchiere!