

Dokumentation

JDK Version 1.8.0_91 oder höher

Offene Punkte:

- Babysteps und ATDD sind schlecht kompatibel. Wir haben uns entschieden immer nur eines von beiden aktiv zu haben
- Desweiteren gehen wir davon aus, dass die XML-Dateien korrekt und nach dem Vorbild der XML-Beispiel-Datei gestaltet sind.
- Die genaue Umsetzung der Refaktorphase wurde uns freigestellt. Daher haben wir uns dafür entschieden es dem User zu erlauben sowohl Tests als auch Codes gleichzeitig zu bearbeiten, da beim anschließenden Übergang in die Testphase sichergestellt wird, dass der Code kompiliert und die Tests immer noch durchlaufen.

Programmorganisation

Haupteinstiegspunkt der Anwendung ist die GUIMain-Klasse, welche von Application erbt und die main-Methode implementiert.

Diese lädt die Programmoberfläche aus einer fxml-Datei und speichert relevante Controls.

Alle Eventhandler für Buttons und dergleichen werden dort festgelegt, Ressourcen wie Grafiken geladen. Hier werden auch die Logik Objekte, wie InterfaceManager, StateManager und TimeManager initialisiert und verwaltet.

Der InterfaceManger sorgt für die Kommunikation zwischen den Logik Elementen und den Elementen der GUI, wie den Textareas oder Labels. Ausgelesener Text wird formatiert und als Code Objekt gespeichert/weitergereicht.

Der gesamte Dateiverkehr wird über einen statischen FileManager organisiert. Der wiederum den statischen XMLManagers nutzt um die Felder und Attribute der XML Datei einzulesen. Dabei unterscheidet man zwischen dem Input der Klassen, sowie der Einstellungen. Die Klassen werden ausgelesen und als Code Objekt gespeichert. Wenn der Nutzer etwas speichert, werden die Änderungen in der XML-Datei eingetragen. Außerdem werden die Einstellung babysteps und atdd dort über das Eintragen von "true" oder "false" gesetzt und für ATDD wird ein Testskelett bereitgestellt.

Das Herzstück des Projekts bildet dabei der StateManager. Dieser hat als Aufgabe gemeinsam mit dem InterfaceManager und dem CodeManager den Code und den TestCode zu verwalten. Vor allem managet er die Phasen des Programms, die da lauten "RED"(Phase für fehlschlagenden Test), "GREEN"(Phase für Code schreiben, der den Test erfüllt), "Refactor"(Möglichkeit zum Verbessern von Code und TestCode) und ATDD(Schreiben eines Akzeptanztest). Mit nextStep() regelt die Klasse dabei den Übergang zwischen den Phasen.

Gesteuert durch den StateManager stellt der CodeManager Funktionen zum Aufrufen und Verwalten der Codeklassen bereit.

Für die Funktion babysteps wird noch die Klasse Timemanager benutzt. Sie stellt die Funktionen und Variablen bereit, die Zeitbegrenzung überwachen und verwalten.

In der Klasse RealCompileManager wird die Schnittstelle zur bereitgestellten Kata Bibliothek übernommen. Hier werden Funktionen bereitgestellt um Code zu compilieren, Tests auszuführen und entsprechende Ergebnisse zu erfragen, die im StateManager benutzt werden um den korrekten Wechsel zwischen den Phasen (siehe Handbuch) zu gewährleisten.

Im Resource Folder befinden sich sowohl die fxml Datei für die GUI als auch zu testzwecken erstellte XML Dateien. Im Test Odner sind einige Tests bereitgestellt, die die Funktionalität der einzelnen Klassen überprüfen.