Projekt #7 - Bericht

#Main.java

Die Starterklasse. Ihre Aufgabe besteht darin mittels launch die GUI zu starten.

#GUI.GUI.java

Die GUI des Programms, also die komplette Benutzeroberfläche. Neben der typischen Grundfunktionen zur Ausgabe der grafischen Benutzeroberfläche, verfügt sie außerdem über die Möglichkeit Code abzuspeichern, sowohl als XML Datei als auch Java Datei.

#GUI.Controller.java

Der Controller für GUI. Dient hauptsächlich zum Ausführen der relevanten Operationen, die bei Interaktion auftreten.

> **void setUpTable(TableView t, int i)**
>
> Erstellt die Tabelle für die Anzeige der Aufgabenauswahl. Unterscheidet zwischen normalem Fall und aktivierten Babysteps.
>
> **void detectRowClick(TableView t)**
>
> Handhabt klick des Nutzers bzw. die Auswahl einer Aufgabe aus dem Katalog und liest diese ein.
>
> **Exercise getCurEx()**
>
> Gibt ausgewählte Aufgabe zurück.
>
> **boolean compileTest(String test, String code,String accCode,String accName,TextArea txtError, boolean ShouldOnlyCompile)**
>
> Überprüft mittels CompileHandler ob die Tests kompilieren, gibt dann true zurück, ansonsten false.
>
> **boolean[] compileOnlyRefactoring(String test, String code,String accCode,String accName,TextArea txtError)**
>
> Führt normalen Test so wie Akzeptanztest aus. Gibt für beide jeweils true oder false aus, je nachdem ob sie laufen oder nicht.

boolean compileOnlyTestAndCode(String test, String code,TextArea txtError,boolean ShouldOnlyCompile)

Kompiliert nur den Code und TestCode und überprüft ob die Tests laufen. True wird im Fall, dass alle Tests laufen zurückgegeben, ansonsten false. Sollte der Test failen wird eine Meldung ausgegeben.

void startTimer(Label lblTimer,TextArea txtTest,TextArea txtCode)

Startet den Timer für die Babysteps und updated die Anzeige dessen in der GUI.

void stopTimer()

Stoppt den Timer.

#GUI.TableData.java

Handhabt den Aufbau der Auswahltabelle für die Aufgaben.

#XMLParser.XMLReader.java

Klasse zum Einlesen von XML Dateien, also der Aufgaben. Die Aufgaben werden in String umgewandelt und anhand von diesen wird ein neues Exercise Objekt erzeugt, das in einer Liste gespeichert wird.
Die Klasse ermöglicht die Rückgabe einer Liste aus Exercise Objekten.

#XMLParser.Exercise.java

Klasse zum Handhaben der Aufgaben. Im Prinzip stellt ein Exercise Objekt eine eingelesene Aufgabe dar und ermöglicht einfachen Zugriff auf alle wichtigen Informationen der Aufgaben, die z.B. für CompileHandler benötigt werden.

#XMLParser.XMLWriter.java

Wie in ProPra Übung besprochen schreibt er das XML (Nodes etc.)

#CompileHandler.CompileHandler.java

Ermöglicht es mit der Hilfe von Jens Bendispostos virtual-kata-lib Code als Strings entgegen zu nehmen, diese zu kompilieren und auch Tests auszuführen. Hauptfunktion liegt darin die anderen Klassen mit den relevanten Informationen über das Kompilieren und den Status der Tests zu versorgen.

Es wird beachtet ob ein Test oder Akzeptanztest kompiliert, ob er erfolgreich ist oder fehlschlägt und überprüft ob jegliche Klasse, mit der er versorgt wird auch kompiliert; ansonsten werden Fehlermeldungen zurückgegeben.

#CoundownTimer.CountdownTimer.java

Der für das Feature Babysteps relevante Timer. Hat Funktionen, die dazu dienen den Status, also die Zeit zurückzugeben, die bisher vergangene Zeit zu messen und den Timer überhaupt zu starten.