

Bericht

Donnerstag, 14. Juli 2016 17:54

Meine Aufgabe in der Gruppe NoSuchFieldException war es die Funktion Babysteps vollständig auszuarbeiten und zu implementieren. Babysteps ist eine Funktion die dem Nutzer aufzwingen soll kleinschrittig zu arbeiten. Der Nutzer kann hierfür einen Timer einstellen der x Minuten lang andauert. Läuft der Timer ab werden alle Änderungen seit dem letzten Phasenwechsel in der aktuellen Phase gelöscht und der Timer wird beendet um dem Nutzer die Möglichkeit zu lassen ihn neu einzustellen oder ohne Babysteps weiterzuprogrammieren. Drückt der Nutzer jedoch rechtzeitig auf "nächster Schritt" bevor der Timer abgelaufen ist, und alle Bedingungen erfüllt sind bzw. die Phase wirklich gewechselt wird, startet sich ein neuer Timer für die nächste Phase. (Außer es handelt sich dabei um die Refactor Phase, welche zeitlich immer unbegrenzt ist.)

Klassenbeschreibung:

Babysteps umfasst einige Klassen, von denen jedoch im Endeffekt nicht alle genutzt wurden (Teamarbeit erfordert Opfer). Das package "babysteps" umfasst die Interfaces BabystepsTimer, BabystepsUser und TDDCycle und die Klassen CustomTimer, Utils und BabystepsCylce. Die beiden Cycle Klassen sollten eigentlich den Ablauf der Phasen regeln, kamen jedoch nie wirklich zum Einsatz, da bereits ein viel weniger sicheres System mit javafx.scene.paint.Color benutzt wurde. Das Interface BabystepsTimer ist vermutlich das Herzstück der Zeitregelung. Dieses Interface verfügt Methoden zum Starten und Stoppen eines Timers, zum Überprüfen der Verbleibenden Zeit und zum Registrieren eines BabystepsUsers, welcher dann in der notifyAllUsersTimeElapsed Methode über das Ablaufen der Zeit informiert wird. Die Klasse CustomTimer erbt dieses Interface und implementiert seine Methoden sinnentsprechend. In einer früheren Version wurde noch zwischen Test- und Codezeit unterschieden, dies hat sich jedoch als zu umfangreich erwiesen. Das Interface BabystepsUser verfügt nur über eine Methode notifyTimerElapsed welche aufgerufen wird, sobald der Timer, in der sie registriert wird, abläuft. Dieses Interface wird in der aktuellen Implementierung von keiner Klasse geerbt, sondern direkt als anonyme Klasse implementiert (siehe MenuController). Die Utils Klasse verfügt lediglich über eine Methode zur Konvertierung eines longs (Zeit in Millisekunden) zu einem formatierten String.

Der eigentliche Ablauf wurde komplett in der MenuController Klasse integriert.

Julian Winter