

Programmierpraktikum 2016

ImmortalsGG

Bewertungsbogen

17.07.2016

Grundlegende Anforderungen

- | | |
|--|---|
| <input checked="" type="checkbox"/> Anwendung lässt sich kompilieren | <input type="checkbox"/> Exceptions werden behandelt |
| <input checked="" type="checkbox"/> Bericht vorhanden | <input type="checkbox"/> Nutzerhandbuch vorhanden |
| <input checked="" type="checkbox"/> Bericht verständlich formuliert | <input type="checkbox"/> Nutzerhandbuch verständlich formuliert |

Kommentar

Beim Klicken auf den Button *Dark Mode* wirft die Anwendung eine **NullPointerException**. Nach einem Klick auf *Dark Mode* ist es nach der Auswahl einer Aufgabe nicht mehr möglich ins Hauptmenü zu wechseln, da ebenfalls eine **Exception** geworfen wird. Das Nutzerhandbuch ist eine Kopie des Berichts und daher nicht vorhanden.

Projektbezogene Anforderungen

- | | |
|--|---|
| <input checked="" type="checkbox"/> Gradle-Skript fürs Bauen der Anwendung | <input checked="" type="checkbox"/> Travis CI eingebunden |
| <input type="checkbox"/> README.md mit Gradle Task und Anleitung | <input checked="" type="checkbox"/> Lizenz vorhanden |

Kommentar

Innerhalb der **README.md** wird weder der zuständige Gradle Task noch die nötigen Schritte zum Starten der Anwendung angegeben.

Anwendungsbezogene Anforderungen

- | | |
|---|---|
| <input checked="" type="checkbox"/> Anwendungsszenario wurde eingehalten | <input checked="" type="checkbox"/> Katalog kann mit Texteditor editiert werden |
| <input checked="" type="checkbox"/> Katalog von möglichen Übungen ist vorhanden | <input type="checkbox"/> Katalog ist nicht fest eingebaut |
| <input checked="" type="checkbox"/> Übungen beinhalten Vorlagen für Klassen und Tests | <input type="checkbox"/> Übungen beinhalten Konfiguration |
| <input checked="" type="checkbox"/> Prozess wird visuell dargestellt | <input checked="" type="checkbox"/> Mindestens zwei Erweiterungen implementiert |
| <input checked="" type="checkbox"/> virtual-kata-lib wird benutzt | <input type="checkbox"/> Ausreichende Menge an Unit-Tests |

Kommentar

Der Aufgabenkatalog wurde nicht mit *UTF-8* kodiert. Aus diesem Grund werden Umlaute und andere Sonderzeichen falsch dargestellt. Das gleiche gilt für Java-Dateien. Außerdem ist der Katalog fest im Programm eingebaut (**TDDUebungTests.java:20**) wodurch ein Austausch des Katalogs nur bei vorliegendem Source Code möglich ist. Das Programm wird an keiner Stelle getestet, da keine Unit-Tests vorhanden sind.

Feedback/Bewertung

Die Anwendung an sich gefällt mir sehr gut, da am Code sehr viel Kreativität zu erkennen ist. Zum Beispiel ist der BabySteps Counter nicht nach einer herkömmlichen Lösung entworfen, sondern zeigt dass sich Gedanken zur benötigten Funktionalität gemacht wurden. Das Design der Anwendung gefällt mir ebenfalls sehr gut, da zu jeder Zeit klar erkennbar ist welche Optionen gewählt werden können und in welchem Zustand sich die Anwendung befindet.

Ein Nutzerhandbuch, das unerfahrenen Benutzern die Anwendung erklärt, ist dennoch eine Anforderung des Projekts und muss nachgereicht werden. Außerdem muss innerhalb der `README.md` beschrieben werden mit welchem Gradle Task sich die Anwendung bauen lässt und wie man diese hinterher starten kann. Zur Zeit findet sich dort bloß der Travis CI Build-Status.

Ein weitere wichtiger Punkt ist der zur Anwendung gehörende Katalog. Zur Zeit wird ein fest eingebauter Katalog (im Package `de.hhu.intgg.uebungen`) verwendet, welcher sich nur durch Editieren des Source-Codes ändern lässt. Hier muss dem Nutzer die Möglichkeit gegeben werden einen Katalog auszuwählen.

Im Wesentlichen ist die Anwendung von Funktion und Design her auf einem sehr guten Stand, muss jedoch noch folgende Punkte erfüllen:

- Der oben genannte im Zusammenhang mit *Dark Mode* stehende Fehler muss behoben werden (`Exceptions` und Hauptmenü)
- Der Aufgabenkatalog muss mit einer vom Programm verständlichen Textcodierung versehen werden.
- `README.md` muss Gradle Task und Anleitung beinhalten
- Der Katalog muss in der Anwendung frei wählbar (von der Festplatte des Nutzers) sein.
- Kritische Stellen des Programms (z.B `TDDCompiler.java`) müssen durch Tests abgedeckt werden.

Die genannten Punkte werde ich über Issues verteilen.

Sie sollen dann auch nur von der jeweiligen Person bearbeitet werden.

Ob ihr individuell bestanden habt, erfahrt ihr im Auas.

Die letzte Abgabefrist ist der **25.07.2016 23:59 Uhr** (andernfalls schaffe ich die Korrektur evtl. nicht).