Nutzerhandbuch

Dieses Nutzerhandbuch soll Teilnehmern der Informatik I Vorlesung helfen TDDT zu verwenden.
Der test driven development Trainer ist eine Anwendung bei der
ein Test vor dem zu testenden Code geschrieben wird.Sie werden 3 verschiedene Phasen durchlaufen. Zuerst kommt die rote Phase bei der man den Test schreibt daraufhin kommt man in die grüne Phase bei der man den Code zum Test schreibt und danach in die refactor Phase bei der man Test und Code überarbeiten kann.

Zunächst einmal müssen Sie nach dem starten des Programms aus dem Katalog eine Übungsaufgabe auswählen.
Hierbei sind manche der Übungen mit “Babysteps” oder “Tracking” gekennzeichnet.

Bei Babysteps hat man für die rote und grüne Phase einen Timer und man muss mit der Phase fertig werden bevor der Timer endet sonst wird der neue Test/Code gelöscht und man kommt in die Phase davor.Babysteps haben den Sinn dem Nutzer die Entwicklung in kleineren Schritten nahezulegen/anzutrainieren.

Bei Tracking wird aufgezeichnet was Sie alles ändern und auch wann. Tracking hat den Nutzen dass Sie wissen für welche Aktivitäten Sie wie lange gebraucht haben.
Die Daten konnen analysiert werden um Probleme zu erkennen in die die Nutzer häufig laufen.Diese Daten werden in Form eines Diagramms ihnen gezeigt.

Sollten sie eine Übung auswählen bei der keine der 2 genannten Optionen dran stehen so haben Sie für jede Phase beliebig viel Zeit und auch keine anderen Sonderfunktionen.

Wenn Sie ihre Aufgabe ausgewählt haben fangen sie in der roten Phase an.Das rechte Textfeld ist für den Test, das linke Textfeld für den Code und das mittlere Textfeld zeigt ihnen die Fehler die sie beim kompilieren machen bzw. zeigt ihnen an auch an wie viele Fehler sie haben.er Button back ist dafür da um in die vorherige Phase zurückzukehren dies ist nur in der grünen Phase möglich.Der Check Button testet ihren bisher geschriebenen Test/Code und der next Button ist anwählbar sobald man in der Lage ist in die nächste Phase zu wechseln dies ist auch die Funktion dieses Buttons.

In der roten Phase schreiben sie in das rechte Textfeld ihren Test.Ziel dieser Phase ist es dass sie beim kompilieren maximal einen Fehler zu haben.

Sollte dies geklappt haben können Sie in die grüne Phase wechseln und anfangen einen Code für den Test in der roten Phase zu schreiben.Hier haben sie nun die Möglichkeit auch zurück in die rote Phase zu wechseln wobei hierbei dann der neu geschriebene Code von grün gelöscht wird.Sollte alles fehlerfrei kompilieren dann können Sie weiter in die refactor Phase.
In der refactor Phase sind sie nun in der Lage Code und Test nochmal zu überarbeiten.

Wenn Sie fertig sind klicken Sie auf den beenden Button oder falls Sie auch die Tracking Funktion aktiv haben auf den Button Beenden/Chart anzeigen.

Sie haben ausserdem auch die Möglichkeit selbst Übungen zu schreiben.Hierfür müssen sie im Ordner resources die xml Datei Exercise öffnen.

Ein Beispiel für eine solche Übung ist hier der Taschenrechner :

```
<exercises>
    <exercise name="Taschenrechner">
        <description>
            Erstellt einen konsolenbasierten Taschenrechner.
        </description>
        <classes>
            <class name="Calculator">
                public class Calculator{

                }
            </class>
        </classes>
        <tests>
            <test name="CalculatorTest">
                import static org.junit.Assert.*;
                import org.junit.Test;

                public class CalculatorTest{
                @Test
                public void testSomething(){

                }
                }
            </test>
        </tests>
    </exercise>
</exercises>
```

Und mit folgenden Zeilen können Sie auch Babysteps oder die Tracking Funktion aktiviert haben.

```
<config>
    <babysteps value="True" time="2:00" />
    <timetracking value="True" />
</config>
```