

Starten des Programms

Zum Starten des Programms in der Kommandozeile zunächst in den Pfad der TDDT.jar Datei wechseln und dann:

```
java -jar TDDT.jar  
eingeben
```

Notwendige Vorbereitungen vor Programmstart

Vor Programmstart müssen die Übungsdateien in den Ordner "Übungen" kopiert werden.

Die Form der Übungsdateien kann in den beigefügten Beispieldateien gesehen werden.

Noch eine Information: Der Klassenname für die CodeKlasse wird nach der Übungsdatei benannt und der Klassenname für die TestKlasse wird nach dem Klassennamen für die CodeKlasse + Test benannt.

Bsp:

Übungsdatei: Beispiel.txt

Klassenname der CodeKlasse: Beispiel

Klassenname der TestKlasse: BeispielTest

!!! Achtung: Übungsdateien sollen nach gültigen Klassennamen benannt werden!!!

Programmfunktionen

Eine kurze Zusammenfassung der Funktionen von den einzelnen Bestandteilen des Programms:

BabySteps:

Ist eine Funktion, die die Zeit des Nutzers für die Phasen in Red und Green limitiert.

Es wird vor dem Start des Hauptprogramms abgefragt, ob der Nutzer diese Funktion nutzen möchte.

Nach Ablauf der Zeit kehrt das Programm zur vorherigen Phase zurück.

(Also bei Green -> Red und bei Red -> Red)

Tracking:

Öffnet ein Fenster in dem Statiken zur Programmnutzung angezeigt werden,

unter anderem wie lange der Nutzer in jeder Phase war und welche Meldungen

bisher angezeigt wurden (bspw. Fehler, Meldungen oder Phasenwechsel)

Hilfe:

-> *Benutzerhandbuch:* Öffnet diese Datei

-> *Info:* Zeigt Informationen zum Programm an

Next-Button: Iteriert in der Liste der Übungsdateien nach unten

Previous-Button: Iteriert in der Liste der Übungsdateien nach oben

Button mit Übungsdatei-Name:

-> *Klick:* Startet das Hauptprogramm mit Konfigurationen aus der Übungsdatei

-> *Mit der Maus rübergehen:* Eine Beschreibung der Übung wird angezeigt

Nun eine etwas ausführlichere Erklärung zum Hauptprogramm:

Der Ablauf des Hauptprogramms ist in mehrere "Phasen" unterteilt.

Die einzelnen Phasen sind:

- Red**
- Green**
- Refactor**

In Red wird der Benutzer aufgefordert Tests zu schreiben,
wobei genau einer dieser Tests fehlschlagen muss.

Wenn diese Bedingung erfüllt ist kann zu Green gewechselt werden.

Oder wenn es einen Kompilierfehler im Code in Red oder in Green gibt.

In Green wird der zu testende Code geschrieben.

Der Benutzer kann nach Wunsch wieder zurück zu Red wechseln.

Wobei dadurch jedoch der geschriebene Code in Green gelöscht wird.

Wenn alle Tests erfolgreich sind kann der Benutzer zu Refactor wechseln.

In Refactor kann der Code aus Green verbessert werden.

Nach erfolgreichem Wechseln von Refactor zu Red wird der bisherige Code gespeichert
und die ganze Prozedur startet von neu.

Zusätzlich kann die Funktion BabySteps aktiviert werden.

Die Zeit in Red und Green wird dann limitiert und nach Ablauf der Zeit wird in die vorherige Phase gewechselt.

Bei Ablauf der Zeit von Green wird zu Red gewechselt und bei Ablauf der Zeit von Red wird zu Red gewechselt.

Weitere Informationen können auf folgender Seite gefunden werden:

<http://www.frankwestphal.de/TestgetriebeneEntwicklung.html>

Wir wünschen noch viel Spaß mit dem Programm!