

Nutzerhandbuch

für den TDDT

1. Was ist TDD?.....	S.2
2. Das Hauptmenü.....	S.2
3. Die drei Phasen.....	S.2
- RED.....	S.2
- GREEN.....	S.2
- REFACTOR.....	S.3
4. Die Erweiterungen.....	S.3
- Babysteps.....	S.3
- Tracking.....	S.3
- ATTD.....	S.3

Was ist TDD?

TDD steht für „Test driven Development“, also testgetriebene Entwicklung. Dabei handelt es sich um eine Methode der Softwareentwicklung, bei der die Tests *vor* dem Code geschrieben werden. Dadurch wird die Arbeit unter anderem in kleinere Schritte aufgeteilt und durch die Tests ist zu jedem Zeitpunkt die Funktionalität des Programms gesichert. Wer weitergehende Informationen zum Prinzip des TDD erhalten möchte, kann sich z.B. diesen Text von Frank Westphal durchlesen:

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

Das Hauptmenü

Das Hauptmenü ist der Start des TDDT. Von hier wählt und startet man die Übungen. Über die Schaltfläche „Darkmode“ in der linken unteren Ecke des Fensters kann man den Hintergrund des Menüs verdunkeln. Weiterhin kann man hier auswählen welche Übung man starten möchte, indem man eine aus der Liste der verfügbaren Übungen auswählt. Weiterhin kann man per Klick auf die entsprechende Schaltfläche auswählen welche Erweiterungen man in seiner Übung haben möchte. Wenn man Babysteps auswählt kann man über das Feld daneben außerdem auswählen, wie viel Zeit man pro Phase jeweils hat. Ist man mit allen Einstellungen zufrieden kann man die Übung starten.

Die drei Phasen

- RED:

Beim TDDT beginnt man immer in Phase RED. In dieser Phase muss man jeweils genau einen fehlschlagenden Test schreiben. Die Tests werden jeweils immer in das rechte Textfeld geschrieben. Ist man mit seinem Test zufrieden, kann man auf speichern klicken. Sollte nun alles kompilieren und es schlägt genau ein Test fehl wird man in die nächste Phase weitergeleitet.

Sollte der Testcode jedoch nicht kompilieren, erhält man eine entsprechende Fehlermeldung mit einem Hinweis auf die Art des Fehlers.

HINWEIS: Wenn der Code nicht kompiliert wird man hier gefragt ob man trotzdem den Modus wechseln möchte. Dies ist für den Fall gedacht, das man einen Test für eine noch nicht existente Methode schreibt. Sollte dies nicht die Quelle des Fehlers sein muss man den Fehler trotzdem korrigieren bevor man zur nächsten Phase übergeht.

-GREEN:

In dieser Phase geht es darum Code zu schreiben, der alle Test erfüllt. Dabei ist das Ziel diesen Code so knapp wie möglich zu formulieren. Der Programmcode wird immer ins linke Textfeld geschrieben. Sobald man mit seinem Code fertig ist kann man auf Refactor drücken. Wenn der Code kompiliert und alle Tests erfüllt sind wird man dadurch zur nächsten Phase weitergeleitet. Sollte der Code nicht kompilieren, oder die Tests nicht erfüllt sein, erhält man eine entsprechende Fehlermeldung und muss den Fehler erst

korrigieren.

HINWEIS: Sollte der Fall eintreten, dass die Tests nicht kompilieren oder nicht erfüllbar sind kann man über die Schaltfläche Menü→Zurück in den TestModus, jederzeit zur letzten Phase zurückkehren und die Tests korrigieren. Dabei geht jedoch jeglicher Fortschritt verloren.

-REFACTOR:

In dieser Phase ist das Ziel den vorhandenen Code zu vereinfachen und zusammen zu fassen. Dafür kann man den Code beliebig bearbeiten. Am Ende dieser Phase muss jedoch weiterhin der gesamte Code kompilieren und alle Tests müssen erfüllt sein. Ist man mit dem Refactoring fertig oder möchte nichts verändern drückt man auf Speichern und das Programm zeigt einem wiederum eventuelle Fehler an. Sind alle Fehler behoben, wird man in Phase RED weitergeleitet und kann einen neuen Test schreiben.

Dieser Zyklus wird wiederholt bis das Programm vollendet ist. Während des gesamten Zyklus kann Man jederzeit über Menü→Zurück ins Hauptmenü, zum Hauptmenü zurückkehren.

Die Erweiterungen

-Babysteps:

Ist Babysteps eingeschaltet ist die Zeit für die Phasen RED und GREEN limitiert auf die vorher eingestellte Zeit(REFACTOR ist davon nicht betroffen; ein üblicher Wert für die Zeit ist 2-3 Minuten). Sollte der Schritt in dieser Zeit nicht beendet werden, wird der bisherige Fortschritt gelöscht und es wird in den vorherigen Schritt zurück gewechselt.

Dadurch wird das Programmieren in kleineren Schritten weiter gefördert.

-Tracking:

Ist Tracking aktiviert, wird die Zeit gemessen, die in den einzelnen Phasen verbracht wird. Durch einen Klick auf den Button Tracking während der Übung kann man diesen Daten dann in Form eines Diagramms einsehen.

-ATTD:

A steht hier für „Acceptance“(Akzeptanz). Ein Akzeptanztest ist ein übergeordneter Test der die Funktionalität eines gesamten Features des Programms überprüft. Der Akzeptanztest muss vor Bearbeitung der Übung in der Datei der Übung eingepflegt werden. Ist ATTD aktiviert wird der Akzeptanztest in einem extra Fenster angezeigt. Er kann nicht bearbeitet werden. Durch einen Klick auf „ACC-Test“ wird getestet ob der Akzeptanztest erfüllt ist. Ist dies der Fall gilt die Aufgabe als erledigt, ist dies nicht der Fall muss nach dem normalen Muster weiter programmiert werden, bis der Akzeptanztest erfüllt wird.