

# **Test Driven Development Trainer (TDDT)**

Gruppe: UnicornDefenders

# **Inhaltsverzeichnis**

# 1 Einführung

Das sogenannte Test Driven Development (TDD) ist eine spezielle Form der Softwareentwicklung. Dabei werden noch vor dem eigentlichen Code Tests geschrieben, die es schrittweise durch Erweiterungen des Codes zu bestehen gilt. Es soll darauf Acht gegeben werden, dass das Programm gerade nur so viele Erweiterungen erfährt, dass der nächste anstehende Test gerade erfüllt wird.

Auch wenn es auf den ersten Blick aufwändiger erscheint, neben dem Code noch Tests zu schreiben, ist es vorteilhaft, testgetrieben zu programmieren. Zum einen wird nämlich durch die Spezifikation der Tests klarer, was genau man programmieren muss. Zum anderen wird sichergestellt, dass das programmiert wird, was auch gefordert ist, da sonst die Tests fehlschlagen würden. Weiterhin bewirkt der Einsatz dieser Technik, dass das Programm keine unnötigen Bestandteile enthält, weil es in jedem Schritt gerade nur den jeweiligen Test erfüllen muss.

Insgesamt wird also auf diese Weise ein tiefergehendes Verständnis des Programms und der Anforderungen an das Programm sichergestellt.

Über den TDD Trainer (TDDT) wird ein Werkzeug bereitgestellt, um testgetrieben zu entwickeln. Dabei besteht die Möglichkeit, aus einem Katalog an Aufgaben eine herauszuwählen und diese nach dem TDD-Prinzip zu programmieren. Das folgende Nutzerhandbuch soll in die Nutzung des TDDT einführen.

Wir wünschen viel Spaß bei der Arbeit mit dem TDDT und bitten um Entschuldigung, falls sich an der einen oder anderen Stelle längere Ladezeiten einstellen sollten.

## 2 Allgemeine Hinweise zur Nutzung

### 2.1 Grundaufbau

Nachdem Sie die Anwendung starten, öffnet sich ein Fenster. Links finden Sie den Katalog an Aufgaben, aus dem Sie per Mausklick wählen können. In der Mitte finden Sie einen Texteingabe-Bereich, in dem Sie ihren Code schreiben sollen, weiter rechts davon finden Sie einen entsprechenden Texteingabe-Bereich, um die Tests zu schreiben.

Diese Komponenten bilden zusammen den Arbeitsbereich.

Unterhalb des Arbeitsbereichs finden Sie ähnlich wie bei einer IDE einen Bereich für die Ausgabe von etwaigen Fehlermeldungen nach der Kompilierung.

Oberhalb des Arbeitsbereichs finden Sie die Aufgabenstellung der Aufgabe, die Sie ausgewählt haben. Darüber wiederum befindet sich eine Leiste mit drei Buttons, mithilfe derer Sie die Möglichkeiten des Trackings und des Refactoring in Anspruch nehmen können oder unter MyProgramm weitere Aktionen ausführen können, auf die im Folgenden noch Bezug genommen wird.

### 2.2 Aufgabenkatalog

Der weiter oben bereits angesprochene Aufgabenkatalog besteht in einer XML-Datei und kann per MyProgramm→Open nachgeladen werden. Eine Übungsaufgabe besteht dabei aus einer Vorlage für die in der Aufgabe zu schreibende Klasse und einer Vorlage für die Tests. Wenn Sie eine Übungsaufgabe aus der Liste links im Fenster anklicken, erscheint die Aufgabenstellung im Bereich oberhalb des Arbeitsbereichs. Der Spalte können Sie jeweils den Namen der Aufgabe entnehmen sowie feststellen, ob in der Konfiguration der Aufgabe die Option „BabySteps“ eingeschaltet worden ist oder nicht. Was die BabySteps-Option konkret ist, wird weiter unten beschrieben.

### 2.3 Ablauf

Nachdem Sie eine Aufgabe gewählt haben, können Sie bereits mit dem TDD beginnen. Sie treten zuerst in die Phase „Rot“ ein. Hier dürfen Sie nur im Test-Bereich schreiben. Es soll hier ein Test geschrieben werden, sodass mindestens ein Test fehlschlägt oder die Kompilierung nicht funktioniert. Wenn Sie noch keinen Code geschrieben haben, weil Sie gerade erst begonnen haben, schlägt die Kompilierung fehl, sodass obige Bedingung auch dann erfüllt ist. Nachdem Sie einen Test geschrieben oder editiert haben, können Sie nun per MyProgramm→Compile kompilieren.

Wenn nun ein Test oder die Kompilierung fehlschlägt, wird in die Phase „Grün“ übergegangen. Nun kann nur der Code geschrieben werden, die Tests können nicht

mehr modifiziert werden. Es besteht auch die Möglichkeit, in die vorige „Rot“-Phase zu wechseln. Allerdings wird dann der geschriebene Code wieder in den Zustand versetzt, der in der vorigen „Rot“-Phase bestand, d.h. jeglicher Fortschritt geht verloren. Wenn Sie wieder mit MyProgramm→Compile fortfahren, wird der Code kompiliert und getestet.

Falls alle Tests erfüllt werden und das Programm erfolgreich kompiliert, kann über das Anklicken des „Refactoring“-Buttons in der Leiste oben zur Refactor-Phase gewechselt werden. Hierbei können Sie Ihren Code verbessern, wobei auch nach der Änderung alle Tests erfüllt sein müssen. Wichtig: Die Refactor-Phase ist rein optional.

Nachdem alle Tests in der „Grün“-Phase laufen bzw. nach Abschluss eines etwaigen Refactorings wird ein Zyklus beendet und wieder in die „Rot“-Phase gewechselt.

Welchen der beiden Eingabebereiche Sie gerade nutzen können, können Sie den Rändern der Eingabebereiche entnehmen. Wenn Sie nur die Tests editieren können, färben sich die Ränder des für den Code vorgesehenen Eingabebereichs in der Mitte rot und die Ränder des für die Tests vorgesehenen Eingabebereichs rechts im Bild grün. Wenn Sie nur den Code editieren können, verhält es sich genau umgekehrt.

## **3 Optionen**

### **3.1 BabySteps**

Wenn Sie eine Aufgabe anklicken, für die die BabySteps-Option auf „Ja“ gesetzt ist, so haben Sie nur eine begrenzte Bearbeitungszeit für die Phasen „Rot“ und „Grün“, d.h. für das Refactoring besteht auch dann keine zeitliche Begrenzung. Dabei können Sie die gewünschte Bearbeitungszeit in Sekunden in der Konfiguration der Aufgabe nach Belieben wählen, wobei zwei bis drei Minuten empfohlen werden, damit das Lernziel dieser Option nicht außer Acht gelassen wird. Wenn die Zeit abgelaufen ist, wird der jeweilige Test oder Code, der in dieser Phase zusätzlich geschrieben wurde, gelöscht und zur vorherigen Phase gewechselt. BabySteps soll dem Anwender das Programmieren in kleinen Schritten nahe legen und ihn im Hinblick darauf sensibilisieren, dass er sich nur auf das Wesentliche konzentriert.

### **3.2 Tracking**

Wenn Sie den Button mit der Aufschrift „Tracking“ in der Leiste oben anklicken, werden die von Ihnen vorgenommenen Änderungen und der Änderungszeitpunkt aufgezeichnet. Mithilfe dieser Funktion kann der Anwender seinen Arbeitsablauf nachverfolgen, sodass er zum einen anhand der Zeitangaben erfahren kann, wie lange er für welche Phase benötigt hat, und zum anderen mögliche Problembereiche bzw. häufige Fehler ausfindig machen kann.

## **5 Sonderfälle**

Wenn Sie eine Aufgabe ausgewählt haben und dann eine andere Aufgabe aus der Liste links im Bild anklicken, geht jeglicher Fortschritt, den Sie zur erstgenannten Aufgabe erreicht haben, verloren.

## **6 Schlusswort**

Wir hoffen, dass Ihnen die Arbeit mit dem TDDT Spaß bereitet hat und Sie Ihre Programmierfähigkeiten gegebenenfalls erweitern konnten. Für Kritik, Verbesserungsvorschläge oder sonstige Anmerkungen senden Sie uns bitte eine Mail an:

`defenders.unicorns@propra.de`.