

# **TDDT - Test Driven Development Trainer**

## **Was ist TDDT?**

TDDT ist ein Programm zum Aneignen der Methode des **Test Driven Development**, zu deutsch **Testgetriebene Entwicklung**. Die Idee dahinter ist, nicht wie bisher einen Code zu schreiben und ihn daraufhin zu testen, sondern den ersten Test noch vor dem Code zu schreiben.

Man setzt sich dadurch, bereits noch bevor der Code geschrieben wird, ein Ziel, welches der eigene Code erfüllen soll.

Sie stellen dadurch sicher, dass Ihr Code zu jeder Zeit alle Anforderungen erfüllt.

Sobald Sie Ihren Code soweit geschrieben haben, dass ihr Test erfolgreich durchlaufen wird, schreiben Sie den nächsten Test, um somit immer mehr Funktionen und Grenzfälle zu implementieren.

Sinn und Zweck ist im Allgemeinen in kleinen Schritten zu arbeiten, da so Fehler gleich zu Beginn gefunden und ausgemerzt werden können. Man durchläuft so einen Zyklus, der das Schreiben von Tests und das implementieren dieser im Code wiederholt.

Nachdem Sie den Code geschrieben haben, können Sie diesen in einer Optimierungsphase verbessern, ohne ihn inhaltlich zu ändern. Dies sorgt zuletzt für einen funktionierenden, aber dennoch eleganten Code.

## **Das Programm**

### **Wahl des Workspace**

Sobald Sie das Programm gestartet haben, erscheint ein Fenster, in welchem Sie den Ordner festlegen, der Ihre gespeicherten Daten enthalten soll, genannt Workspace.

Klicken Sie dazu auf die Fläche Browse und wählen Sie Ihren gewünschten Ordner.

Bestätigen Sie Ihre Wahl anschließend mit OK.

Beim erneuten Start des Programmes können Sie einen anderen Workspace festlegen.

### **Katalog laden**

Um das Programm nutzen zu können, benötigen Sie einen Katalog, welcher Ihre Aufgaben enthält. Damit Sie einen solchen nutzen können, muss dieser zunächst geladen werden.

Klicken Sie dazu auf das Feld Start links oben. Es erscheint nun eine Auswahl mit Katalog Speichern und Katalog Laden.

Wählen Sie nun Katalog laden und navigieren Sie in den Ordner, in dem sich Ihr Katalog befindet. Bestätigen Sie zuletzt ihre Auswahl.

### **Eine Aufgabe beginnen**

Nachdem ein Katalog geladen wurde, kann nun eine Aufgabe ausgewählt werden, welche Sie bearbeiten möchten. Im Fenster können Sie sich weiterhin eine Beschreibung zur Aufgabe sowie weitere Details ansehen.

Zum Starten einer Aufgabe klickt man mit Hilfe eines Doppelklicks auf die gewünschte Aufgabe im geladenen Katalog. Wenn Sie bereits eine Aufgabe laufend hatten, werden Sie danach gefragt, diese zu speichern.

## **Eine Aufgabe bearbeiten**

### **Status**

Sie beginnen im Status Rot. Dort schreiben Sie einen fehlschlagenden Test oder überarbeiten Ihren Code erneut, falls er nicht compiliert.

### **Einen Test schreiben**

Um das Prinzip des TDD zu erfüllen, schreiben Sie zunächst einen Test, der sich zum Ziel setzt, einen Teil Ihres späteren Codes zu prüfen und somit sicherzustellen, dass ihr Code am Ende alle Anforderungen besteht. Versuchen Sie dabei sich beim Testen möglichst kleine Ziele zu setzen.

### **Der Code compiliert nicht**

Wenn Ihr geschriebener Code nicht compilieren sollte, befinden sie sich weiterhin im Status Rot und müssen Ihren fehlerhaften Code noch einmal überarbeiten.

Sie befinden sich nun in Status Grün.

### **Code verändern**

Sie haben nun die Möglichkeit Ihren Programmcode so zu verändern, dass er alle Tests besteht.

### **Zu Rot zurück**

Alternativ können Sie sich zurück in den Status Rot bewegen, um ihre Tests zu überarbeiten. Beachten Sie dabei, dass ihr Code gelöscht wird, sofern Sie dies tun.

### **Optimierung**

Wenn ihr Code alle Tests besteht und erfolgreich compiliert, können Sie den Optimierungsmodus wählen. Dort können Sie Ihren Code aufbessern und komprimieren. Beachten Sie jedoch, dass alle Tests nach der Optimierungsphase weiterhin bestanden werden müssen. Es ist daher hilfreich, darauf zu achten, inhaltlich nichts oder möglich wenig am Code zu ändern.

## **Zusätzliche Funktionen**

### **1. Babysteps**

#### **Was ist Babysteps?**

Diese Funktion können Sie aktivieren, um ihre Bearbeitungszeit in den Phasen Rot und Grün einzugrenzen.

Dies sorgt dafür, dass während des Programmierens ein Timer läuft, der nach Ablauf Ihren neuen Code, beziehungsweise Test, löscht und zum vorherigen Status zurückwechselt.

### **Wofür verwendet man diese Methode?**

Babysteps sind besonders für diese angehenden Programmierer sinnvoll, welche zu viele Schritte gleichzeitig abarbeiten wollen. Es ist sinnvoll seinen Code mittels Tests zu validieren, weshalb es besonders für Neulinge besonders nützlich ist, seinen Code mit vielen Tests abzudecken. Wichtig ist dabei, dass jeder Test einen anderen Grenzfall abdeckt, um so sicherzustellen, dass man nach jedem Testdurchlauf mit seinem Code in kleinen Schritten richtung Ziel steuert.

## **2. Akzeptanztests - ATDD**

### **Was ist ein Akzeptanztest?**

Ein Akzeptanztest ist ein gutes Mittel zur Übersicht über den eigenen Code. Ein Akzeptanztest gibt an, ob sie ein Feature vollständig implementiert haben. In diesem Fall wird der Test grün.

### **Wie verwende ich Akzeptanztests?**

Ihre Arbeitsweise ändert sich im Vergleich zur Standard TDD-Variante nun geringfügig. Bevor Sie mit Ihrem gewohnten TDD-Zyklus beginnen, schreiben Sie zunächst einen Akzeptanztest, welcher prüft, ob ein bestimmtes Feature vollständig implementiert ist.

Der Akzeptanztest wird daher erst grün, wenn dieses bestimmte Feature vollständig ist.

Sie schreiben erst den nächsten Akzeptanztest, wenn ihr Aktueller grün ist.