

Nutzerhandbuch

Die Anwendung TDDT (Test Driven Development Trainer) soll Studierenden helfen, die Technik der testgetriebenen Entwicklung zu üben.

Bei der testgetriebenen Entwicklung handelt es sich um eine Technik zur Softwareentwicklung, bei der ein Test vor dem zu testenden Code geschrieben wird.

Das Szenario soll in etwa so ablaufen :

1. Der Benutzer wählt aus einem Katalog von Übungsaufgaben eine Aufgabe aus.
2. **RED** Das Werkzeug erlaubt es dem Nutzer nur den Test zu editieren, bis es **genau einen fehlschlagenden Test gibt** oder der Code nicht compiliert. Der Nutzer soll dem Programm mitteilen, dass er bereit ist für den nächsten Schritt. **Der Wechsel zum nächsten Schritt darf aber nur erfolgen, wenn die Bedingung erfüllt ist.**
3. **GREEN** Das Programm erlaubt es dem Nutzer nun ausschließlich den Code zu modifizieren, bis alle Tests laufen. Alternativ kann der Benutzer zurück zu **RED** wechseln. **Beim Zurückwechseln zu den Tests wird der neue Code gelöscht.**
4. **REFACTOR** Nachdem alle Tests laufen, darf der Nutzer den Code verbessern (refactoring). Der Wechsel von **GREEN** nach **REFACTOR** muss der Benutzer dem Programm explizit mitteilen. Es ist nur erlaubt zu wechseln, wenn die Bedingungen **(alles kompiliert und alle Tests laufen durch)** erfüllt sind. **Wichtig ist aber das Refactoring immer bedeutet, dass alle Tests immer vorher und nachher laufen.**
5. Wenn der Nutzer die Verbesserungsarbeiten beendet hat, beginnt der Prozess wieder bei Schritt 2 (**RED**).

Programmablauf:

- Zum starten des Programms muss das Projekt kompiliert und die Klasse MainApp ausgeführt werden.
- Über die Menüleiste am oberen Rand muss eine Übung aus dem bereitgestellten Katalog ausgewählt werden.
- Die Übung wird in den Arbeitsspeicher geladen, die aktuelle Datei im Textfeld, aller Übungsaufgaben links in einem TreeView und Beschreibung, sowie weitere Buttons, um in nächste Phasen zukommen rechts in einer Pane.
- Nachdem genau ein fehlschlagender Test geschrieben wurde (zum Testen der Kompilierung (ohne die Phase zu wechseln), den Compile Button drücken), kann in die nächste Phase (**GREEN**) gewechselt werden.
- Nachdem der dazugehörige Code geschrieben wurde, kann entweder, bei bestehen aller Tests, in die **REFACTOR** Phase, oder unter löschen des bereits geschriebenen Codes zurück zu **RED** gewechselt werden. (Dazu die jeweiligen Buttons klicken)
- In der **REFACTOR** Phase wird der "weiter zu RED" Button enabled, der bei erfolgreichen Kompilieren eine weitere Iteration des Programmablaufs einleitet.

Zudem kann jederzeit oben in der Menüleiste der aktuelle Stand im Arbeitsspeicher auf die Festplatte geschrieben werden.

Das Programm enthält zwei Features :

- Babysteps
 - implementiert über die xml Datei der jeweiligen Übung
- Tracking
 - implementiert über einen Button, der die Daten graphisch darstellt

Babysteps:

Wenn für eine Übung Babysteps eingeschaltet sind, wird die Zeit, die der Nutzer in den Phasen **RED** und **GREEN** hat, limitiert. Ist die Zeit abgelaufen, wird der neue Test/Code gelöscht und es wird in die vorangegangene Phase zurückgewechselt. Babysteps haben den Sinn dem Nutzer die Entwicklung in kleineren Schritten nahezulegen/anzutrainieren.

Tracking:

Die Tracking Funktion zeichnet auf, was der Benutzer wann geändert hat. Tracking soll erlauben herauszufinden für welche Aktivitäten der Nutzer viel Zeit benötigt und welche Fehler auftreten. Die Daten können analysiert werden um Probleme zu erkennen in die der Nutzer häufig laufen.