

1.0) Menu

Nach dem Programmstart beginnen sie automatisch im Menu.

Hier werden sie dazu aufgefordert, eine Preset- und eine Codedatei mit einem jeweiligen Klick auf "Load presetfile" bzw. "Load codefile" anzugeben.

Beiden müssen hierbei in Form einer .txt Datei vorhanden sein.

Sollten sie keine Presetdatei erhalten haben, können sie sich im menu eine eigene Presetdatei erstellen. (siehe "Presetdatei erstellen")

Es ist unbedingt notwendig, dass sie eine Presetdatei(geladen oder eigens angelegt) und eine Codedatei angegeben haben.

Erst dann wird der "Continue" Button freigeschaltet um fortzufahren.

Ist ATDD aktiviert, gehen sie weiter zu "ATDD", ist es deaktiviert fahren sie fort mit "Hauptprogramm".

1.1) Presetdatei erstellen

Um eine Presetdatei zu erstellen geben sie im Menu folgende Dinge an:

Classname:

Geben sie hier den Klassennamen ihrer Codevorlage an.

Testname:

Geben sie hier den gewünschten Klassennamen der Testdatei an, welche sie während des Programmablaufs erstellen werden.

Babysteps:

Falls sie ein Zeitlimit für je die Test- bzw. Codephase einstellen wollen, aktivieren sie den "Babysteps"-Schalter und geben sie in die folgenden Schriftfelder die Minuten und Sekunden der Zeitbegrenzung an.

ATDD:

Wollen sie zusätzlich zu den normalen Test bestimmt Features extra testen, aktivieren sie den "ATDD" Schalter.

Haben sie all ihre Eingaben getätigt, betätigen sie final den "Create preset"- Schalter. Hierdurch wird ihre eingene Presetdatei angelegt.

Hinweis:

Nach dem Erstellen einer Presetdatei ist es nicht notwendig diese mit Hilfe des "Load presetfile" Buttons anzugeben.

2.0) Hauptprogramm

Im Hauptprogramm der Software befindet sich Links und Rechts jeweils ein Textfeld.

Das linke Feld dient zum schreiben von Unit-Tests, das rechte Feld zum schreiben des dazu gehöri-gen Codes.

Zusätzlich befindet sich in der Mitte des Bildschirms ein Kreis, unterteilt in drei Teile, welcher als indikator dessen dient, in welcher Phase des Programms sie sich befinden. Unter diesem befinden sich ein "Check"- und ein "Back"-Button und eine Textanzeige mit dem Titel "Terminal".

Zu Beginn ist nur das Test-Textfeld beschreibbar.

Schreiben sie in dieses nun zuerst einen fehlschlagenden Unit-Test.

Schließen sie ihre Eingabe mit einem Klick auf den "Check"-Button.

Sollte beim Compilieren ihres Tests ein Fehler auftreten wird ihnen dieser in der "Terminal" Textanzeige ausgegeben und sie dürfen diesen korrigieren.

Compiliert ihr Test fehlerlos wird das Code-Textfeld aktiviert und kann nun von ihnen beschrieben werden.

Geben sie hier ihren Code ein, welcher den zuvor geschriebenen Test besteht.

Schließen sie ihre eingabe durch erneuten Druck auf "Check" ab.

Compiliert ihr Code fehlerlos und besteht den Test, beginnt das Refactoring.

In dieser Phase haben sie die gelegenheit, sowohl ihren Test als auch ihren Code so anzupassen, dass sie möglichst viel repetitiven Code durch intelligenteren, knapperen Code zu ersetzen.

Diese Phase können sie allerdings nur dann durch einen erneuten Druck auf "Check" verlassen, wenn weiterhin alle Tests bestanden werden.

Nach dieser Phase beginnt der Zyklus erneut und sie müssen einen neuen, fehlschlagenden Test schreiben.

Hinweis:

Ist "Babysteps" aktiviert, haben sie während des Test- und Codeschreibens jeweils nur begrenzt Zeit, bis das geschriebene "eingecheckt" und korrekt kompiliert sein muss.

Andernfalls wird der Code auf den Stand zurückgesetzt, in dem er sich zu beginn der jeweiligen Phase befand.

2.1) ATDD

Durch "ATDD" haben sie die möglichkeit, bestimmte Features zu programmieren, welche mehrere Zyklen des Hauptprogramms benötigen.

Ist in der Presetdatei "ATDD" aktiviert, starten sie nicht direkt mit dem Hauptprogramm, sondern in einem Fenster in dem sich ein Textfeld und ein "Terminal" befindet.

Hier können sie nun einen fehlschlagenden Unit-Test, welcher ihr programm auf ein bestimmtes Feature testet, implementieren und "einchecken".

Um ihren bisher zur verfügung stehenden Programmcode einzusehen, können sie am oberen Rand des Fensters zwischen den Tabs "ATDD" und "Written Code" hin und her wechseln.

Compiliert ihr Testcode korrekt, und genau ein ATDD-Test schlägt fehl, fährt die Software mit dem Hauptprogramm fort.

Die "ATDD"-Phase wird jedes mal vor der Testphase ausgelöst, wenn kein ATDD-Test mehr fehlschlägt.