

## Projekt #7

TDDT (Test Driven Development Trainer)

package Logik

### Zuständigkeit

Die Klasse Logik beinhaltet den Testzyklus, der die einzelnen Phasen **RED**, **GREEN** und **REFACTOR** enthält, wobei **REFACTOR** in zwei Schritte unterteilt werden kann. Sie ist zuständig für den Zyklusablauf und unterrichtet die grafische Benutzeroberfläche über den aktuellen Schritt, damit sie entsprechende Eingaben zulassen oder verweigern kann. Die Klasse Logik enthält selbst keine Ausführungsschleife, sondern wird von der grafischen Oberfläche aufgerufen, den Test durchführen zu lassen und dann gegebenenfalls einen Zyklusschritt weiter zu gehen. Dabei besteht die Möglichkeit, einen zweiten **REFACTOR**-Schritt zuzulassen, in dem auch die Test optimiert werden dürfen. Auch das Ausführen eines Abschlusstests, der bestanden werden muss, ist möglich. Dazu werden der Klasse Logik Instanzen der Klassen **Testen** und **Konfig** übergeben, die dann aufgerufen werden können. Aus **Konfig** werden die relevanten Einstellungen ausgelesen, **Testen** führt die Tests aus.

Die Klasse LogikRueckgabe verwaltet die Rückgabe der Klasse Logik an die grafische Benutzeroberfläche. Dazu stellt sie zunächst die Funktion **wurdeBestanden** zur Verfügung, nach deren Rückgabewert Textfelder gesperrt oder freigegeben werden können.

### Methoden

```
class Logik
void Logik(Tester tester, Konfig konfig)
LogikRueckgabe weiter()
```

```
class LogikRueckgabe
void LogikRueckgabe(String rueckgabe)
boolean wurdeBestanden()
```

## Interaktion mit anderen Klassen

Die hier angegebenen Interfaces sind nur Vorschläge und Anregungen für den Zuständigen des entsprechenden Packages. Die Namen dienen dabei nur der eindeutigen Kennzeichnung der Funktion und sind nicht als Vorgaben zu verstehen.

### `interface GUI`

Diese Klasse soll Logik initialisieren und durch Aufrufe von `weiter()` den Programmfluss fortsetzen. Dabei wird stets eine Struktur zurückgegeben, welche der grafischen Benutzeroberfläche den aktuellen Zyklusschritt und den letzten Testdurchlauf informiert. Logik bekommt bei der Initialisierung eine Instanz von `Konfig` übergeben, aus der Einstellungen ausgelesen werden können.

### `interface Tester`

Diese Klasse soll für Logik einen Test ausführen und das Ergebnis in einer Instanz von `TesterRueckgabe` zurückgeben.

### `interface Tester`

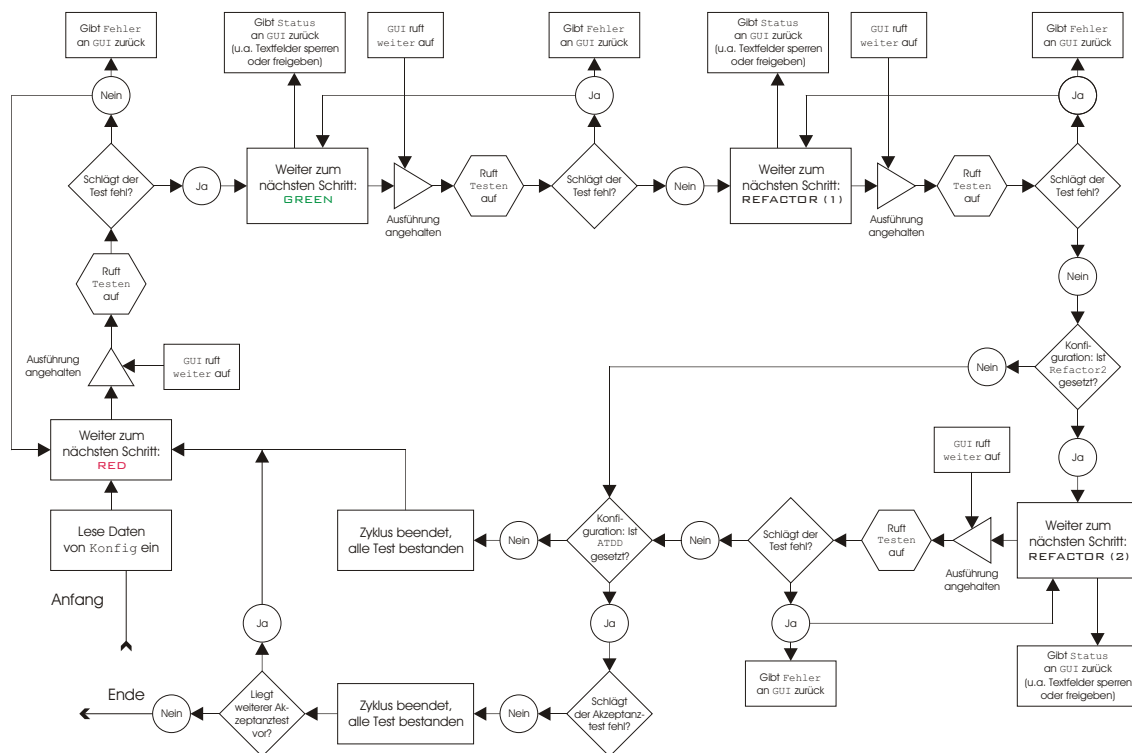
Diese Klasse verwaltet den Rückgabewert, den `Tester` nach Ausführung eines Tests zurückgibt. Er enthält Werte, aus denen ablesbar ist, ob der Test bestanden wurde oder nicht. Auch Fehler bei der Kompilierung oder Ausführung des Programms werden davon erfasst.

### `interface Konfig`

Diese Klasse soll für Logik eine Funktion `Einstellung` zur Verfügung stellen, die einen Eigenschaftennamen übergeben bekommt und den zugehörigen Wert zurückgibt. So wird beispielsweise die Funktion mit `ATDD` aufgerufen und gibt einen Wert zurück, aus dem ausgelesen werden kann, ob `ATDD` gesetzt ist oder nicht.

## Flussdiagramm

Das folgende Flussdiagramm veranschaulicht die Funktionsweise eine Instanz von Logik. Nach der Initialisierung werden die relevanten Einstellungen aus der Konfigurationsdatei geladen. Diese sind ATDD und Refactor2. Anschließend wird jeder Zyklus durchlaufen, indem der GUI die Datenstruktur ausgegeben wird, mit der sie dann Textfelder sperrt oder freigibt. Sie ruft nach erfolgter Eingabe des Nutzer die Funktion weiter() der Klasse Logik auf, die dann Tester aufruft und das Ergebnis in einer Datenstruktur erhält, die bei einem Fehlschlag an GUI zurückgegeben wird. Bei Erfolg wird zusätzlich der nächste Zyklusschritt durchgeführt. Wurde der Schritt REFACTOR(1) abgeschlossen und der Wert Refactor ist nicht gesetzt und REFACTOR(2) wird übersprungen. Falls ATDD gestetzt ist, wird nach beendigung des Zyklus noch der Akzeptanztest durchgeführt. Ist danach kein weiterer Akzeptanztest vorhanden, wird der Entwicklungszyklus beendet. Ist ATDD nicht gesetzt, wird stets zum ersten Zyklusschritt RED gesprungen.



## **Geplante Änderungen**

Es soll eine weitere Klasse `TestAuswertung` geben, die `TesterRueckgabe` in `LogikRueckgabe` konvertiert und dabei die Konsolenausgabe auswertet. Für die Funktionalität von *BabySteps* müssen noch weitere Funktionalitäten geschaffen werden. In diesem Bericht sollen die Klassen besser getrennt werden.

## **Änderungsprotokoll**

21.06.2016 Das Dokument wurde erstellt