Prof. Dr. Stefan Betermieux
Fakultät Informatik
Hochschule Furtwangen

Open Source-basierte Softwareentwicklung

Praktikum 6 - Softwaretests

Szenario

In dieser Woche lernen Sie die ersten Schritte mit dem Test-Werkzeug »JUnit 4«. Zu diesem Zweck verwenden wir das im letzten Praktikum angelegten Maven-Projekt.

Arbeitsschritte

Schritt 1

Öffnen Sie Ihr in der letzten Woche angelegtes Maven-Projekt in Eclipse. Vergewissern Sie sich, dass es von Eclipse als…

- …Maven-Projekt erkannt wird (kleines blaues M links oben im Projekt-Icon)
- …Java-Projekt erkannt wird (kleines blaues J rechts oben im Projekt-Icon)
- …Git-Projekt erkannt wird (kleine gelbe Tonne rechts unten im Projekt-Icon)

Wenn eins der Icons nicht angezeigt wird, müssen Sie das Projekt für die nächsten Schritte in diesen Ausgangszustand versetzen.

Schritt 2

Überprüfen Sie, ob bei Ihnen JUnit 4 verwendet wird. Dazu muss in der pom.xml in der Abhängigkeit für Junit die Versionsnummer mit 4.XX anfangen. Wenn dort noch 3.XX steht, dann ändern Sie den Eintrag entsprechend ab:

```
<groupId>junit</groupId>
<artifactId>junit</artifactId>
<version>4.12</version>
```

Schritt 3

Laden Sie von http://webtech.informatik.hs-furtwangen.de/oss/junit/ die Dateien `Queue.java` und `Util.java` herunter. Integrieren Sie die Dateien in Ihr Projekt unter `src/main/java`. Passen Sie die Package-Struktur bei Bedarf an. Eclipse zeigt keine Compiler-Fehler an.

Schritt 4

Schreiben Sie einen JUnit-Test für die Methode `Util.istErstesHalbjahr(int monat)`. Führen Sie den Test durch das Ausführen von »`mvn test`« aus und korrigieren Sie evtl. vorhandene Fehler!

Schritt 5

Schreiben Sie einen JUnit-Test für die Klasse Queue (die formalen Anforderungen an die Queue stehen am Ende dieses Schritts). Nehmen Sie für den Test eine Arraylänge von 3 für die Queue an. Konstruieren Sie die Testfälle für einen zustandsbezogenen Test. Führen Sie die Tests wieder mit »`mvn test`« aus und korrigieren Sie evtl. vorhandene Fehler!

Anforderungen an die Queue:

> Es wurde eine Queue (Warteschlange) auf Basis eines Arrays implementiert. Die Queue unterstützt die beiden Funktionen `enqueue` und `dequeue`. Mit `void enqueue(int i)` wird die Zahl `i` in die Queue aufgenommen. Wenn die Queue voll ist, dann wird der Wert am Ende der Queue einfach überschrieben. Mit `int dequeue()` wird die Zahl, die am längsten in der Queue ist, logisch aus der Queue genommen und zurückgeliefert. Wenn die Queue leer ist, dann ist die Operation `dequeue` nicht erlaubt und es wird ein Fehler signalisiert. Damit der Speicherplatz im Array möglichst gut genutzt wird, ist die Queue in Form eines logischen Ringspeichers organisiert. Wenn das Ende des Arrays erreicht ist, dann wird vorne im Array weitergearbeitet, wenn sich dort nicht der Anfang der Warteschlange befindet. Ansonsten wird der Wert am Ende der Queue überschrieben.

Schritt 6

Führen Sie einen JUnit-Test in Eclipse aus, indem Sie mit der rechten Maustaste auf die Testklasse klicken und Run as…/JUnit Test auswählen. Der grüne Balken sollte in der View »JUnit« erscheinen. In der Listenansicht können Sie die Testergebnisse detailliert betrachten.

Schritt 7

Installieren und verwenden Sie das Eclipse-Plugin »Infinitest« in Ihrem Projekt (Marketplace Infinitest suchen und installieren).

Schritt 8

Laden Sie Ihre Änderungen in das entfernte Git-Repository hoch.