



Testen mit JUnit

Florian Pix

18.04.2019

SWT Übung SoSe19

Anleitung der Fakultät zum Einstieg in JUnit

1. Testfalltabellen
2. Fixtures
3. Testrunner
4. Testsuite
5. JUnit 3 und 4

Testfalltabellen

Bestimme die Zustände:

in denen sich ein Objekt
nach einer Anweisung befinden muss

Bestimme die Extremwerte:

der Parameter der zu testenden Methode
z.B. 0 oder null

Bestimme die Randwerte:

der Parameter der zu testenden Methode
z.B. 1.1. und 31.12.

Testfalltabelle für die Methode `grow(int move)` aus einem Snakespiel

Testfall	Erwarteter Status	Klasse	Parameter	Erwartete Ausgabe
S1	Exception	Snake	-1	Eine Schlange kann nicht schrumpfen.
S2	ok	Snake	1	2
S3	ok	Snake	5	6

Fixtures

Eine Fixierung ist ein fixer Status von Objekten der als Grundlagen des Durchlaufen von Tests genutzt wird.

Der Sinn einer solchen Fixierung ist es sicher zu stellen dass es eine bekannte Umgebung gibt in der Resultate reproduzierbar sind.

Beispiel:

Vorbereitung von Input Daten:

- Erstellung von Mock-Objekten

- Laden einer Datenbank mit bekannten Datensatz

Dazu muss ebenfalls sichergestellt werden dass bei einem Test die Umgebung nicht so verändert wurde dass andere Tests beeinflusst werden könnten.

JUnit 3.8.4

setUp()

tearDown()

Testrunner

Testeinstieg/Main-Methode

Beispiele:

junit.textui.TestRunner

junit.swingui.TestRunner

Testsuite

Testsuite

Wenn wir viele Tests auf einmal ausführen wollen können wir diese in einer Testsuite zusammenfassen.

```
1 TestSuite suite= new TestSuite();  
2 suite.addTest(new FirstTest("testEquals"));  
3 suite.addTest(new FirstTest("testSimpleAdd"));  
4 TestResult result= suite.run();
```

Oder:

```
1 TestSuite suite= new TestSuite(FirstTest.class);  
2 TestResult result= suite.run();
```

JUnit 3 und 4

JUnit 3 und 4

Feature	JUnit 3.x	JUnit 4.x
test annotation	testXXX pattern	@Test
run before the first test method in the current class is invoked	None	@BeforeClass
run after all the test methods in the current class have been run	None	@AfterClass
run before each test method	override setUp()	@Before
run after each test method	override tearDown()	@After
ignore test	Comment out or remove code	@Ignore
expected exception	catch exception assert success	@Test(expected = ArithmeticException.class)
timeout	None	@Test(timeout = 1000)

Quelle: "asjava.com"

Hilfe zu JUnit4