JUnit

Stefan Wehr
Prof. Dr. Peter Thiemann

21. Dezember 2005

Unit Tests

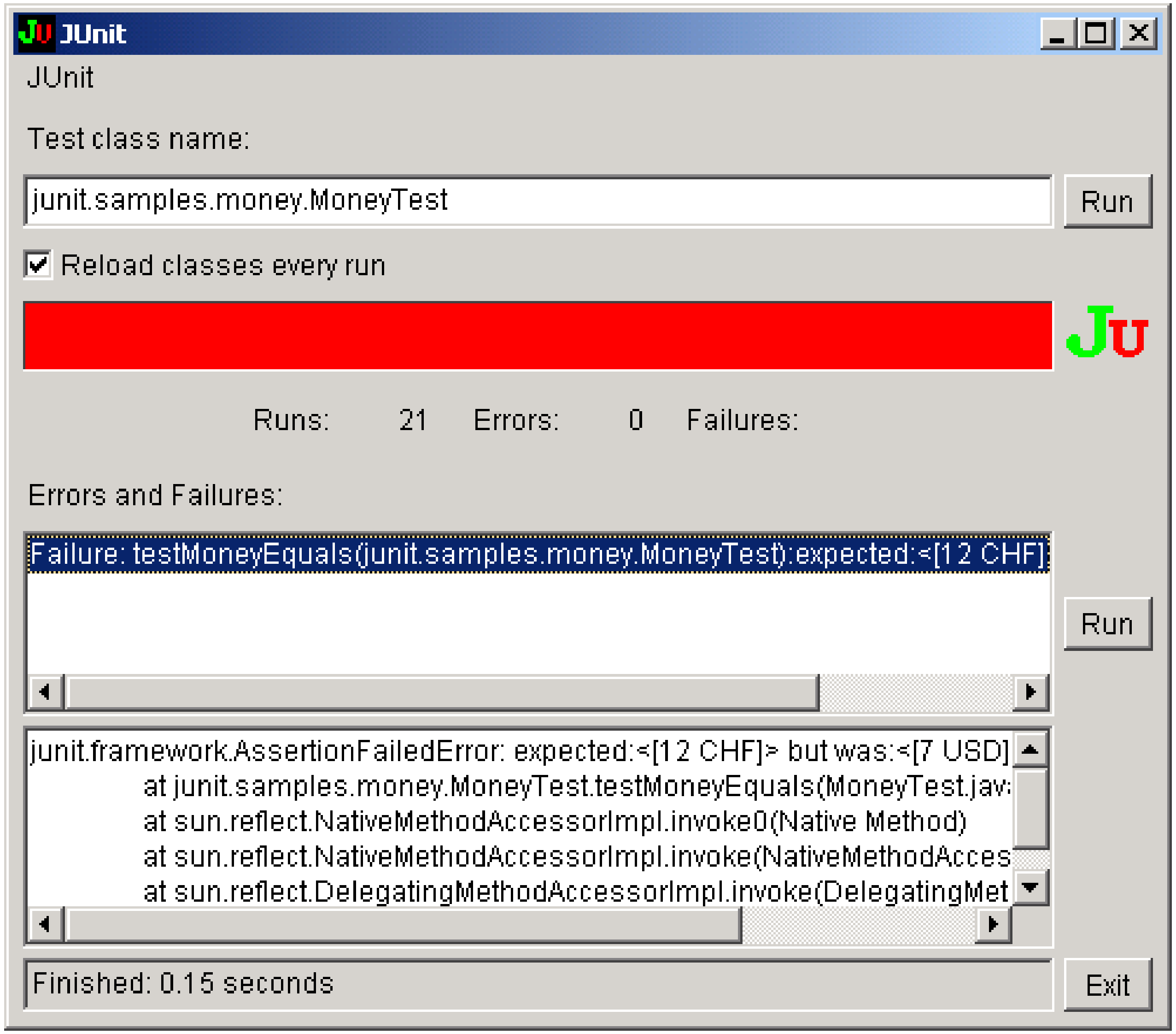
Idee

- Schreibe zu jeder Klasse eine Testklasse.
- Archiviere alle Tests, so dass sie später wiederholt werden können.

Nutzen

- Software enthält weniger Fehler
- Software kann leichter geändert werden
- Testklasse dient als eine Art Spezifikation für die eigentliche Klasse

JUnit


JUnit
JUnit
Test class name:
junit.samples.money.MoneyTest
Run
Reload classes every run
Runs: 21 Errors: 0 Failures:
Errors and Failures:
Failure: testMoneyEquals(junit.samples.money.MoneyTest):expected:<[12 CHF]
Run
junit.framework.AssertionFailedError: expected:<[12 CHF]> but was:<[7 USD]
at junit.samples.money.MoneyTest.testMoneyEquals(MoneyTest.jav
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAcces
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMet
Finished: 0.15 seconds
Exit

Assertions (deutsch: *Aussage, Behauptung*)

Vergleichen erwartetes mit wirklichem Verhalten.

Wichtige Assertions

- `assertEquals(Object expected, Object actual)`
- `assertEquals(double expected, double actual, double delta)`
- `assertSame(Object expected, Object actual)`
- `assertTrue(boolean b)`
- `assertFalse(boolean b)`
- `assertNull(Object actual)`
- `assertNotNull(Object actual)`
- `fail(String message)`

Beispiele für Assertions

- `assertNotNull(myObject)`
  schlägt fehl falls `myObject` null ist.
- `assertEquals(1, x)`
  schlägt fehl falls `x` nicht den Wert `1` hat.

Assertions für Ausnahmen

- Erwartete Ausnahmen:

```
try {
  new File("/foo/bar");
  fail("FileNotFoundException expected!");
} catch (FileNotFoundException ex) {
  // ok, exception expected
}
```

- Unerwartete Ausnahmen werden von JUnit behandelt

Tests in JUnit

Ein Test (Interface `Test`) in JUnit ist

- ein einzelner Testfall (Klasse `TestCase`) oder
- eine Sammlung von Tests (Klasse `TestSuite`).

Mit Hilfe von Eclipse kann man Tests bequem ausführen.

Beispiel für TestCase

```
public class EuroTest extends TestCase {
  private Euro two;
  // setUp wird vor Ausführen jeder Testmethode aufgerufen.
  protected void setUp() {
    two = new Euro(2.0);
  }
  public void testAdding() {
    Euro sum = two.add(two);
    assertEquals(new Euro(4.0), sum);
  }
  public void testFromDollar() {
    Euro one = Euro.fromDollar(1.2);
    assertEquals(new Euro(1.0), one, 0.001);
  }
}
```

Beispiel für TestSuite

```
public class AllTests {
  public static Test suite() {
    TestSuite suite = new TestSuite();
    suite.addTestSuite(EuroTest.class);
    suite.addTest(TestsForSubcomponent.suite());
    return suite;
  }
}
```

Success, Failure und Error

Mögliche Ergebnisse eines Tests:

Success Alle Assertions erfolgreich.

Failure Mind. eine Assertion ist fehlgeschlagen.

Error Testmethode hat (unerwartete) Ausnahme verursacht.