Web-Testen mit JUnit und HttpUnit

Kai Schmitz-Hofbauer
Lehrstuhl für Software-Technik
Ruhr-Universität Bochum

Inhalt

- Entwicklertests in der Praxis
- Unit-Testing
- JUnit
- HttpUnit
- Praktisches Beispiel
- Bewertung
- Literatur

Entwicklertests in der Praxis

- Häufige Praxis beim Testen von Software
  - Verwendung von adhoc-Code
    - Testausgaben in der Form `if (debug) System.out.println("OK");`
    - Diese Art Testtreiber und Testumgebung sind in der Regel temporär ("OK"?, dann löschen)
  - Testfälle werden kaum dokumentiert und sind nur schwer reproduzierbar
  - Fehlschlag oder Erfolg eines Tests erschließt sich nur dem Eingeweihtem (Programmierer)
  - Hoher Aufwand durch häufiges Neuerstellen von Umgebung und Treibern

Unit-Testing

- Bei umfangreichen Softwareprodukten ist es nicht sinnvoll, erst das fertige Produkt zu testen
- Es ist sinnvoll, einzelne abgeschlossene Einheiten des Gesamtprojektes (Module oder Units) zu testen Þ ***Unit-Testing***
- Bei diesem Vorgehen können Fehler leichter lokalisiert und korrigiert werden

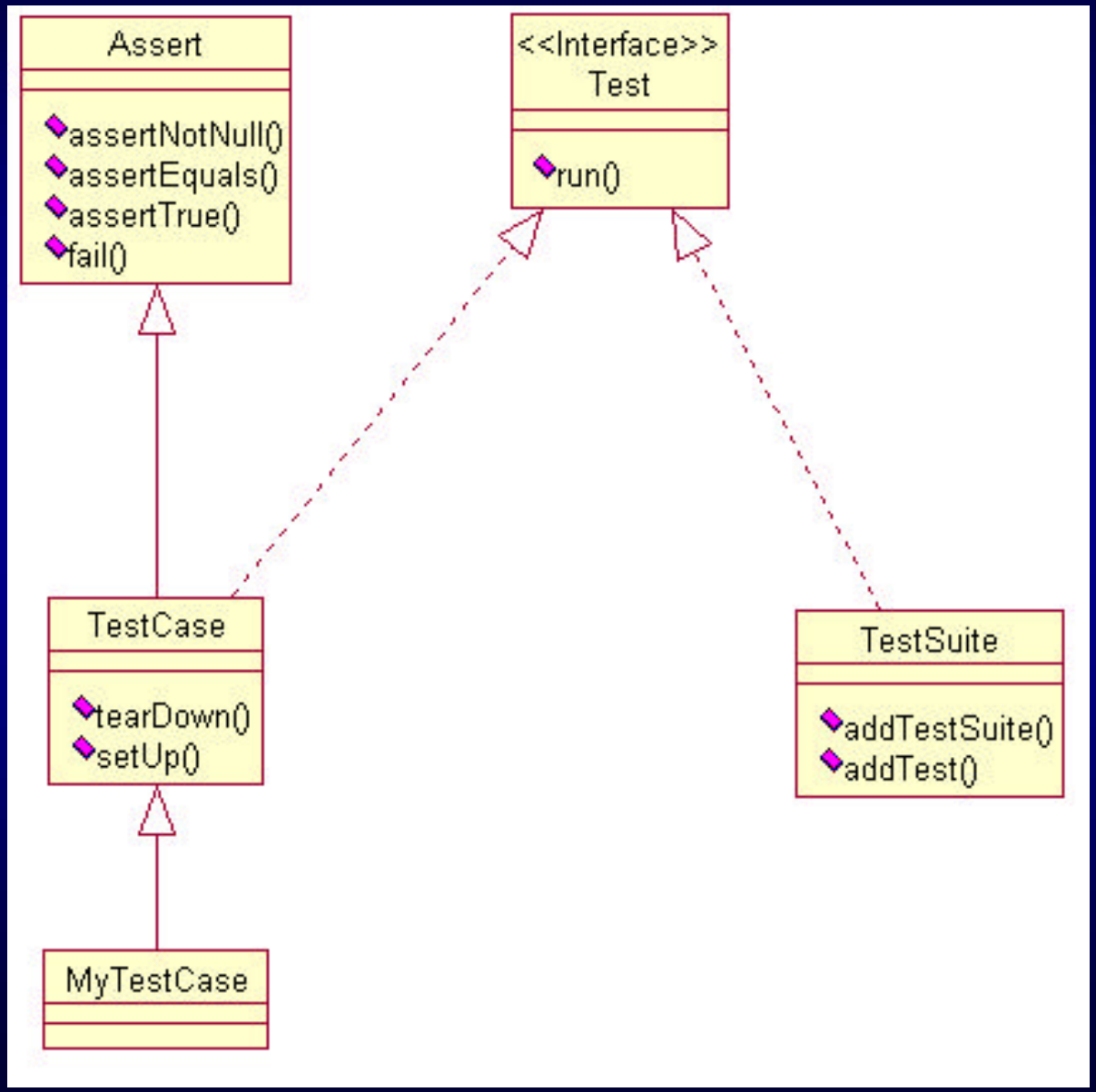
JUnit

- JUnit ist ein Java-Framework zum Ausführen **automatisierter** Unit-Tests
- Testfälle werden in Java codiert, compiliert und dann ausgeführt
- JUnit stellt zur Unterstützung Java-Klassen und Schnittstellen zur Verfügung
- eignet sich zum Testen von Java-Klassen
  - im Bereich Web-Testen daher gut geeignet zum Testen von JavaBeans

JUnit

- Black-Box-Testverfahren
- Der Einsatz von JUnit ist in erster Linie für den Entwickler gedacht und nicht für für das Qualitätsmanagment
- Ständiges Ausführen dieser Tests schafft Vertrauen in den eigenen Code
  - Erneutes Ausführen aller Tests bei Änderungen am Code
  - Seiteneffekte einer Änderung werden sofort erkannt

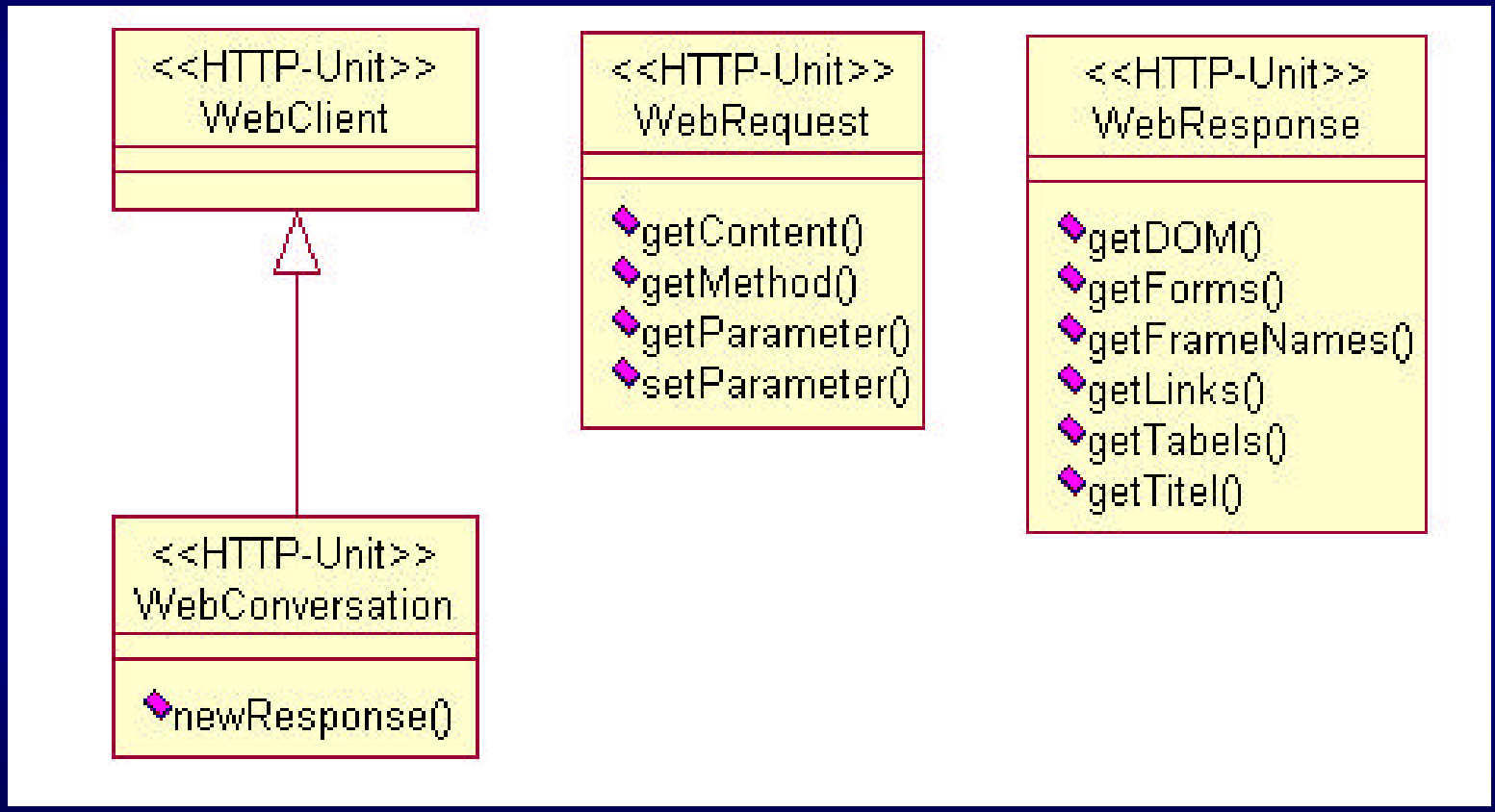
Die wichtigsten Klassen von JUnit



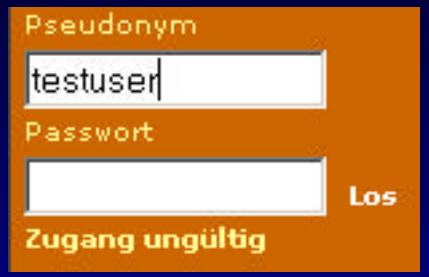
HttpUnit

- Üblicherweise greift man auf eine Webseite mit einem Browser zu
- Dieses Vorgehen ist für eine automatisierte Testdurchführung ungeeignet
- Vielmehr ist es wünschenswert, auf eine Webseite direkt aus einem Programm, wie z. B. einem JUnit-TestCase, zuzugreifen
- HttpUnit stellt diese Funktionalität zur Verfügung
- HttpUnit ist ebenfalls in Java geschrieben und erweitert JUnit um die Fähigkeit, Webseiten (z. B. JSPs) zu testen

Die wichtigsten Klassen von HttpUnit



Testen des W3L-Login-Frames




```
import junit.framework.*;
import com.meterware.httpunit.*;
public class LoginTest extends TestCase
{
  private static final String url= "http://mars.swt.rub.de/w3l/jsp/startportal.jsp

  public void setUp() { createTestUser("testuser","123456"); }
  public void tearDown { deleteTestUser("testuser"); }

  public void testValidLogin()  throws Exception { ....}
  public void testInvalidLogin()  throws Exception { ....}

 public static void main (String [] args )  {
     junit.textui.TestRunner.run(suite());   }

  public static TestSuite suite()   {
     return new TestSuite(LoginTest.class);  }
}
```

Testen des W3L-Login-Frames

```
public void testInvalidLogin()  throws Exception
{
    WebConversation conversation = new WebConversation();
    WebRequest request = new GetMethodWebRequest(url);
     request.setParameter("principal","W3L");
     WebResponse response = conversation.getResponse(request);
     assertNotNull(response);
     WebResponse loginFrame  =
         conversation.getFrameContents("login_startportal");
     assertNotNull(loginFrame);
     WebForm loginForm = loginFrame.getForms()[0];
     assertNotNull(loginForm);
     request = loginForm.getRequest();
     request.setParameter("user",“testuser");
     request.setParameter("password","");
     response = conversation.getResponse( request );
     assertTrue(response.getText().indexOf("Zugang ungültig" ) != -1 );
}
```

Bewertung

- HttpUnit
  - eignet sich sehr gut zum Testen von dynamisch erstellten HTML-Seiten
  - Leider wird Java-Script nur in geringem Maße unterstützt
  - Browser-spezifische Java-Script-Anweisungen werden gar nicht unterstützt
  - Da Java-Script bei vielen Web-Anwendungen zum Einsatz kommt, ist die Kombination von JUnit und HttpUnit nur eingeschränkt zum Web-Testen geeignet

Bewertung

- JUnit
  - ist ein kleines, aber mächtiges Framework zum Durchführen von Komponententests
  - ist mittlerweile weit verbreitet
  - richtet sich an den Entwickler, nicht ans QM
  - eignet sich sehr gut zum Testen von Java-Klassen
  - Sinnvolle Anwendung von JUnit beim Web-Testen :
    - Falls bei der Web-Anwendung Java Server Pages (JSPs) eingesetzt werden und diese Java Beans benutzen, kann JUnit verwendet werden, um die Java Beans zu testen.

Literatur

- Die Homepage von JUnit:
  - www.junit.org
- Die Homepage von HttpUnit:
  - www.httpunit.org
- Artikel Unit-Testing mit JUnit
  - www.frankwestphal.de/UnitTestingmitJunit.html
- Projektarbeit Automatisiertes Erstellen und Testen von Web-Anwendungen (FH Dortmund 2002 / Peter Kesch)
- Java-Magazin 1/03
  - Artikel JUnit-Tests automatisieren

- Danke!