

# Welten vereinigt euch!

Testen von  
PHP Applikationen  
mithilfe von Java Tools

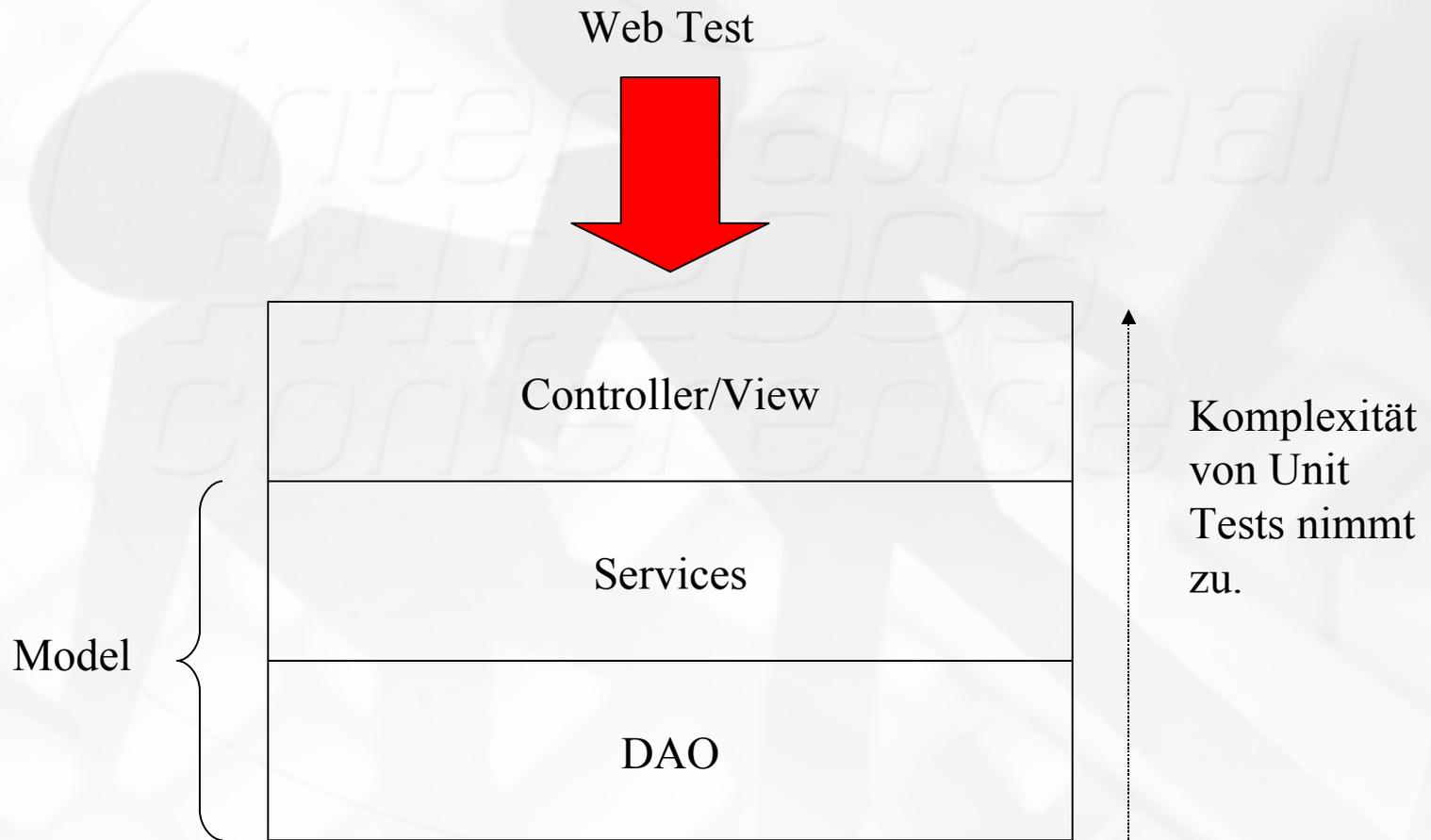
# Agenda

- Unit Tests
- Web Testing
- Durchführung von Web Tests
- JWebUnit
- Tests mit JWebUnit
- Verschiedene HTML Elemente
- Testautomatisierung und Reporting mit Ant
- Testing und Datenbanken
- Quellen

# Unit Tests

- Aufgabe von Unit Tests
  - Entwickler Tests
  - Extreme Programming Ansatz
  - Zuerst den Test schreiben, dann die Implementation.
  - ...

# Warum Web Tests?



# Warum Web Tests?

- Unit Tests haben Grenzen
  - Graphische Benutzer Oberflächen (GUI)
    - Model-View-Controller (MVC) Strukturen
      - Model Ok
      - View + Controller mithilfe anderer Frameworks
  - Nebenläufiger Code
  - Mehrschichten Architekturen
    - Mock-Objects (Test Pattern)

# Warum Web Test?

- Um Logik usw. zu testen, die mit Unit Tests nicht abgedeckt werden kann.
  - z.B. Workflow etc. der durch Controller/View dargestellt wird.
- Test aus Sicht des Anwenders.
- Eventuell einfacher als ein Controller/View Test Framework.
- ...

# Durchführung von Web Tests

- Von Hand
  - Vorteil:
    - Einfach
  - Nachteil:
    - Nicht reproduzierbar!
    - Sehr zeitaufwendig!

# Durchführung von Web Tests

- Mithilfe eines Werkzeuges:
  - Vorteil:
    - Reproduzierbar!
    - Ablaufzeit ist kalkulierbar!
    - Integration in den Entwicklungsprozess
  - Nachteil:
    - Je nach Entwicklungsprozess aufwendiger zu erstellen.
    - Einarbeitung

# Durchführung von Web Tests

- Kommerzielle Werkzeuge
  - e-Test suite von empirix
  - SilkTest von segue
  - WinRunner mercury
  - etc.

# Durchführung von Web Tests

- Kommerzielle Werkzeuge
- Nachteile:
  - Anschaffungskosten (Investitionskosten)
  - Einarbeitungs- bzw. Schulungsaufwand
  - Portabilität

# Durchführung von Web Tests

- Kommerzielle Werkzeuge
- Vorteile:
  - Grafische Benutzer Oberfläche (GUI)
    - Nutzung durch nicht technisches Personal.
  - Je nach Projektgröße/Struktur
    - Kosteneinsparung

# Durchführung von Web Tests

- OpenSource Werkzeuge
  - canoo WebTest
  - HTMLUnit
  - HttpUnit
  - JWebUnit
  - etc.

# Durchführung von Web Tests

- OpenSource Werkzeuge
- Nachteile:
  - Erstellung der Testfälle ist je nach Tool mehr technisch als fachlich orientiert (zum Teil programmiert).
  - Teilweise keine GUI

# Durchführung von Web Tests

- OpenSource Werkzeuge
- Vorteile:
  - Einfache Integration in den Entwicklungsprozess.
  - Frühere Rückmeldung bzgl. der fachlichen Anforderungen.

# JWebUnit

- JWebUnit
  - Die Applikation wird aus der Sicht des Anwenders getestet.
  - JWebUnit simuliert die Interaktion mit der Web Applikation.
  - Einfache Aufstellung der Test-Cases wie bei Unit Tests.
  - Integration in die IDE (z.B. Eclipse, IntelliJ).
  - Einfache Integration in den Build/Deployment Prozess per Ant.

# Warum Java und nicht PHP?

- Teilweise fehlende Integration in IDE's bei PHP.
- Unabhängigkeit von der PHP Version
- Andere Sicht der Dinge
  - Senkung der Fehlerwahrscheinlichkeit

# Warum Java und nicht PHP?

- Wird vielleicht im Projekt sowieso schon verwendet.
- Werkzeuge schon lange im Einsatz, daraus folgt eine gewisse Qualität.
- Erfahrungszuwachs durch den Einsatz einer anderen Programmiersprache bzw. Werkzeugen.

# Test Planung

- 1. Test
  - Seite aufrufen.
  - Prüfen, ob auch alle notwendigen Angaben auf der Seite sind.
  - ...
- ...

# 1. Test mit JWebUnit

**PAWAs**  
**Projekt Aufwanderfassungs- und Abrechnungssystem**

**Anmeldung**

Mandant-Nr:   
Anmeldung:   
Passwort:

PAWAs V0.50.1 Revision \$Rev: 25 \$ © Karl Heinz Marbaise

# 1. Test mit JWebUnit

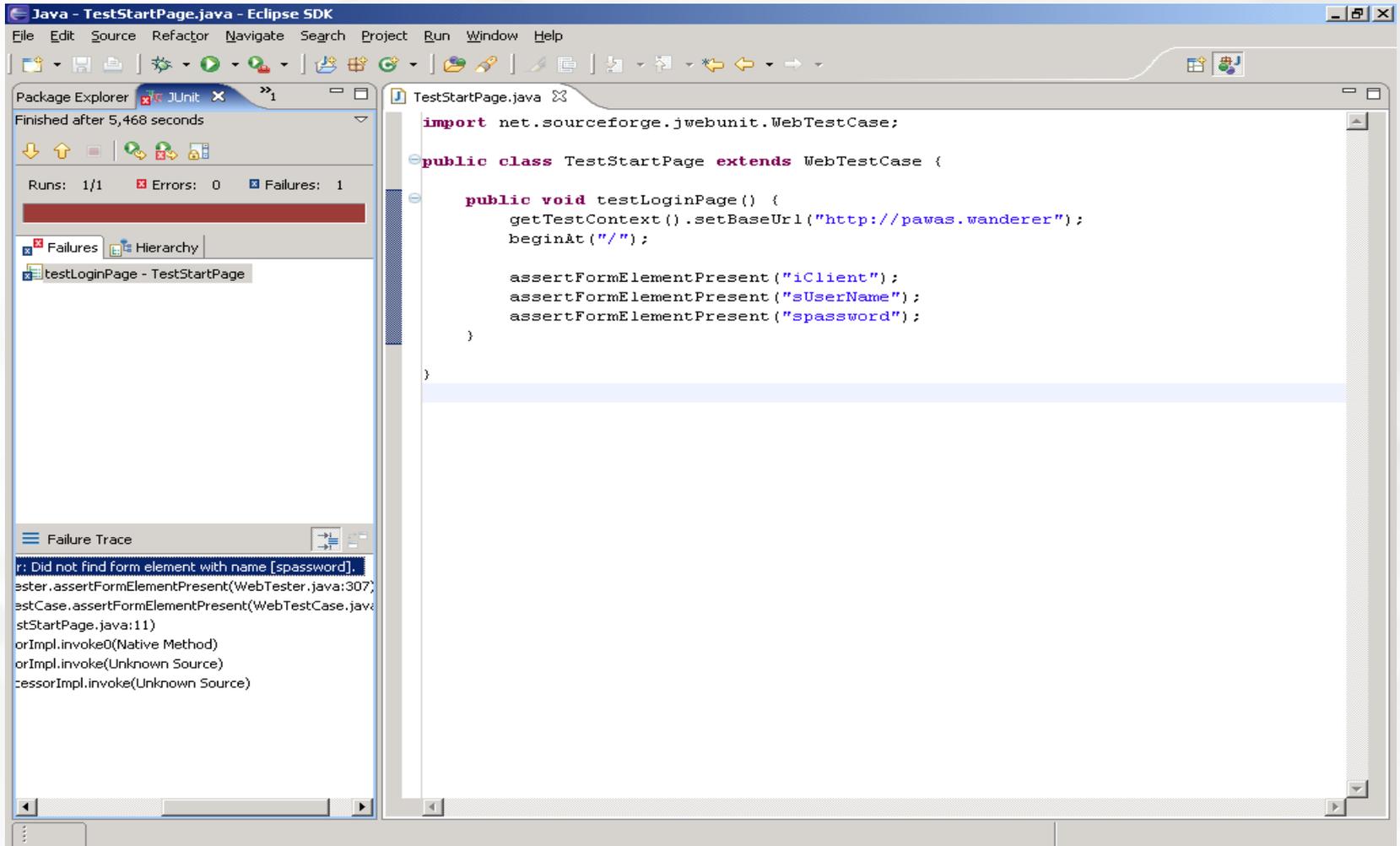
```
import net.sourceforge.jwebunit.WebTestCase;

public class TestStartPage extends WebTestCase {

    public void testLoginPage () {
        getTestContext ()
        .setBaseUrl ("http://pawas.wanderer");
        beginAt ("/");

        assertFormElementPresent ("iClient");
        assertFormElementPresent ("sUserName");
        assertFormElementPresent ("spassword");
    }
}
```

# 1. Test mit JWebUnit



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left indicates the test has finished after 5,468 seconds with 1 failure. The Failure Trace window shows the error: "r: Did not find form element with name [spassword]". The main editor displays the following Java code for TestLoginPage.java:

```
import net.sourceforge.jwebunit.WebTestCase;

public class TestLoginPage extends WebTestCase {

    public void testLoginPage() {
        getTestContext().setBaseUrl("http://pawas.wanderer");
        beginAt("/");

        assertFormElementPresent("iClient");
        assertFormElementPresent("sUserName");
        assertFormElementPresent("spassword");
    }
}
```

# 1. Test mit JWebUnit

The screenshot shows the Eclipse IDE interface. The main editor displays the following Java code:

```
import net.sourceforge.jwebunit.WebTestCase;

public class TestStartPage extends WebTestCase {

    public void testLoginPage() {
        getTestContext().setBaseUrl("http://pawas.wanderer");
        beginAt("/");

        assertFormElementPresent("iClient");
        assertFormElementPresent("sUserName");
        assertFormElementPresent("spassword");
    }
}
```

The Package Explorer on the left shows the test results: "Finished after 5,468 seconds", "Runs: 1/1", "Errors: 0", and "Failures: 1". The Failure Trace window at the bottom left shows the following error message:

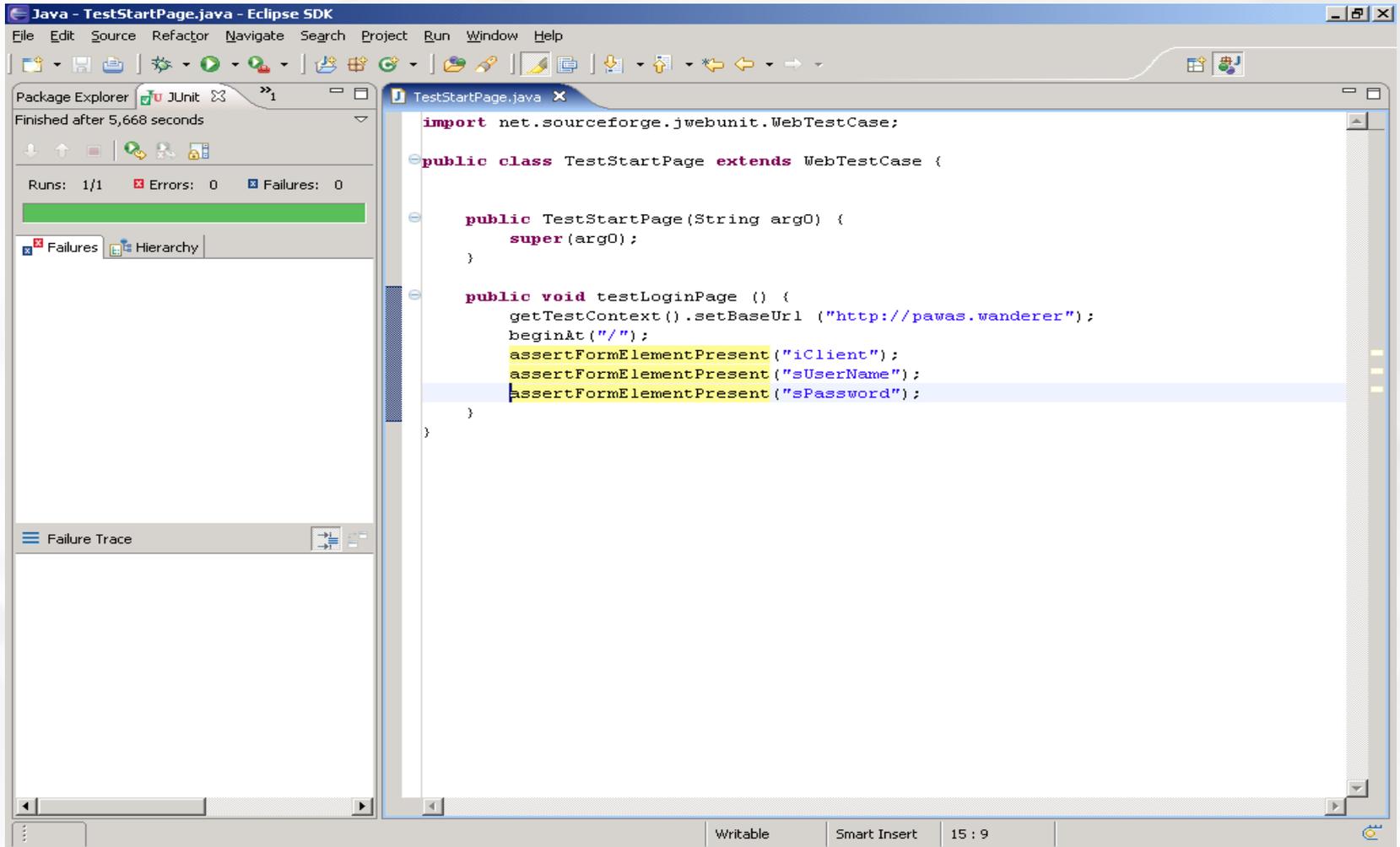
```
Failure Trace
r: Did not find form element with name [spassword].
ester.assertFormElementPresent(WebTester.java:307)
estCase.assertFormElementPresent(WebTestCase.java:
stStartPage.java:11)
orImpl.invoke0(Native Method)
orImpl.invoke(Unknown Source)
cessorImpl.invoke(Unknown Source)
```

# 1. Test mit JWebUnit

```
public void testLoginPage () {  
    getTestContext().setBaseUrl ("http://pawas.wanderer");  
    beginAt ("/");  
    assertFormElementPresent ("iClient");  
    assertFormElementPresent ("sUserName");  
    assertFormElementPresent ("spassword");  
}
```

Tippfehler „p“ muss groß geschrieben werden.

# 1. Test mit JWebUnit



# Test Planung

- 1.Test
  - Seite aufrufen.
  - Prüfen, ob auch alle notwendigen Angaben auf der Seite sind.
  - ...
- 2.Test
  - Anmeldung durchführen
  - Prüfen, ob alle erwarteten Elemente auf der Seite vorhanden sind.
- ...

# 2. Test mit JWebUnit

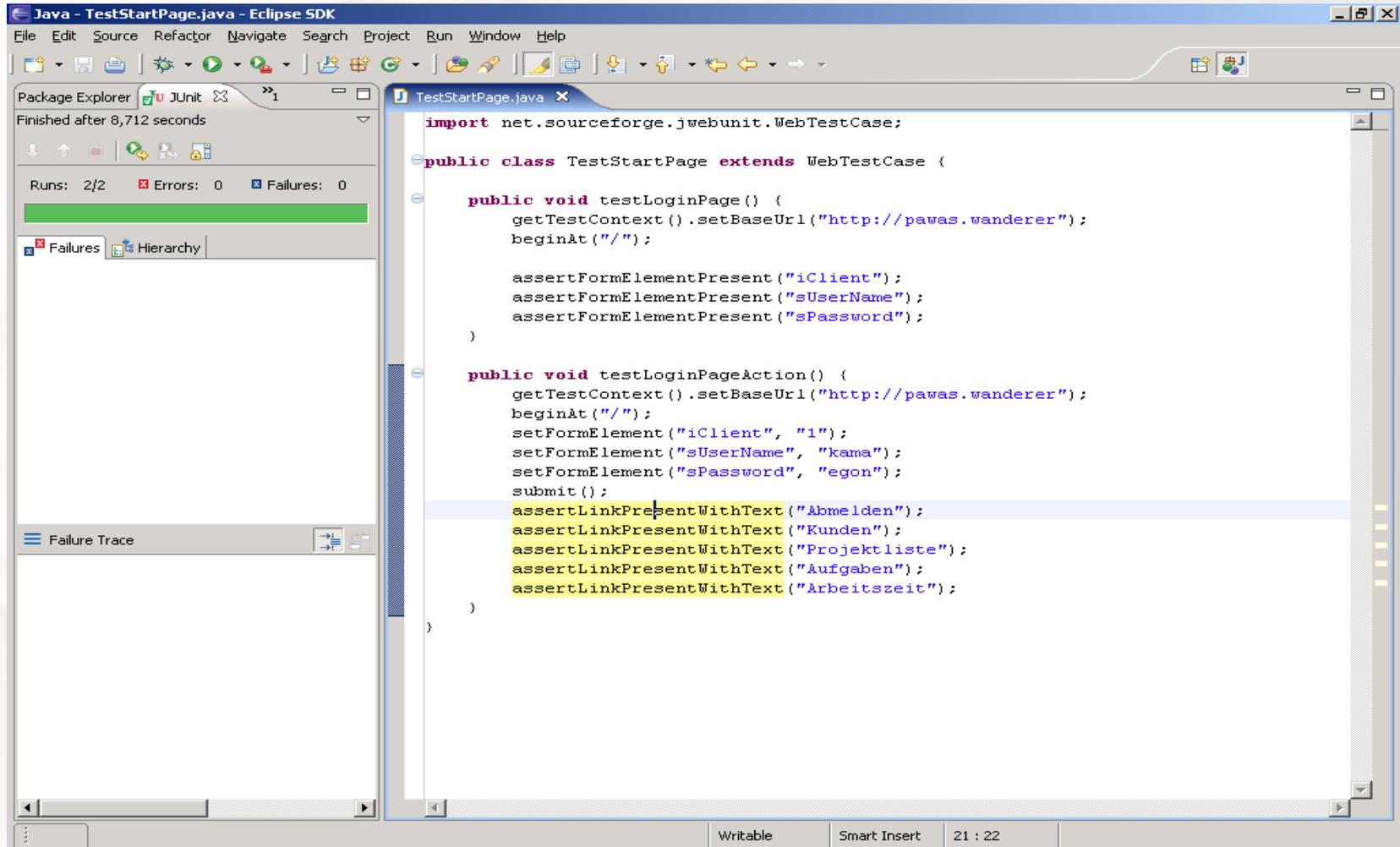
Abmelden | Kunden | Projektliste | Aufgaben | Arbeitszeit PAWAs Rel.0.50.1

Aktionen		Name	Kurzbezeichnung	Löschen	Ordnung
		<a href="#">Kunde Nummer 1 C1</a>	KuNr1		▼
		<a href="#">Kunde Nummer 4 C1</a>	KuNr4		▲

## 2. Test mit JWebUnit

```
public class TestStartPage extends WebTestCase {
    ...
    public void testLoginPageAction() {
        getTestContext()
        .setBaseUrl("http://pawas.wanderer");
        beginAt("/");
        setFormElement("iClient", "1");
        setFormElement("sUserName", "kama");
        setFormElement("sPassword", "egon");
        submit();
        assertLinkPresentWithText("Abmelden");
        assertLinkPresentWithText("Kunden");
        assertLinkPresentWithText("Projektliste");
        assertLinkPresentWithText("Aufgaben");
        assertLinkPresentWithText("Arbeitszeit");
    }
}
```

# 2. Test mit JWebUnit



The screenshot shows the Eclipse IDE with a Java test class named `TestStartPage.java` open. The class extends `WebTestCase` and contains two test methods: `testLoginPage()` and `testLoginPageAction()`. The `testLoginPageAction()` method includes assertions for the presence of links with specific text.

```
import net.sourceforge.jwebunit.WebTestCase;

public class TestStartPage extends WebTestCase {

    public void testLoginPage() {
        getTestContext().setBaseUrl("http://pawas.wanderer");
        beginAt("/");

        assertFormElementPresent("iClient");
        assertFormElementPresent("sUserName");
        assertFormElementPresent("sPassword");
    }

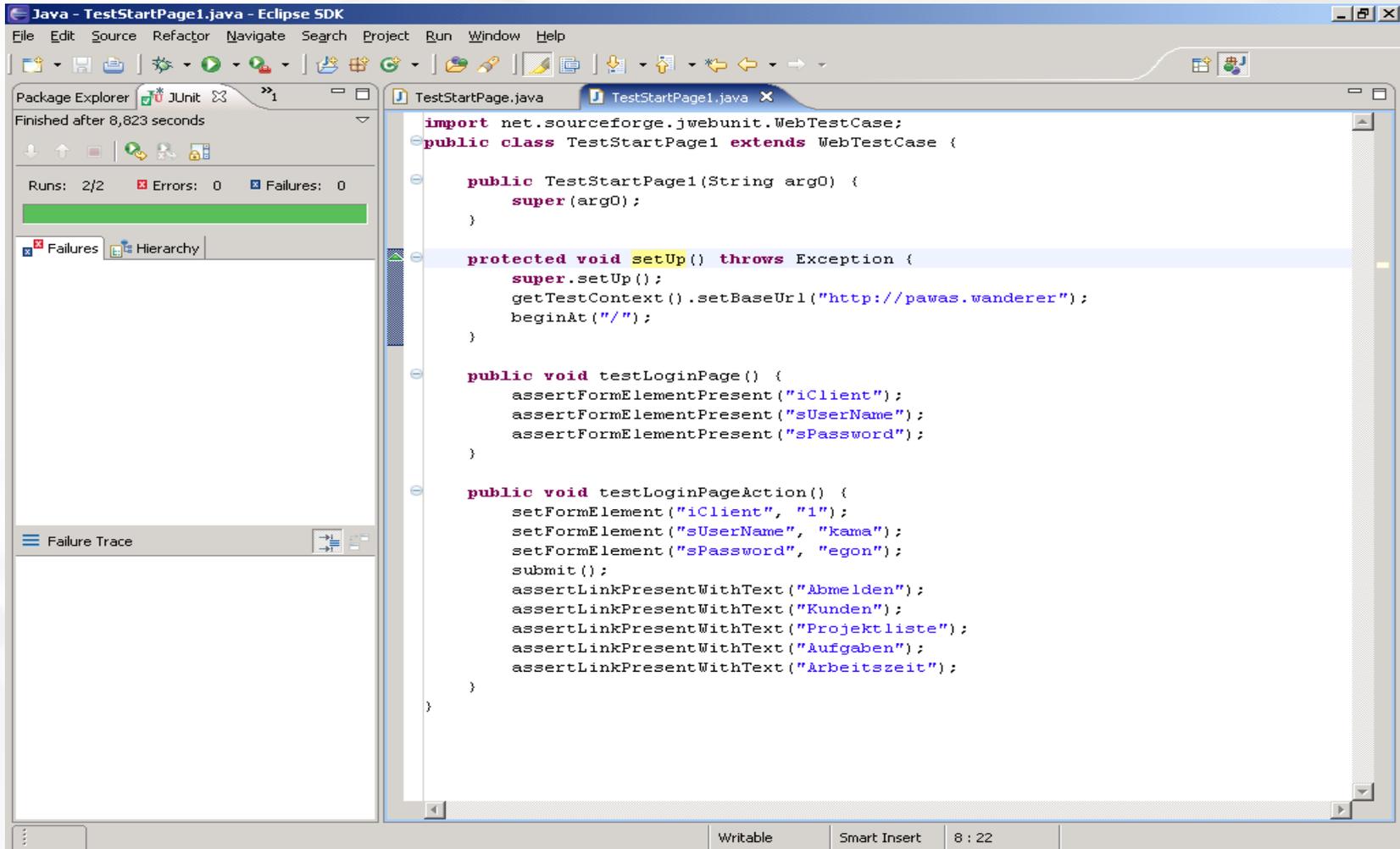
    public void testLoginPageAction() {
        getTestContext().setBaseUrl("http://pawas.wanderer");
        beginAt("/");
        setFormElement("iClient", "1");
        setFormElement("sUserName", "kama");
        setFormElement("sPassword", "egon");
        submit();
        assertLinkPresentWithText("Abmelden");
        assertLinkPresentWithText("Kunden");
        assertLinkPresentWithText("Projektliste");
        assertLinkPresentWithText("Aufgaben");
        assertLinkPresentWithText("Arbeitszeit");
    }
}
```

The Package Explorer on the left shows the test results: "Finished after 8,712 seconds", "Runs: 2/2", "Errors: 0", and "Failures: 0". The Failure Trace is empty.

## 2. Test mit JWebUnit

```
public class TestStartPage1 extends WebTestCase {  
    ...  
    protected void setUp () throws Exception {  
        super.setUp ();  
        getTestContext ()  
        .setBaseUrl ("http://pawas.wanderer");  
        beginAt ("/");  
    }  
    public void testLoginPageAction() {  
        setFormElement ("iClient", "1");  
        setFormElement ("sUserName", "kama");  
        setFormElement ("sPassword", "egon");  
        ...  
    }  
}
```

# 2. Test mit JWebUnit



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left indicates that the tests finished after 8,823 seconds with 2/2 runs, 0 errors, and 0 failures. The main editor displays the following Java code for `TestStartPage1.java`:

```
import net.sourceforge.jwebunit.WebTestCase;
public class TestStartPage1 extends WebTestCase {

    public TestStartPage1(String arg0) {
        super(arg0);
    }

    protected void setUp() throws Exception {
        super.setUp();
        getTestContext().setBaseUrl("http://pawas.wanderer");
        beginAt("/");
    }

    public void testLoginPage() {
        assertFormElementPresent("iClient");
        assertFormElementPresent("sUserName");
        assertFormElementPresent("sPassword");
    }

    public void testLoginPageAction() {
        setFormElement("iClient", "1");
        setFormElement("sUserName", "kama");
        setFormElement("sPassword", "egon");
        submit();
        assertLinkPresentWithText("Abmelden");
        assertLinkPresentWithText("Kunden");
        assertLinkPresentWithText("Projektliste");
        assertLinkPresentWithText("Aufgaben");
        assertLinkPresentWithText("Arbeitszeit");
    }
}
```

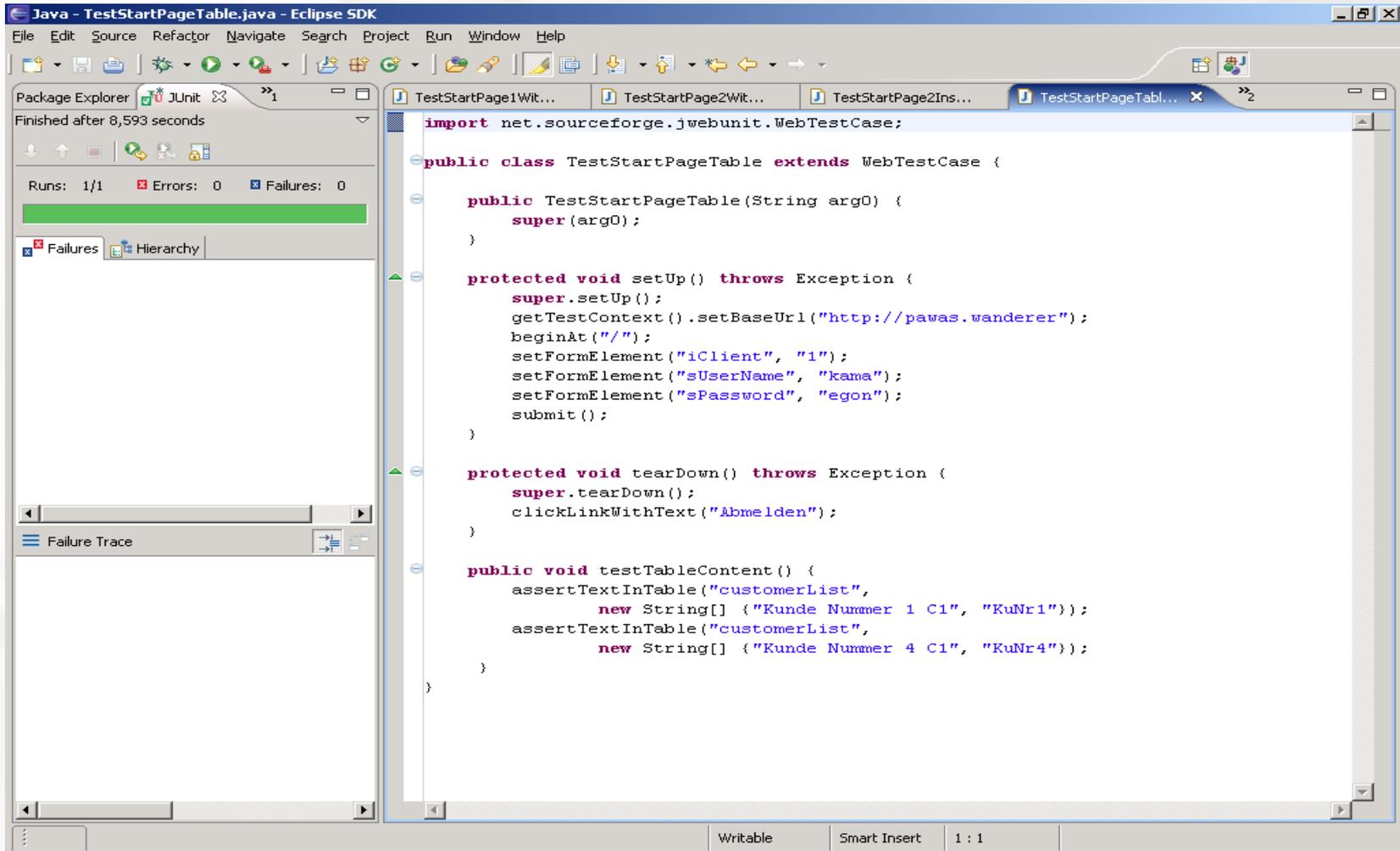
# Test Planung

- ...
- **2.Test**
  - Anmeldung durchführen
  - Prüfen, ob alle erwarteten Elemente auf der Seite vorhanden sind.
    - Tabelleninhalte prüfen
- ...

## 2. Test mit JWebUnit

```
public class TestStartPageTable extends WebTestCase{
    ...
    public void testTableContent() {
        assertTextInTable("customerList",
            new String[] {
                "Kunde Nummer 1 C1",
                "KuNr1"
            });
        assertTextInTable("customerList",
            new String[] {
                "Kunde Nummer 4 C1",
                "KuNr4"
            });
    }
}
```

# 2. Test mit JWebUnit



```
import net.sourceforge.jwebunit.WebTestCase;

public class TestStartPageTable extends WebTestCase {

    public TestStartPageTable(String arg0) {
        super(arg0);
    }

    protected void setUp() throws Exception {
        super.setUp();
        getTestContext().setBaseUrl("http://pawas.wanderer");
        beginAt("/");
        setFormElement("iClient", "1");
        setFormElement("sUserName", "kama");
        setFormElement("sPassword", "egon");
        submit();
    }

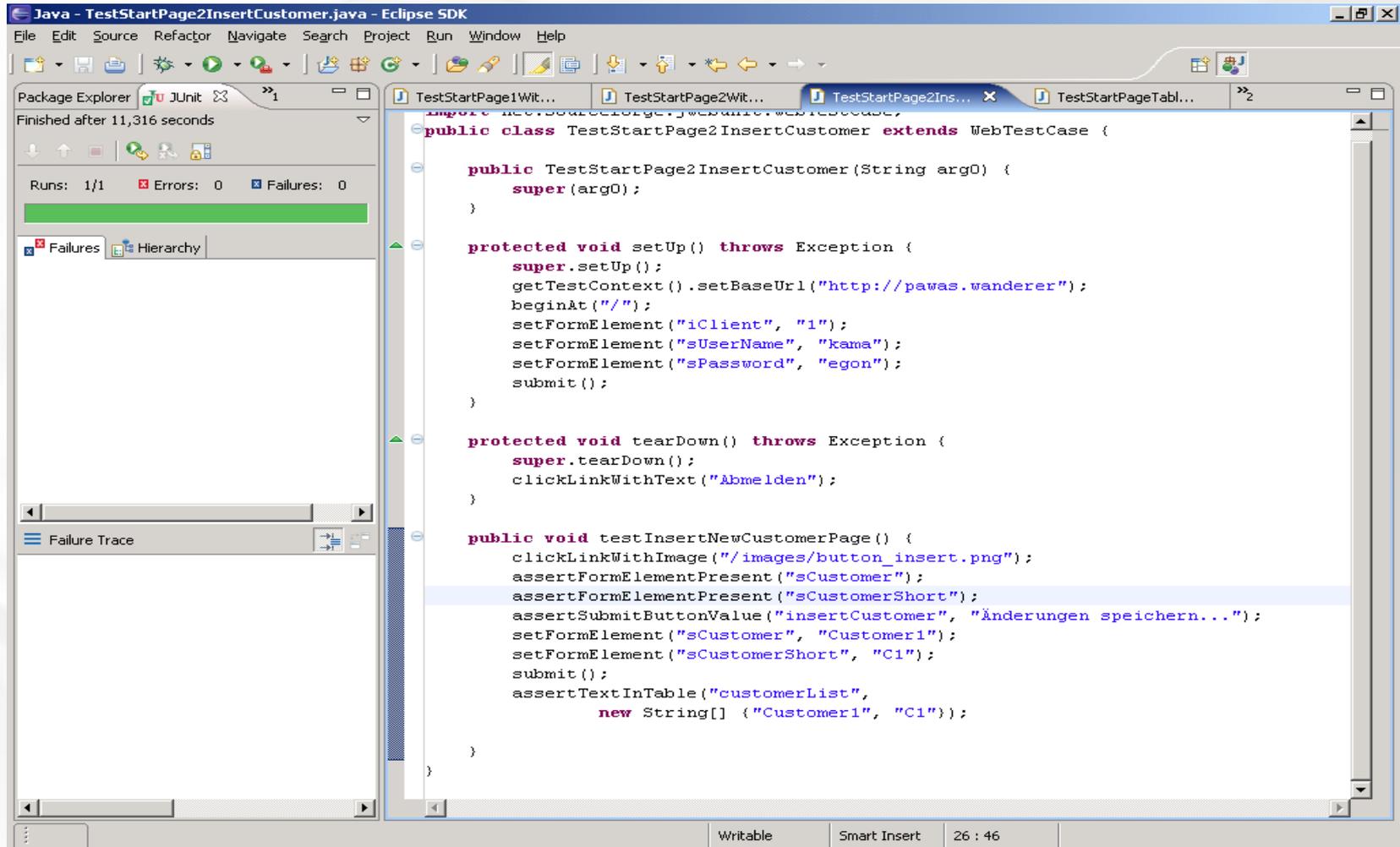
    protected void tearDown() throws Exception {
        super.tearDown();
        clickLinkWithText("Abmelden");
    }

    public void testTableContent() {
        assertTextInTable("customerList",
            new String[] {"Kunde Nummer 1 C1", "KuNr1"});
        assertTextInTable("customerList",
            new String[] {"Kunde Nummer 4 C1", "KuNr4"});
    }
}
```

# Test Planung

- ...
- 3. Test
  - Anmeldung durchführen
  - “Kunden anlegen” auswählen
  - Prüfen, ob alle erwarteten Elemente da sind
  - Form mit Daten füllen und abschicken.
  - Tabelle auf entsprechenden Eintrag überprüfen.
- ...

# 3. Test mit JWebUnit



```
import net.sourceforge.jwebunit.junit.WebTestCase;

public class TestStartPage2InsertCustomer extends WebTestCase {

    public TestStartPage2InsertCustomer(String arg0) {
        super(arg0);
    }

    protected void setUp() throws Exception {
        super.setUp();
        getTestContext().setBaseUrl("http://pawas.wanderer");
        beginAt("/");
        setFormElement("iClient", "1");
        setFormElement("sUserName", "kama");
        setFormElement("sPassword", "egon");
        submit();
    }

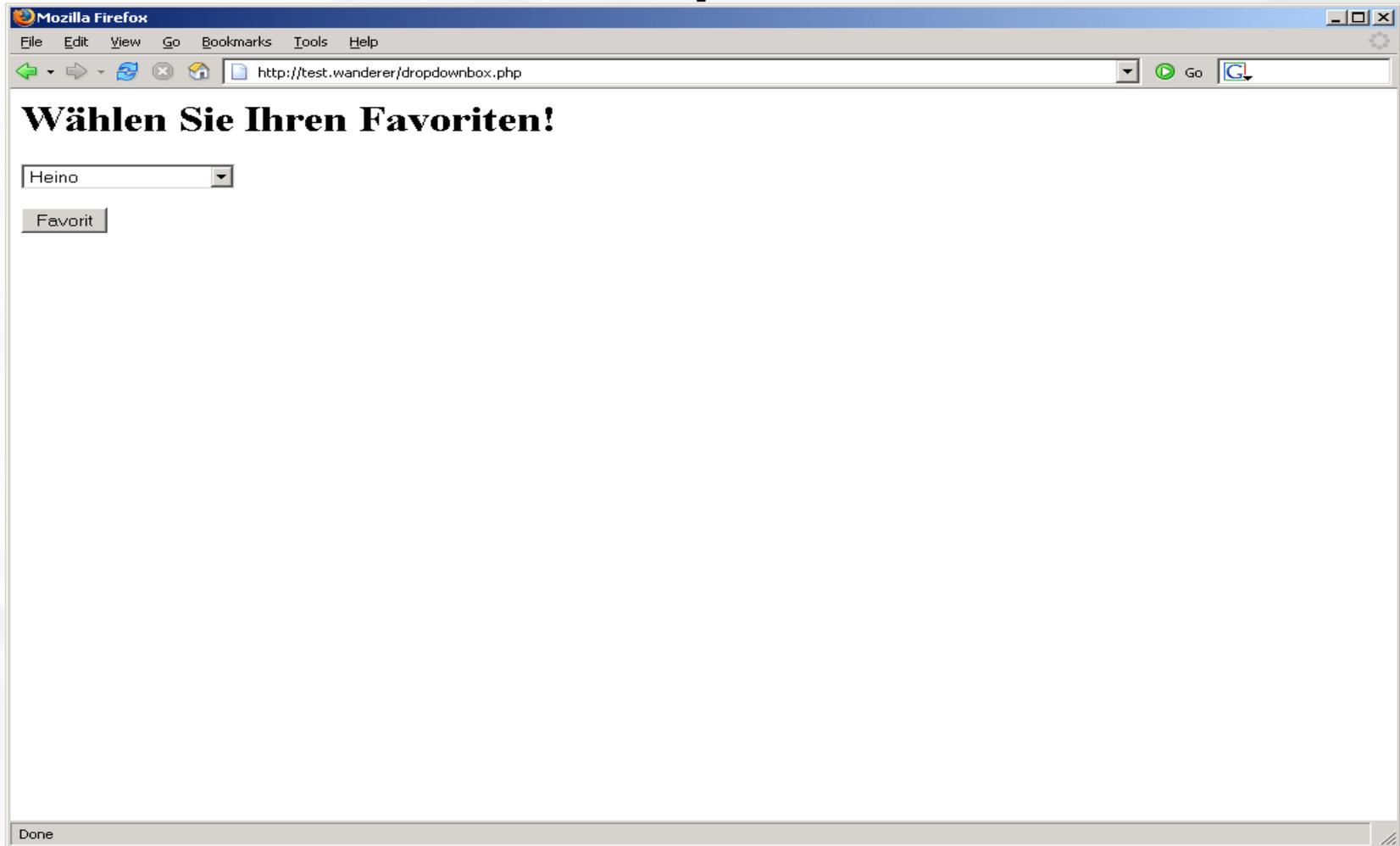
    protected void tearDown() throws Exception {
        super.tearDown();
        clickLinkWithText("Abmelden");
    }

    public void testInsertNewCustomerPage() {
        clickLinkWithImage("/images/button_insert.png");
        assertFormElementPresent("sCustomer");
        assertFormElementPresent("sCustomerShort");
        assertSubmitButtonValue("insertCustomer", "Änderungen speichern...");
        setFormElement("sCustomer", "Customer1");
        setFormElement("sCustomerShort", "C1");
        submit();
        assertTextInTable("customerList",
            new String[] {"Customer1", "C1"});
    }
}
```

# Formen

- Testen, ob eine bestimmte Form auf einer Seite vorhanden ist:
  - Existenz kann mithilfe von `assertFormPresent("LoginForm");` geprüft werden.
- Mehrere Formen auf einer Seite:
  - Auswahl der gewünschten Form mithilfe von `setWorkingForm("bottomForm");`

# HTML: DropDown-Box



# HTML: DropDown-Box

...

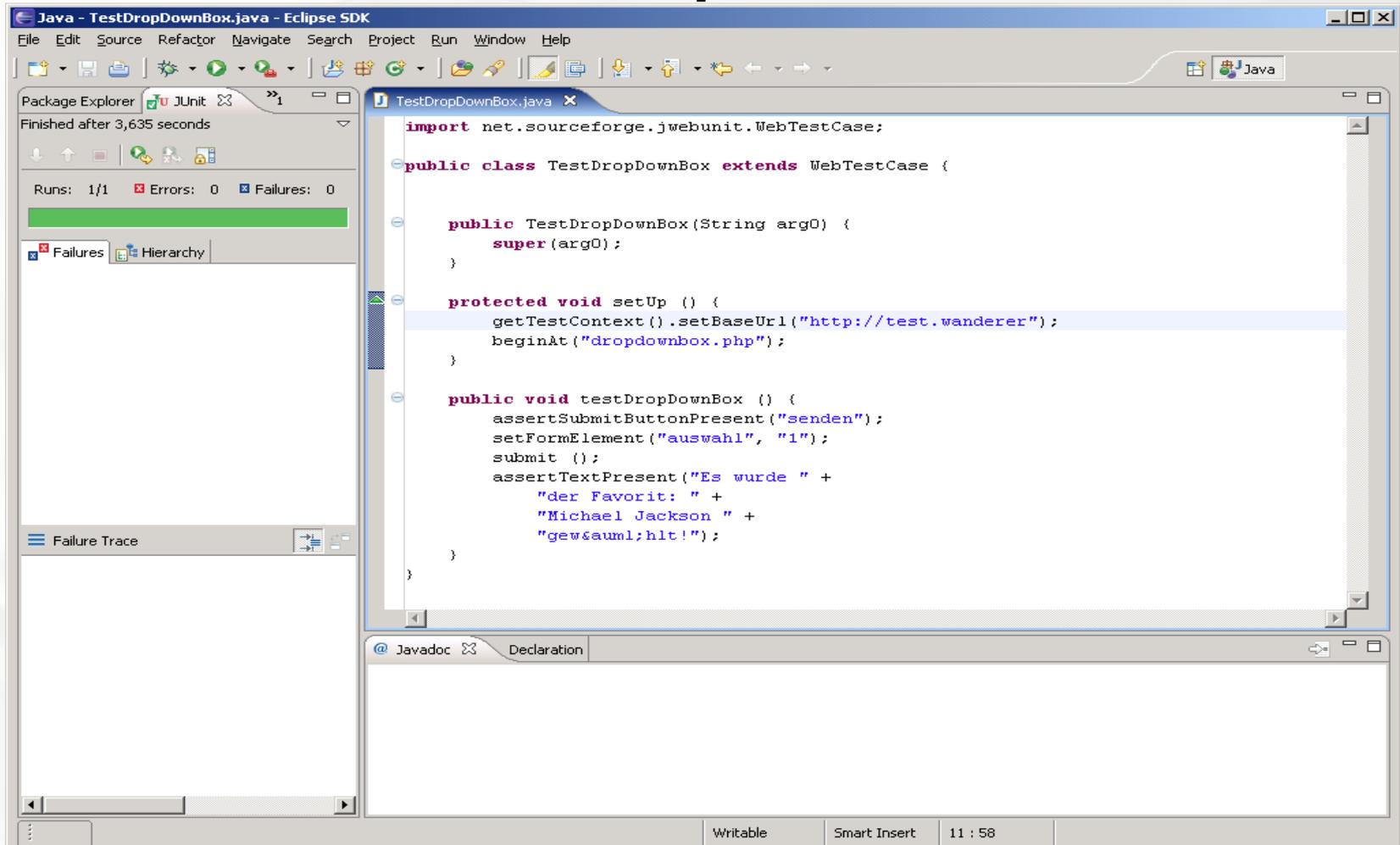
```
<select name="auswahl">  
  <option value="0">Heino</option>  
  <option value="1">Michael Jackson</option>  
  <option value="2">Tom Waits</option>  
  <option value="3">Nina Hagen</option>  
  <option value="4">Marianne  
  Rosenberg</option>  
</select>
```

...

# HTML: DropDown-Box

```
public class DropDown extends WebTestCase {
    ...
    public void testDropDownBox () {
        setFormElement("auswahl", "1");
        submit ();
        assertTextPresent("Es wurde "
            + "der Favorit: "
            + "Michael Jackson "
            + "gew&auml;hlt!");
    }
}
```

# HTML: DropDown-Box



The screenshot shows the Eclipse IDE interface. The main editor window displays the following Java code for `TestDropDownBox.java`:

```
import net.sourceforge.jwebunit.WebTestCase;

public class TestDropDownBox extends WebTestCase {

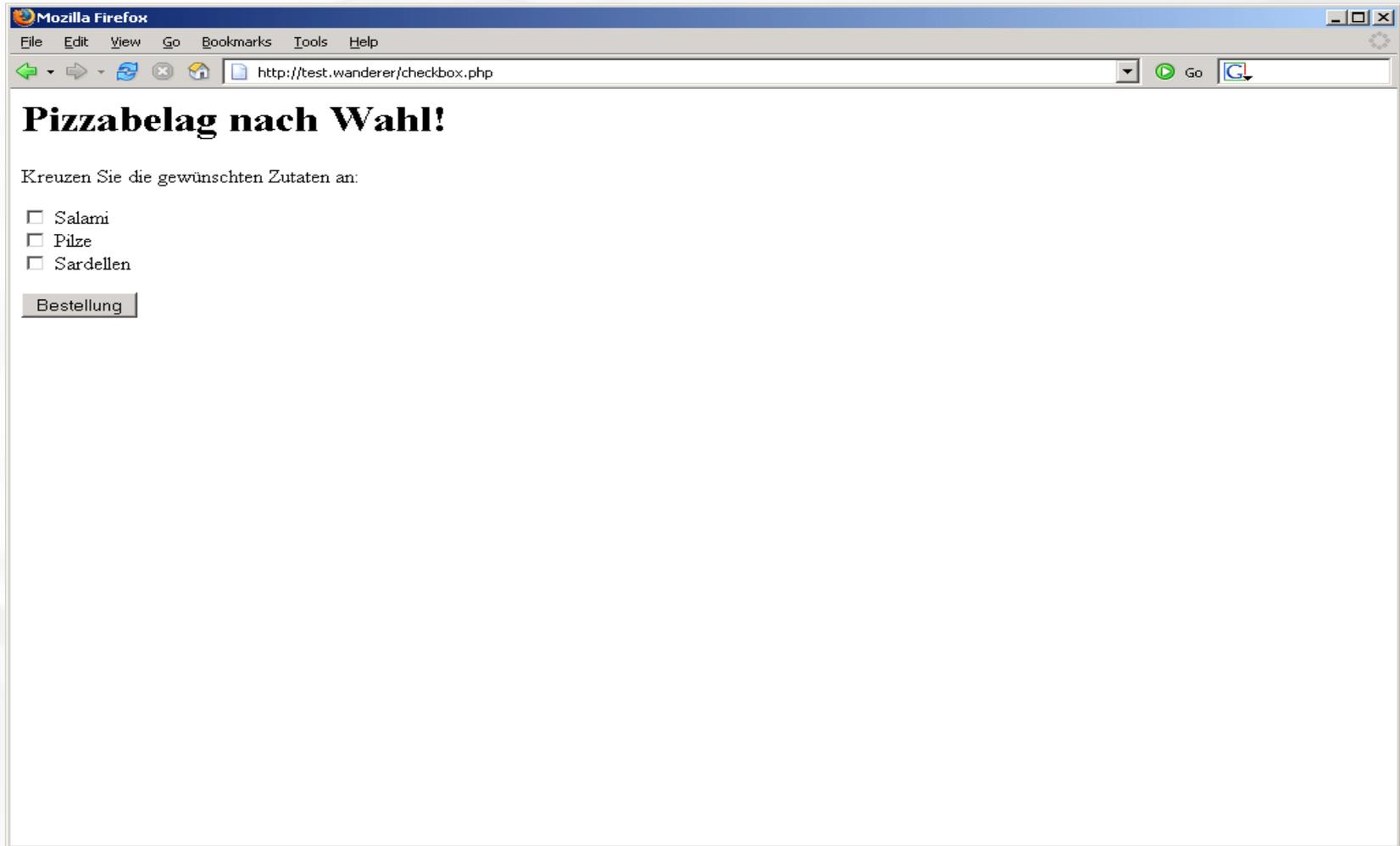
    public TestDropDownBox(String arg0) {
        super(arg0);
    }

    protected void setUp () {
        getTestContext().setBaseUrl("http://test.wanderer");
        beginAt("dropdownbox.php");
    }

    public void testDropDownBox () {
        assertSubmitButtonPresent("senden");
        setFormElement("auswahl", "1");
        submit ();
        assertTextPresent("Es wurde " +
            "der Favorit: " +
            "Michael Jackson " +
            "gew&auml;hlt!");
    }
}
```

The Package Explorer on the left shows a successful test run: "Finished after 3,635 seconds", "Runs: 1/1", "Errors: 0", and "Failures: 0". The bottom status bar indicates "Writable", "Smart Insert", and the time "11 : 58".

# HTML: CheckBoxen



# HTML: CheckBoxen

...

```
<input type="checkbox"
       name="zutat[]"
       value="salami"> Salami<br>
<input type="checkbox"
       name="zutat[]"
       value="pilze"> Pilze<br>
<input type="checkbox"
       name="zutat[]"
       value="sardellen"> Sardellen
```

...

# HTML: CheckBoxen

```
...  
public void testCheckBox () {  
    setFormElement (  
        "zutat[ ]",  
        "salami");  
    submit ();  
    assertTextPresent ("Es wurden "  
+ "die Zutaten: "  
+ " salami "  
+ "gew&auml;hlt!");  
}  
...
```

# HTML: Radio Buttons

...

```
<input type="radio"
       name="Zahlmethode"
       value="Mastercard">Mastercard<br>
<input type="radio"
       name="Zahlmethode"
       value="Visa">Visa<br>
<input type="radio"
       name="Zahlmethode"
       value="AmericanExpress">American
Express
```

...

# HTML: Radio Buttons

```
...  
public void testRadio () {  
    setFormElement (  
        "Zahlmethode",  
        "Visa");  
    submit ();  
    assertTextPresent ("Es wurde "  
+ "die Zahlungsart Visa "  
+ "ausgew&auml;hlt!");  
}  
...
```

# Sonstiges zu JWebUnit

- Multi Select
  - Funktioniert nicht mit JWebUnit 1.2
    - Aber eventuell mit JWebUnit 1.3 ?
    - Leider ist noch nicht klar wann die kommt!
  - Prüfung von Tabelleninhalten
    - Vor allem bei Spalten / Zeilen, die ignoriert werden sollen, ein wenig umständlich.
    - Marke Eigenbau!

# Sonstiges zu JWebUnit

- JavaScript
  - Es wird derzeit ein gewisser Rahmen von JavaScript unterstützt.
    - ECMA-262/ECMAScript Spezifikation bzw. ISO-16262 (siehe Rhino Doku)
    - Eventuell die aktuelle Version von Rhino in JWebUnit übernehmen.
  - Im Zweifel ausprobieren bzw. abschalten.

# Sonstiges zu JWebUnit

- Zu Fehlersuche im Test ;-)
  - Zur direkten Ausgabe auf Console:
    - `dumpResponse(System.out);`
  - Zur eventuellen Weiterverarbeitung:
    - `String t = getDialog().getResponseText();`
  - Abschalten von JavaScript:
    - `HttpUnitOptions.setScriptionEnabled(false);`

# Sonstiges zu JWebUnit

- Simulation eines anderen Browsers:

```
TestContext ct = getTestContext();
ClientProperties cp =
    ct
    .getWebClient()
    .getClientProperties();
cp.setUserAgent("BrowserName");
```

# Test Automatisierung mit Ant

- Was ist Ant?
  - Ein Build Tool das ähnlich wie Make/GNU Make arbeitet aber viele Probleme von Make beseitigt.
  - Es ist in Java geschrieben und wurde auch ursprünglich als ein portables Build Tool entwickelt.
  - Erweiterung über die Erstellung sg. Task (in Java) recht einfach möglich.

# Test Automatisierung mit Ant

- Was ist Ant?
  - Aufgaben und Abhängigkeiten usw. werden in Form eines XML Files abgelegt (build.xml).
    - Compilierung von C/C++/Java Quellen.
    - Ausführen von Remote Befehlen per scp.
    - Erzeugung von tar/zip/gzip/bzip2/jar/war/ear Archiven.
    - etc.

# Warum Ant und nicht Phing?

- Keine Unterstützung von CVS/Subversion etc. bzw. nicht dokumentiert.
  - Kein checkin möglich per Phing
- Kein Increment von Properties
  - Build Nummer etc.
- Keine Unterstützung von ftp/scp o.ä.
  - Transfer von Dateien auf Test bzw. Produktiv Umgebung
  - Keine Ausführung von Kommandos auf Remote Rechner.

# Test Automatisierung mit Ant

- Testdurchläufe wann immer man möchte ohne selbst in die Tasten greifen zu müssen.
- Verknüpfung mit dem Deployment Prozess.

# Ant Build File Einführung

...

```
<property name="src.dir" location="src"/>
```

```
<property name="jars.dir" location="lib"/>
```

...

```
<property  
  name="build.dir"  
  location="build"/>
```

```
<property  
  name="build.classes.dir"  
  location="${build.dir}/classes"/>
```

# Ant Build File Einführung

...

```
<path id="library">  
  <fileset dir="{jars.dir}">  
    <include name="*.jar"/>  
  </fileset>  
</path>
```

...

# Ant Build File Einführung

...

```
<target
  name="init"
  description="Create all needed directories">
  <mkdir dir="${dist.dir}" />
  <mkdir dir="${dist.docs.dir}" />

  <mkdir dir="${build.dir}" />
  <mkdir dir="${build.classes.dir}" />

  <mkdir dir="${reports.tests.dir}" />
  <mkdir dir="${reports.tests.html.dir}" />
</target>
```

...

# Ant Build File Einführung

...

```
<target
  name="compile"
  depends="init"
  description="Compile all Java files" >
  <javac
    srcdir="${src.dir}"
    destdir="${build.classes.dir}"
    debug="yes">
    <classpath refid="library"/>
  </javac>
</target>
```

...

# Ant Lauf

```
wanderer.gallien.de - PuTTY
kama@wanderer:~/projekte/Vortraege/PHPConference2005/beispiele> 1
insgesamt 56
drwxr-xr-x  6 kama users  4096 2005-11-06 12:25 ./
drwxr-xr-x  5 kama users  4096 2005-11-05 14:02 ../
drwxr-xr-x  2 kama users  4096 2005-11-05 17:15 bin/
-rw-r--r--  1 kama users  3325 2005-11-06 12:25 build.xml
-rw-r--r--  1 kama users  3271 2005-11-06 12:23 build.xml~
-rwxr--r--  1 kama users 12288 2005-11-06 12:25 .build.xml.swp*
-rw-r--r--  1 kama users   992 2005-11-05 14:07 .classpath
drwxr-xr-x  2 kama users  4096 2005-11-04 20:02 lib/
drwxr-xr-x  2 kama users  4096 2005-11-05 18:32 php/
-rw-r--r--  1 kama users   372 2005-10-28 19:14 .project
drwxr-xr-x  2 kama users  4096 2005-11-06 12:25 src/
-rw-r--r--  1 kama users  3539 2005-11-04 19:57 x.out
kama@wanderer:~/projekte/Vortraege/PHPConference2005/beispiele> ant junit
Buildfile: build.xml

init:
[mkdir] Created dir: /home/kama/projekte/Vortraege/PHPConference2005/beispiele/dist
[mkdir] Created dir: /home/kama/projekte/Vortraege/PHPConference2005/beispiele/dist/docs
[mkdir] Created dir: /home/kama/projekte/Vortraege/PHPConference2005/beispiele/build
[mkdir] Created dir: /home/kama/projekte/Vortraege/PHPConference2005/beispiele/build/classes
[mkdir] Created dir: /home/kama/projekte/Vortraege/PHPConference2005/beispiele/reports/tests
[mkdir] Created dir: /home/kama/projekte/Vortraege/PHPConference2005/beispiele/reports/tests/htm
1

compile:
[javac] Compiling 11 source files to /home/kama/projekte/Vortraege/PHPConference2005/beispiele/b
uild/classes

junit:
[junit] Running TestCheckBox
[junit] Tests run: 1, Failures: 0, Errors: 0, Time elapsed: 0,695 sec
[junit] Running TestRadioButton
[junit] Tests run: 1, Failures: 0, Errors: 0, Time elapsed: 0,696 sec
[junitreport] Transform time: 419ms

BUILD SUCCESSFUL
Total time: 4 seconds
kama@wanderer:~/projekte/Vortraege/PHPConference2005/beispiele> █
```

# Ant Lauf

Unit Test Results - Mozilla Firefox

file:///Z:/projekte/Vortraege/PHPConference2005/beispiele/reports/tests/html/index.html

[Home](#)

**Packages**

[<none>](#)

**Classes**

[TestCheckBox](#)

[TestRadioButton](#)

## Unit Test Results

Designed for use with [JUnit](#) and [Ant](#).

### Summary

Tests	Failures	Errors	Success rate	Time
2	0	0	100.00%	1.391

Note: *failures* are anticipated and checked for with assertions while *errors* are unanticipated.

### Packages

Name	Tests	Errors	Failures	Time(s)
<a href="#">&lt;none&gt;</a>	2	0	0	1.391

# Ant Lauf

Unit Test Results

Designed for use with [JUnit](#) and [Ant](#).

**Class TestCheckBox**

Name	Tests	Errors	Failures	Time(s)
<a href="#">TestCheckBox</a>	1	0	0	0.695

**Tests**

Name	Status	Type	Time(s)
testCheckBox	Success		0.455

[Properties »](#)

Home

Packages

[<none>](#)

[<none>](#)

Classes

[TestCheckBox](#)

[TestRadioButton](#)

# Web Testing und Datenbanken

- Bei Web Applikationen ist immer eine Datenbank mit im Spiel.
- Was ist mit dem Zustand der Datenbank?
  - Der Zustand der Datenbank wird durch die Applikation ständig geändert.
  - Somit keine Wiederholbarkeit der Test-Fälle gewährleistet!

# Web Testing und Datenbanken

- Lösung:
  - SQL Statements bzw. vollständige Datenbestände können mithilfe von Ant in die Datenbank übertragen werden.
  - Dadurch wird ein definierter Zustand der Datenbank erreicht.

# Web Testing und Datenbanken

...

```
<target name="db">  
  <sql  
    driver="com.mysql.jdbc.Driver"  
    url="jdbc:mysql://localhost:3306/pawasphp"  
    userid="pawasphp"  
    password="pawasphp"  
    src="pawasphp-dump.sql"  
  />  
</target>
```

...

# Ant und Datenbanken

```
wanderer.gallien.de - PuTTY
kama@wanderer:~/projekte/Vortraege/PHPConference2005/beispiele> ant db
Buildfile: build.xml

db:
    [sql] Executing file: /home/kama/projekte/Vortraege/PHPConference2005/beispiele/pawasphp-dump.
sql
    [sql] 147 of 147 SQL statements executed successfully

BUILD SUCCESSFUL
Total time: 1 second
kama@wanderer:~/projekte/Vortraege/PHPConference2005/beispiele> █
```

# Was fehlt noch?

- Im Zusammenhang mit Script Sprachen wie z.B. PHP ist die Code-Coverage Analyse (CCA) ein wichtiger Punkt.
  - Mithilfe der Xdebug Erweiterung lassen sich Informationen über den Code-Coverage aufzeichnen.
  - Export der Informationen in Form einer XML Datei.
  - Auswertung mithilfe von Java und Konvertierung in einen entsprechenden Report.
  - Integration in Ant in Form einer Task.

# Nutzung von Web-Tests

- Unit Tests haben sich mehr oder weniger durchgesetzt.
- Die Akzeptanz bei Web-Tests scheint noch recht gering zu sein.
- Im Java Bereich ist die Nutzung von Web Tests schon recht verbreitet

# Alternativen zu JWebUnit

- PHP
  - SimpleTest
    - Derzeit in PHP4; funktioniert auch unter PHP5
    - Keine JavaScript Unterstützung
  - ?

# Alternativen zu JWebUnit

- Java
  - Canoo WebTest
    - Alles in XML
  - Anteater
    - Sowohl WebTest als auch für Web-Services
    - Definition etc. in XML.
- etc

# Weitergehende Informationen

- JUnit / PHPUnit/2
  - [www.junit.org](http://www.junit.org)
  - [www.phpunit.de](http://www.phpunit.de)
- Ant
  - [ant.apache.org](http://ant.apache.org)
    - Subversion Task für Ant
      - <http://subclipse.tigris.org/svnant.html>
  - für PHP
    - [www.phing.info](http://www.phing.info)

# Weitergehende Informationen

- Java Test Tools
  - [jwebunit.sourceforge.net](http://jwebunit.sourceforge.net)
  - [httpunit.sourceforge.net](http://httpunit.sourceforge.net)
  - [webtest.canoo.com/webtest](http://webtest.canoo.com/webtest)
  - [aft.sourceforge.net](http://aft.sourceforge.net)
- SimpleTest (PHP)
  - [www.lastcraft.com/simple\\_test.php](http://www.lastcraft.com/simple_test.php)
- Xdebug Erweiterung (CCA)
  - [www.xdebug.org](http://www.xdebug.org)

# Weitergehende Informationen

- **Java Test Tools**
  - [jwebunit.sourceforge.net](http://jwebunit.sourceforge.net)
  - [httpunit.sourceforge.net](http://httpunit.sourceforge.net)
  - [webtest.canoo.com/webtest](http://webtest.canoo.com/webtest)
  - [aft.sourceforge.net](http://aft.sourceforge.net)
- **Performance Test/Functional Testing**
  - **JMeter**
    - [jakarta.apache.org/jmeter/index.html](http://jakarta.apache.org/jmeter/index.html)
  - **Grinder**
    - [grinder.sourceforge.net](http://grinder.sourceforge.net)
  - **Jameleon**
    - [jameleon.sourceforge.net](http://jameleon.sourceforge.net)

# Weitergehende Informationen

- Eclipse
  - [www.eclipse.org](http://www.eclipse.org)
- PHP und Eclipse
  - [www.phpeclipse.de](http://www.phpeclipse.de)
- Testen
  - [www.extremeprogramming.org](http://www.extremeprogramming.org)
  - <http://www.agilealliance.com/articles/AgileArtic>

# Weitergehende Informationen

- Testen / Testgetriebene Entwicklung
  - [www.extremeprogramming.org](http://www.extremeprogramming.org)
  - <http://www.agilealliance.com/articles/AgileArtic>
  - [www.testdriven.com/modules/news](http://www.testdriven.com/modules/news)
- Allgemeine Informationen
  - [de.wikipedia.org/wiki/Unit-Test](http://de.wikipedia.org/wiki/Unit-Test)

# Noch Fragen?

- EMail: [phpcon@soebes.de](mailto:phpcon@soebes.de)
- Vielen Dank für Ihre Aufmerksamkeit.