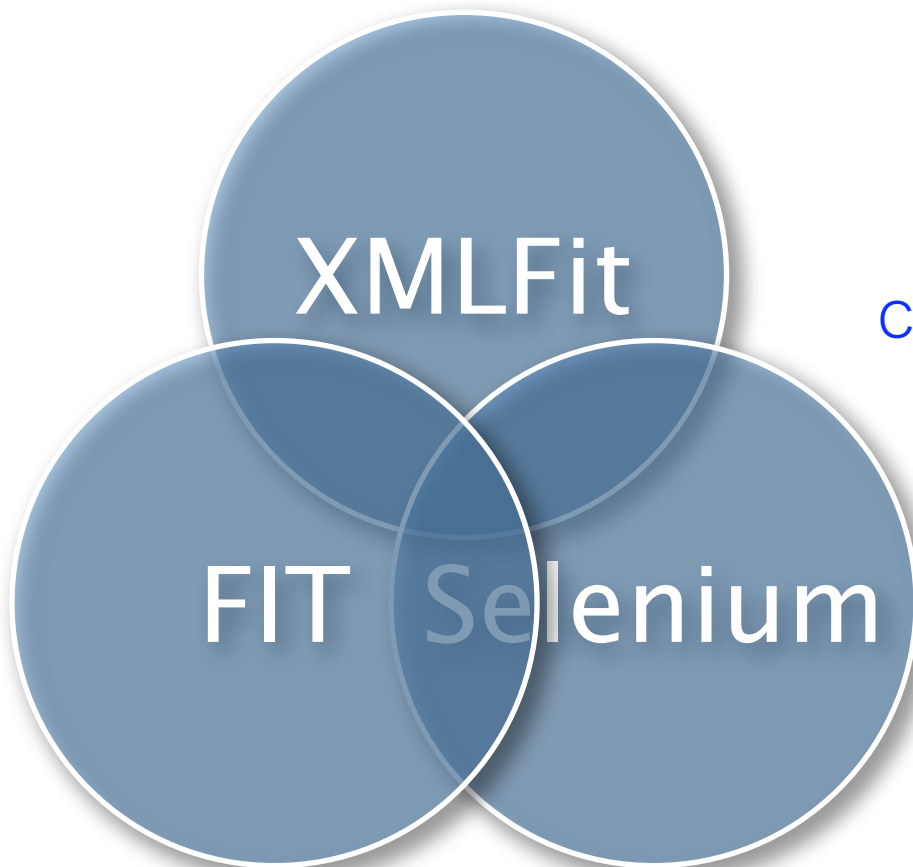


Modellgetriebene FIT-Tests

am Beispiel XMLFit

Test Tage 2009



Christian Baranowski

Christian.Baranowski@seitenbau.com

Christian Faigle

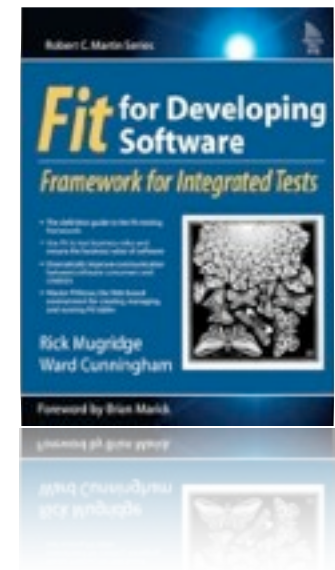
Christian.Faigle@seitenbau.com

Agenda

- Teil I – Einführung in FIT
 - Testen mit Fit Tabellen
 - Fit Tabellen und Applikation verknüpfen
- Teil II – Webapplikation testen mit Fit und Selenium
 - Einführung in Selenium
 - Selenium in Fit Tests nutzen (Selenium Fixture)
- Teil III – Einführung in XMLFit
 - Fitnessse – Testen mittels Wiki und Fit
 - XMLFit – Fit Tests mit XML DSL entwickeln
- Übungsaufgabe

Überblick Fit – Framework for Integrated Tests

- FIT wurde entwickelt von **Ward Cunningham** WIKI-Erfinder
- Motivation von Fit Tests
 - Automatisierung von **Akzeptanztests** mittels Tabellen
 - Framework für Datengetriebene Tests
 - **Data-Driven-Test** (xUnit Test Patterns – Gerard Meszaros)
- Fit Design – **Open Framework**
 - Beautiful Code beschreibt in „Fit Tests– BEAUTY THROUGH FRAGILITY“ das Design von Fit als ein Open



Testen mit Fit Tabellen

- Fit Tests für den Beispiel **Zinsrechner**

Betrag	<input type="text" value="0.0"/>
Zinssatz	<input type="text" value="0.0"/>
Laufzeit	<input type="text" value="0.0"/>
<input type="button" value="Reset"/>	<input type="button" value="Submit"/>
<input type="button" value="Reset"/>	<input type="button" value="Submit"/>

Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Betrag

10

Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Betrag	Zinssatz
10	2,5

Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Betrag	Zinssatz	Laufzeit
10	2,5	2

Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Betrag	Zinssatz	Laufzeit	Summe
10	2,5	2	10,51

Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Eingehende - Werte

Prüfung eines Ergebnis

	Betrag	Zinssatz	Laufzeit	Summe()
Testfall	10	2,5	2	10,51

Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Betrag	Zinssatz	Laufzeit	Summe()
10	2,5	2	10,51
10	10	3	25,94
10	10	10	100000
...

Testfälle



Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Applikation Anbindung (Java Klasse)

Prüfung des Ergebnis

Eingehende - Werte

onlinerechner.ZinsenFixture

Betrag	Zinssatz	Laufzeit	Summe()
10	2,5	2	10,51
10	10	3	25,94
10	10	10	100000
...

Testfälle

Testen mit Fit Tabellen

- Fit Tests als **ColumnFixture** Tabelle

Rot = Fehler in der Anwendung

onlinerechner.ZinsenFixture			
betrag	zinssatz	laufzeit	summe()
10	2.5	2	10.51 <i>expected</i>
10	2.5	2	10.50625 <i>actual</i>
10	2.5	2	10.50625
			Kein Wert
			java.lang.NumberFormatException: For input string: "Kein Wert" at sun.misc.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:1392) at java.lang.Double.parseDouble(Double.java:510) at onlinerechner.ZinsenFixture.summe(ZinsenFixture.java:210) at sun.reflect.NativeMethodAccessorImpl.invoke0(NativeMethodAccessorImpl.java:62) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:498) at org.junit.runners.model.FrameworkMethod\$1.runReflectiveCall(FrameworkMethod.java:59) at org.junit.internal.runners.model.ReflectiveCallable.run(ReflectiveCallable.java:12) at org.junit.runners.model.FrameworkMethod\$2.runInThread(FrameworkMethod.java:153) at java.lang.Thread.run(Thread.java:750)

Gelb = Ausnahme - Fehler im Test

Grün = erfolgreicher Test

Testen mit Fit Tabellen

- Fit Test als **ActionFixture** Tabelle

Befehle / Kommandos

fit.ActionFixture *Anbindung an die Applikation (java)*

start ZinsenActionFixture

enter Betrag 10 *Argumente*

enter Zinssatz 2,5

enter Laufzeit 2

press Submit

check Summe 10,51

Testfall

Objekt / Auswahl / Target

Testen mit Fit Tabellen

- Fit Test als **ActionFixture** Tabelle

fit.ActionFixture		
start	onlinerechner.ZinsenActionFixture	
enter	betrag	10
enter	zinssatz	2.5
enter	laufzeit	2
press	submit	
check	summe	10.51 <i>expected</i>
		10.50625 <i>actual</i>
check	summe	10.20832 <i>actual</i>
		10.21 <i>expected</i>

Anwendung Starten

Eingaben

Aktion - Zinsen Berechnen

Prüfung

Testen mit Fit Tabellen

- Weiteres Beispiel für eine **ActionFixture** Tabelle

fit.ActionFixture

start	jugs.Zinsrechner	
enter	Betrag	0
enter	Zinssatz	2,5
enter	Laufzeit	0
press	Submit	
check	Fehler	Kein gültiger Betrag
check	Fehler	Keine gültige Laufzeit

Testen mit Fit Tabellen

- Mengen testen mit **RowFixture** Tabellen

Jugs.TeilnehmerListe

Titel	Author
Keynote - Wie ich lernte, das Testen zu lieben	Johannes Link
Glass-Box-Test zur Testsuite-Optimierung	Rainer Schmidberger
Wertvolle Unittests durch DSL orientiertes API Design	Ulf Müller
...	...

Testen mit Fit Tabellen

- Geordnete Listen testen mit **RowFixture** Tabellen

Jugs.TeilnehmerListe

Nr	Titel	Author
1	Keynote - Wie ich lernte, das Testen zu lieben	Johannes Link
2	Glass-Box-Test zur Testsuite-Optimierung	Rainer Schmidberger
3	Wertvolle Unittests durch DSL orientiertes API Design	Ulf Müller
...		...

Testen mit Fit Tabellen

- Testen mit Sequenzen von Tabelle

→ Beispiel Fit Test ...

fit.ActionFixture

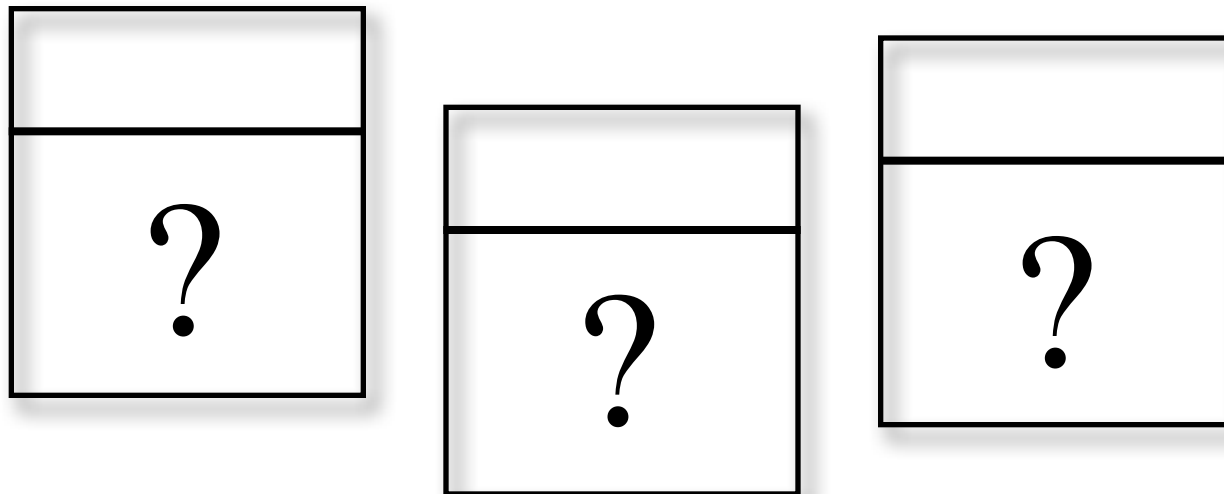
start	jugs.TestTageAnmelden	
enter	name	Baranowski
enter	Titel	Fit Tests
press	Submit	
check	einladung	true

Jugs.TeilnehmerListe

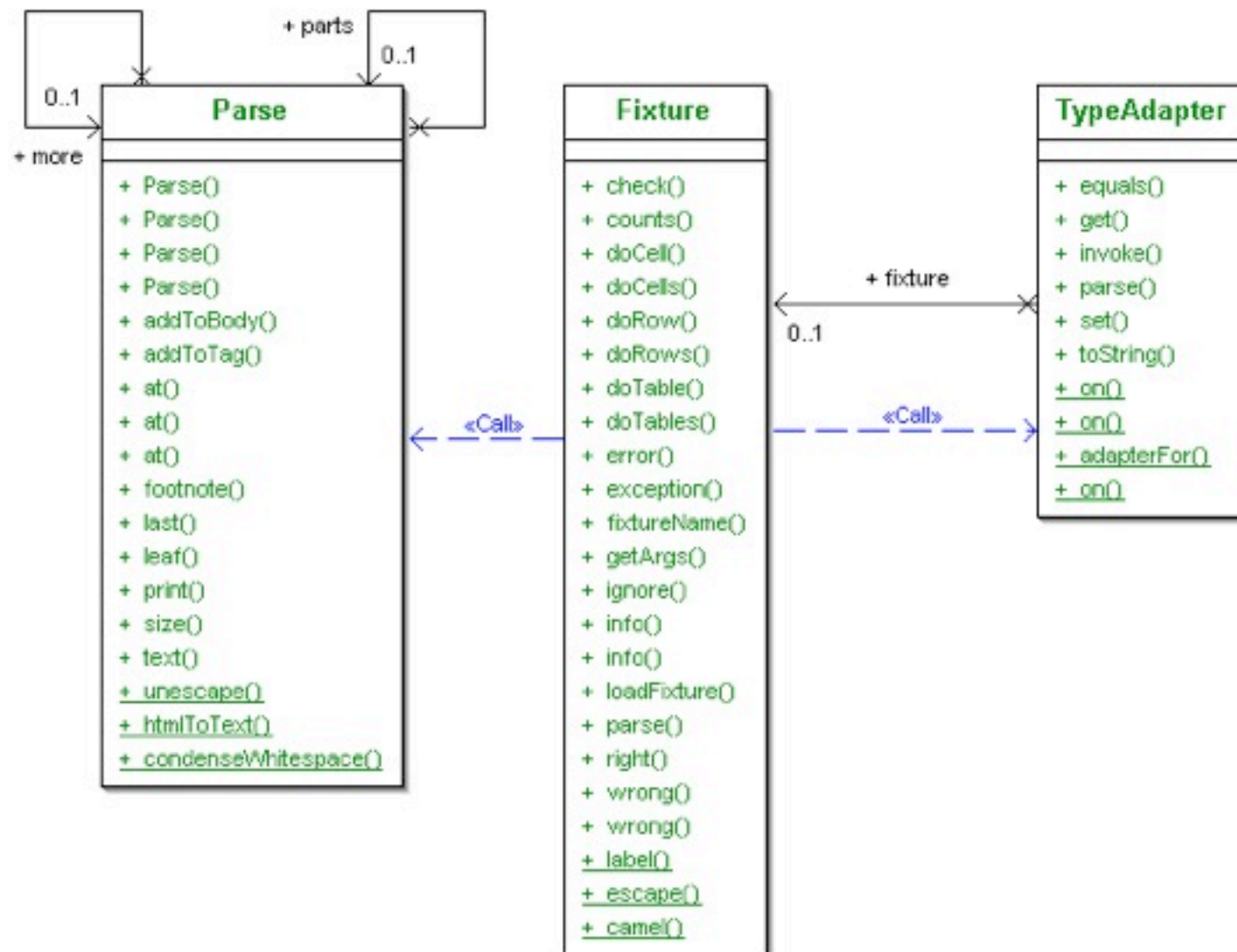
Titel	Author
Keynote - Wie ich lernte, das Testen zu lieben	Johannes Link
Glass-Box-Test zur Testsuite-Optimierung	Rainer Schmidberger
Ulf Müller	Wertvolle Unittests durch DSL orientiertes API Design
...	...

Fit Tabellen und Applikation verknüpfen

- Fit Design – drei Klassen Framework ...

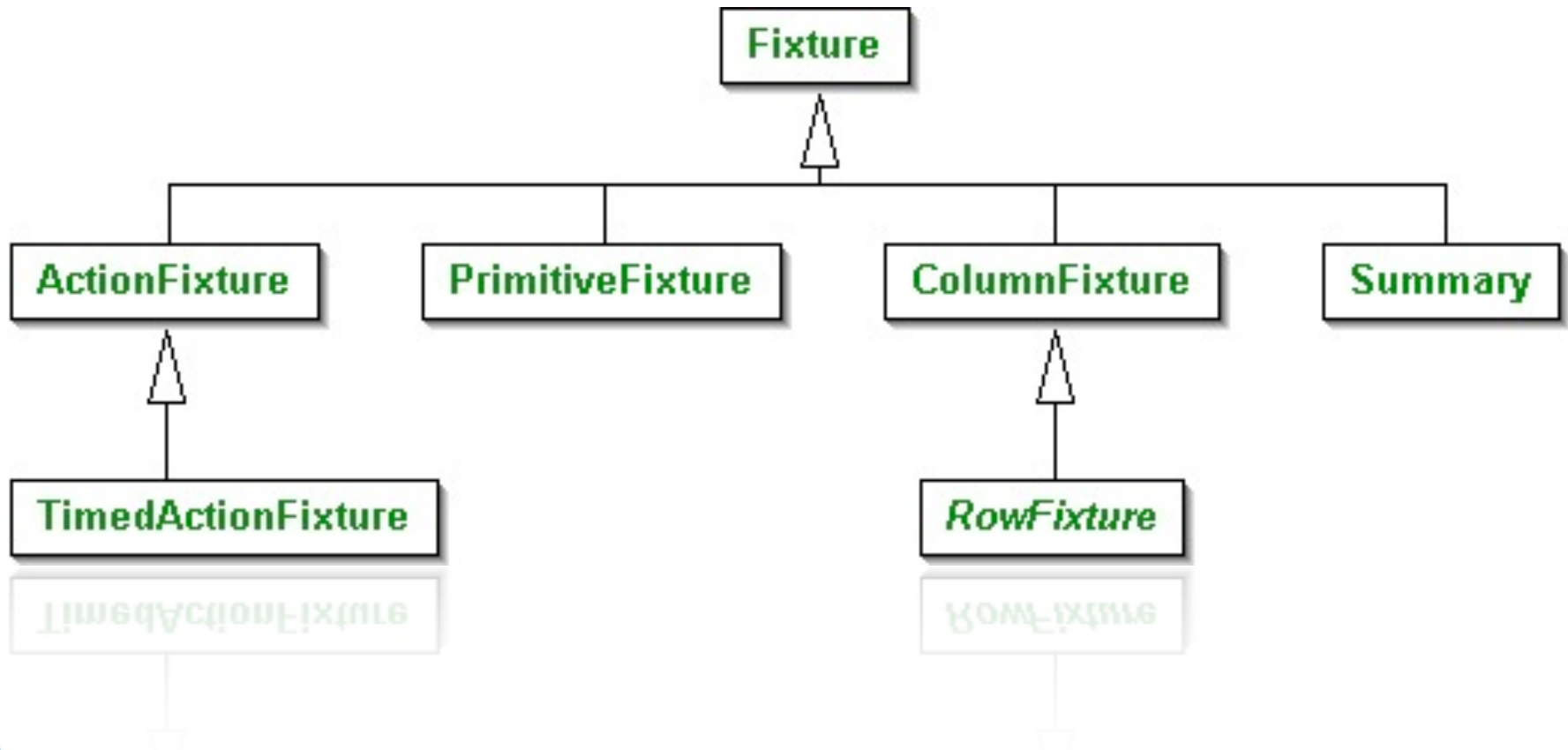


• Fit Design – drei Klassen Framework



Fit Tabellen und Applikation verknüpfen

- Fit Design – Fixtures



Fit Tabellen und Applikation verknüpfen

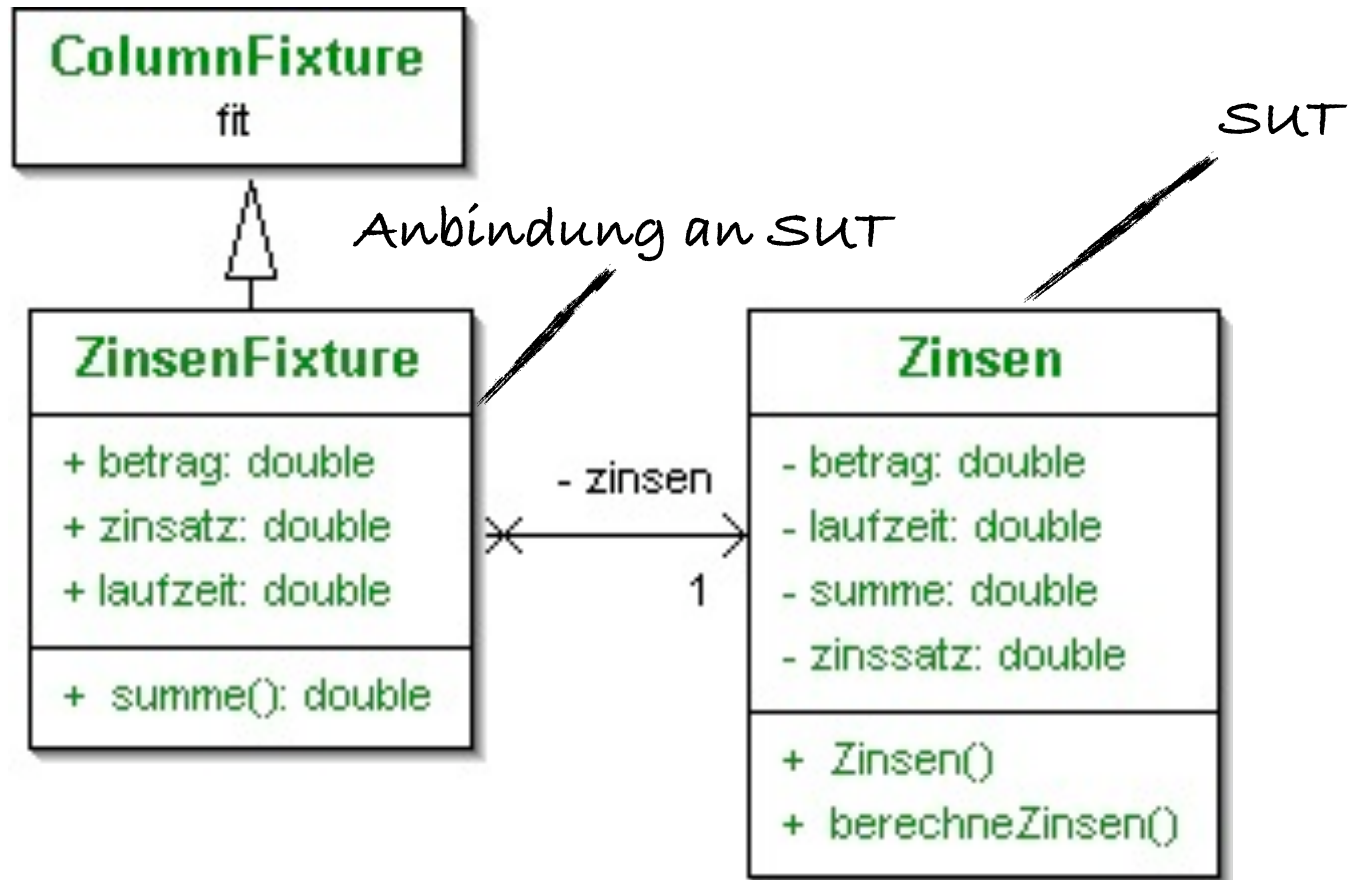
- Beispiel – ColumnFixture entwickeln

onlinerechner.ZinsenFixture

Betrag	Zinssatz	Laufzeit	Summe()
10	2,5	2	10,51
10	10	3	25,94
20	10	10	100000
...

Fit Tabellen und Applikation verknüpfen

- Beispiel – ColumnFixture



• Beispiel Code – ColumnFixture

```
public class ZinsenFixture extends ColumnFixture {  
    private Zinsen zinsen = new Zinsen();  
  
    public double betrag;  
    public double zinsatz;  
    public double laufzeit;  
  
    public double summe() {  
        zinsen.setBetrag(betrag);  
        zinsen.setLaufzeit(laufzeit);  
        zinsen.setZinssatz(zinsatz);  
        return zinsen.berechneZinsen();  
    }  
}
```

Basis Klasse
für
Fit Extension

SUT

Eingehende - Werte

Prüfung - Ausgabe

Anbindung
an SUT

Fit Tabellen und Applikation verknüpfen

- Beispiel – ActionFixture entwickeln

fit.ActionFixture

start	onlinerechner.ZinsenActionFixture	
-------	-----------------------------------	--

enter	Betrag	10
-------	--------	----

enter	Zinssatz	2,5
-------	----------	-----

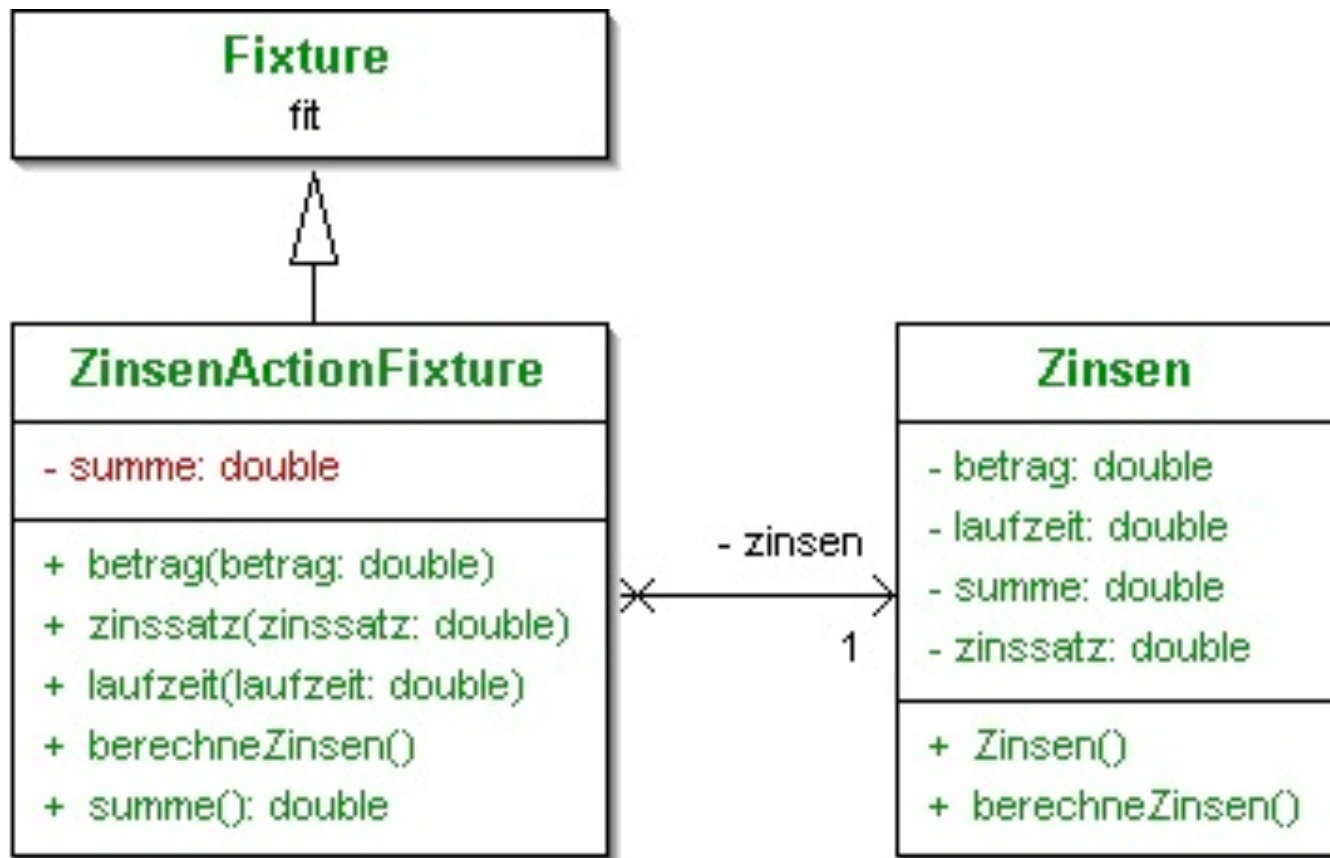
enter	Laufzeit	2
-------	----------	---

press	berechneZinsen	
-------	----------------	--




check	Summe	10,51
-------	-------	-------

Fit Tabellen und Applikation verknüpfen

- Beispiel – ActionFixture entwickeln



• Beispiel Code – ActionFixture

```
public class ZinsenActionFixture extends Fixture {  
    Zinsen zinsen = new Zinsen();  
    double summe;  
  
    public void betrag(double betrag) {  
        zinsen.setBetrag(betrag);  Eingaben / Enter  
    }  
  
    ...  
  
    public void berechneZinsen() {  Aktion / Press  
        summe = zinsen.berechneZinsen();  
    }  
  
    public double summe() {  
        return summe;  Ausgabe Prüfung / Check  
    }  
}
```


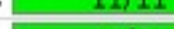


Fit Test Abdeckung

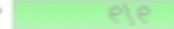
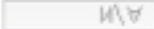
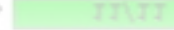
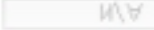
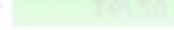
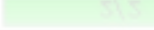




- Statement Coverage
- Branch Coverage
- Werkzeuge
 - Emma
 - Cobertura

Fit Test Abdeckung

Coverage Report - onlinerechner

Package /	# Classes	Line Coverage	Branch Coverage	Complexity
onlinerechner	7	25%  33/132	17%  2/12	1,13

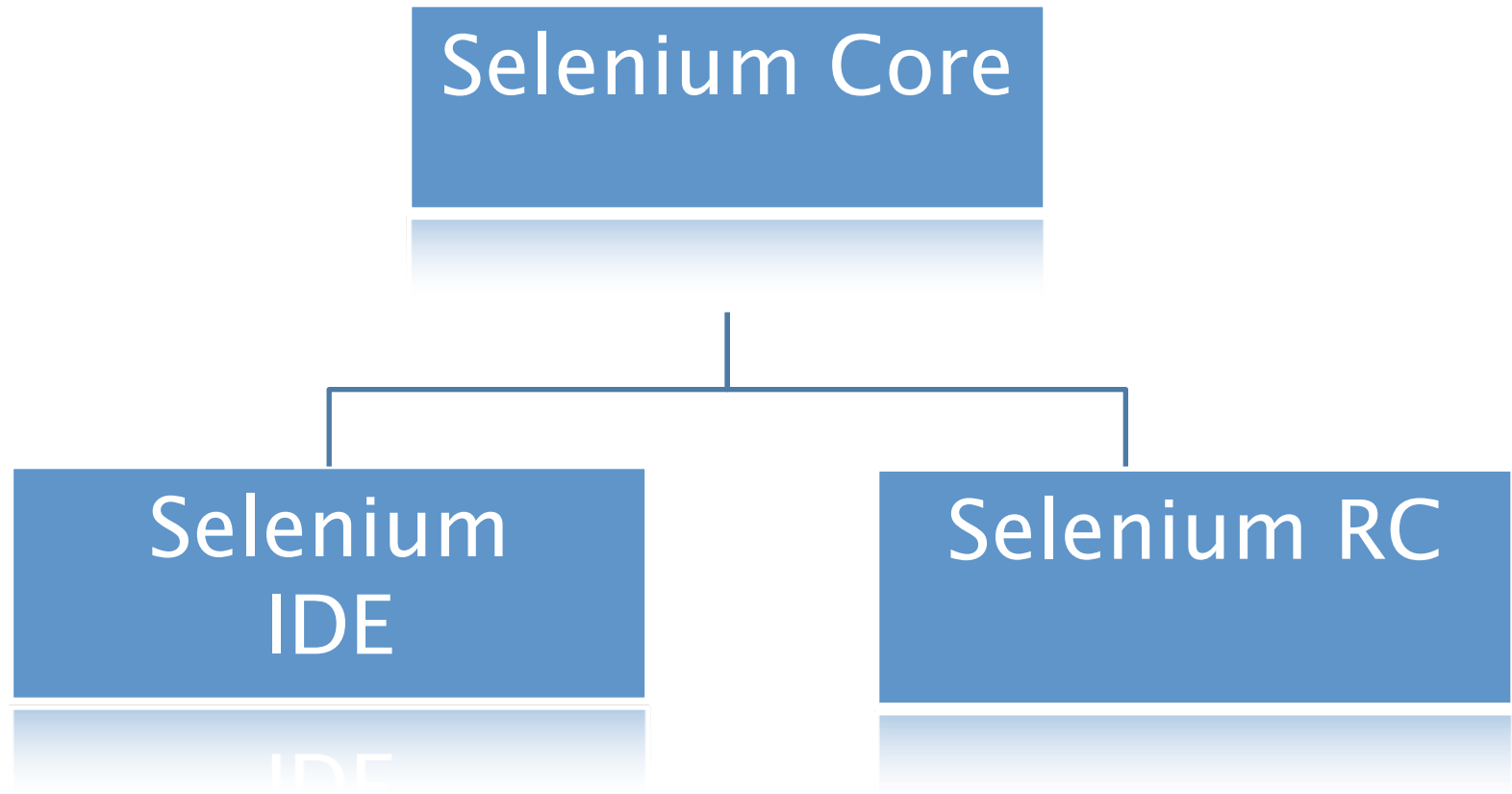
Classes in this Package /	Line Coverage	Branch Coverage	Complexity
AutoKosten	0%  0/51	0%  0/2	1,087
HypoZinsen	0%  0/35	0%  0/4	1,154
WaehrungsRechner	0%  0/5	N/A  N/A	1
WaehrungsServer	0%  0/4	0%  0/4	3
Zinsen	80%  16/20	100%  2/2	1,111
ZinsenActionFixture	100%  11/11	N/A  N/A	1
ZinsenFixture	100%  6/6	N/A  N/A	1

ZinsenFixture	100%	 6/6	N/A	 N/A	1
ZinsenActionFixture	100%	 11/11	N/A	 N/A	1
Zinsen	80%	 16/20	100%	 2/2	1,111
WaehrungsServer	0%	 0/4	0%	 0/4	3
WaehrungsRechner	0%	 0/5	N/A	 N/A	1

Agenda

- Teil I – Einführung FIT
 - Testen mit Fit Tabellen
 - Fit Tabellen und Applikation verknüpfen
- Teil II – Webapplikation testen Fit und Selenium
 - Einführung in Selenium
 - Selenium in Fit Tests nutzen (Selenium Fixture)
- Teil III – Einführung in XMLFit
 - Fitnessse – Testen mittels Wiki und Fit
 - XMLFit – Fit Tests mit XML DSL entwickeln

Einführung in Selenium



Selenium Core

- Selenium Core – Beispiel Zinsrechner

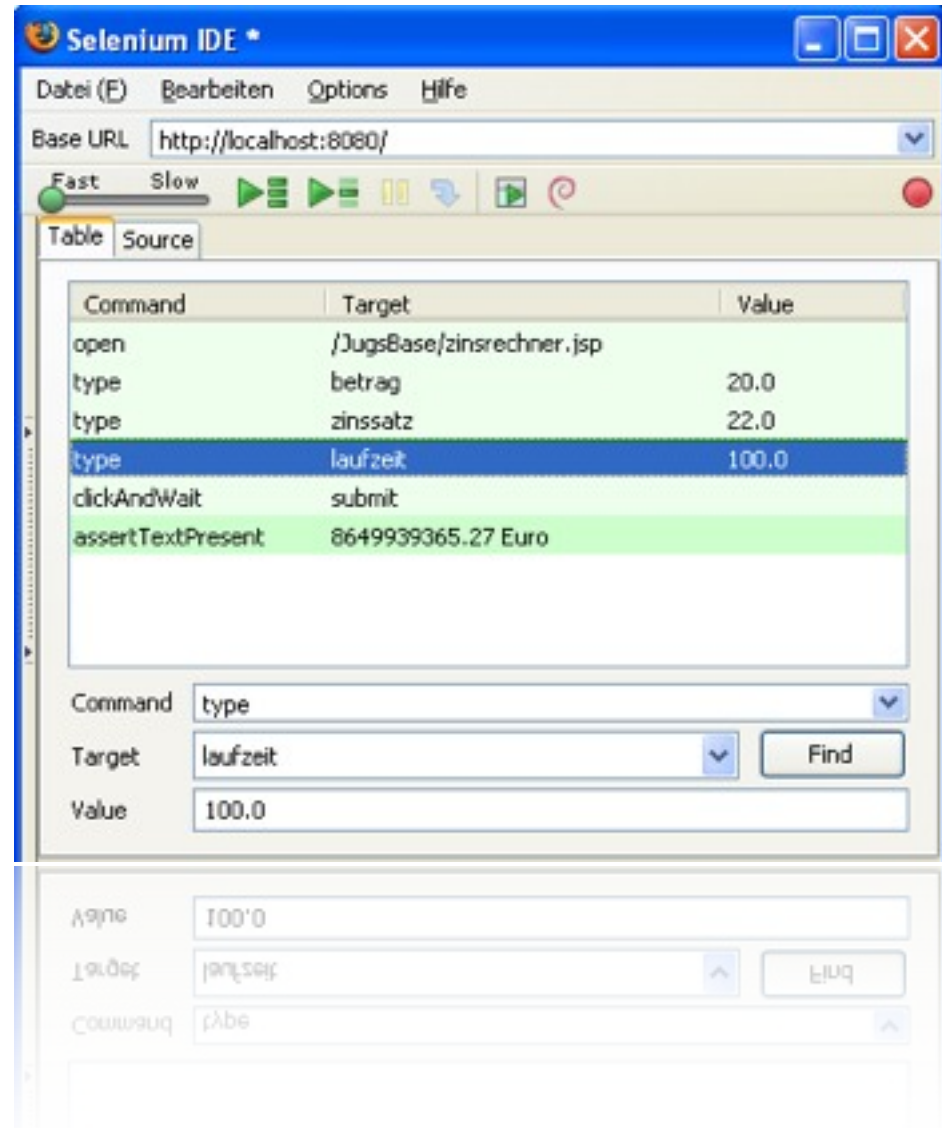
Action - Tabelle

Selenium Befehl

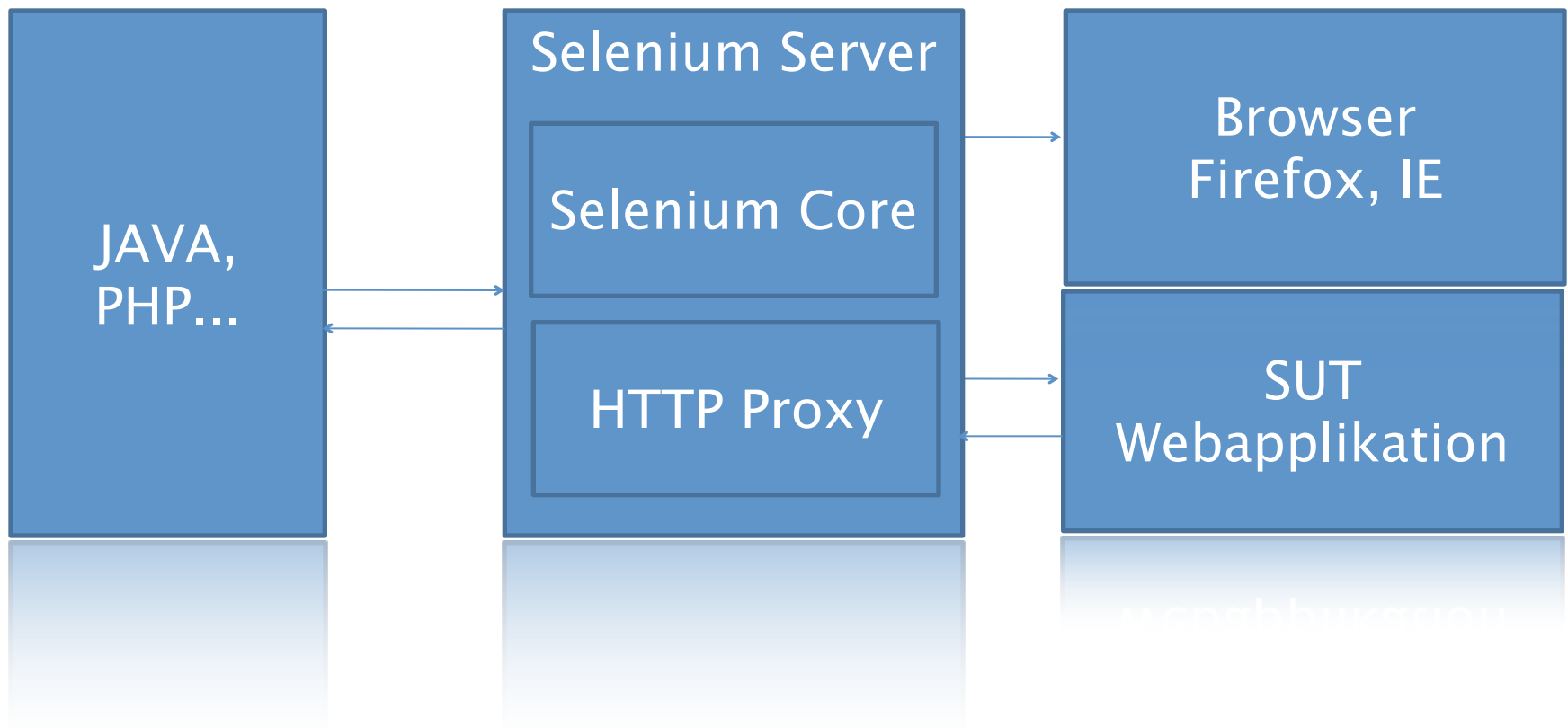
open	/JugsBase/zinsrechner.jsp		
type	betrag	20.0	
type	zinsatz	Target 22.0	Argumente
type	laufzeit	100.0	
clickAndWait	submit		
assertTextPresent	8649939365.27 Euro		

Selenium IDE

- Selenium IDE–
Capture &
Replay Tests



Selenium Remote Control



Selenium Remote Control

- Selenium RC– JUnit Beispiel Zinsrechner

```
@Test
public void testZinsenBerechnen() throws Exception {
    selenium.open("/JugsBase/zinsrechner.jsp");
    selenium.type("betrag", "10.0");
    selenium.type("zinssatz", "10.0");
    selenium.type("laufzeit", "5.0");
    selenium.click("submit");
    selenium.waitForPageToLoad("30000");
    assertTrue(selenium.isTextPresent("16.11 Euro"));
}
```

Selenium Remote Control

- Fit und Selenium RC mittels allgemeinem Selenium Fixture

fit. SeleniumFixture

open	/JugsBase/zinsrechner.jsp	
type	betrag	10.0
type	zinssatz	10.0
type	laufzeit	5.0
clickAndWait	submit	
assertTextPresent	16.11 Euro	

Selenium Remote Control

- Selenium RC– Selenium Fixture für Fit

```
...  
public void type(Argument selektor, Argument arg) throws  
Exception {  
    try  
    {  
        selenium.type(selektor.text(), arg.text());  
    } catch(Exception e) {  
        exception(selektor, e);  
        return;  
    }  
    right(selektor); right(arg);  
}  
...
```

Selenium Remote Control

- Selenium RC in Fit Fixture – Beispiel ColumnFixture

onlinerechner.SeleniumZinsenFixture

Betrag	Zinssatz	Laufzeit	Summe()
10	2,5	2	10,51
10	10	3	25,94
20	10	10	100000
...

Selenium Remote Control

- Selenium RC– Fit Beispiel Zinsrechner

...

```
public String summe() {  
    selenium.open("/JugsBase/zinsrechner.jsp");  
    selenium.type("betrag", String.valueOf(betrag));  
    selenium.type("zinssatz", String.valueOf(zinssatz));  
    selenium.type("laufzeit", String.valueOf(laufzeit));  
    selenium.click("submit");  
    selenium.waitForPageToLoad("30000");  
    String summe = selenium.getText(  
        "//table[@id='zinsergebnis']//tr[4]/td[2]");  
    return summe;  
}
```

...

Agenda

- Teil I – Einführung FIT
 - Testen mit Fit Tabellen
 - Fit Tabellen und Applikation verknüpfen
- Teil II – Webapplikation testen Fit und Selenium
 - Einführung in Selenium
 - Selenium in Fit Tests nutzen (Selenium Fixture)
- Teil III – Einführung in XMLFit
 - Fitnessse – Testen mittels Wiki und Fit
 - XMLFit – Fit Tests mit XML DSL entwickeln

Einführung in XMLFit

- Warum Fit als Framework für Systemtests?
 - **Nachvollziehbarkeit:** Testdaten und Testablauf sind nicht getrennt!
 - **Lesbarkeit:** Tests können von der Fachabteilung verstanden werden und mit formuliert werden...
 - Tests können zur Kommunikation mit dem Kunden dienen
 - **Plattformunabhängig:** Testlogik (Implementierung) ist Plattformunabhängig
 - Testergebnis kann direkt als Testprotokoll genutzt werden
 - ...

Einführung in XMLFit

- **Nachteile von Fit als Motivation**

- Tests mittels HTML formulieren ist sehr umständlich und unübersichtlich
- HTML mittels WYSIWYG schwierig
- Wiederverwendbare Komponenten, z.B. Login, nicht einfach möglich
- Testabläufe mit verschiedenen Testdaten nicht einfach modellierbar
- ...


Einschub – Überblick Fitness

- Fitness – Testen mittels Wiki und Fit



```
|ZinsenTestFixture|  
|betrag|zinsatz|laufzeit|summe?|  
|10|2.5|2|10.50625|  
|10|2.5|2|10|
```

Einschub – Überblick Fitnessse



- Test
- Edit
- Properties
- Refactor
- Where Used
- Search
- Files
- Versions
- Recent Changes
- User Guide
- Test Guide
- Recent Changes
- Versions

ZinsTest

Assertions: 1 right, 1 wrong, 0 ignored, 0 exceptions

classpath: D:\develop\workspace-all\jugsBase\WebRoot\WEB-INF\classes

ZinsenTestFixture[?]			
betrag	zinsatz	laufzeit	summe?
10	2.5	2	10.50625
10	2.5	2	<div style="text-align: left; padding-left: 5px;"> 10 <i>expected</i> <hr style="width: 50%; margin: 2px 0;"/> 10.50625 <i>actual</i> </div>

[Front Page](#) | [User Guide](#)
[SetUp\[?\]](#), [TearDown\[?\]](#) for this page | [root](#) (for global !path's, etc.)

Einschub – Überblick Fitnessse

- Vorteile von Fitnessse

- DSL für HTML Tests einfacher klassische HTML Fit Tests
- WIKI, kann von verteilten Teams genutzt werden
- Test Komponenten durch Page include
- ...

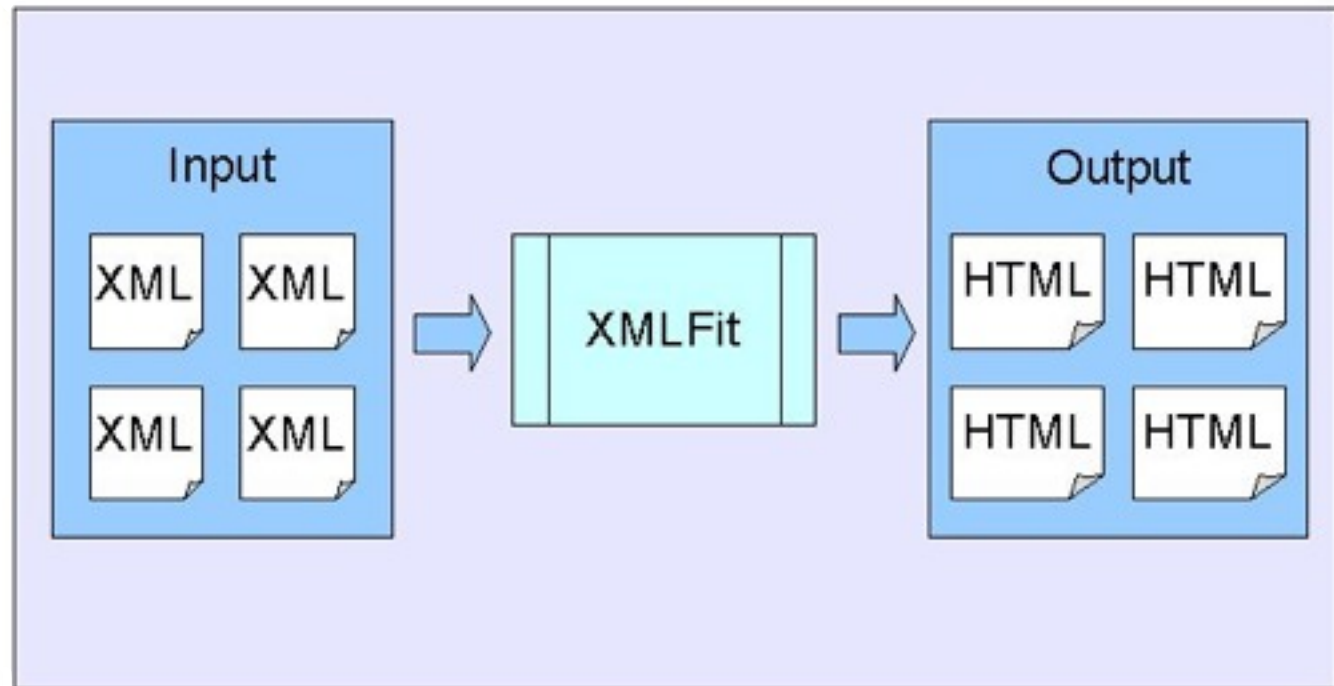
Einschub – Überblick Fitnessse

- **Nachteile von Fitnessse**

- Debugging der Fixture und der Tests nur remote möglich
- Fixture bereitstellen
- Kein so einfacher Test Report der archiviert werden kann
- Hosting von Fitnessse
- Testabdeckung kann nur schwierig gemessen werden
- Für Daten getriebene Entwicklertests / Komponententests nicht geeignet
- ...

Einführung in XMLFit

- XMLFit einen XML basierte DSL für Fit Tests

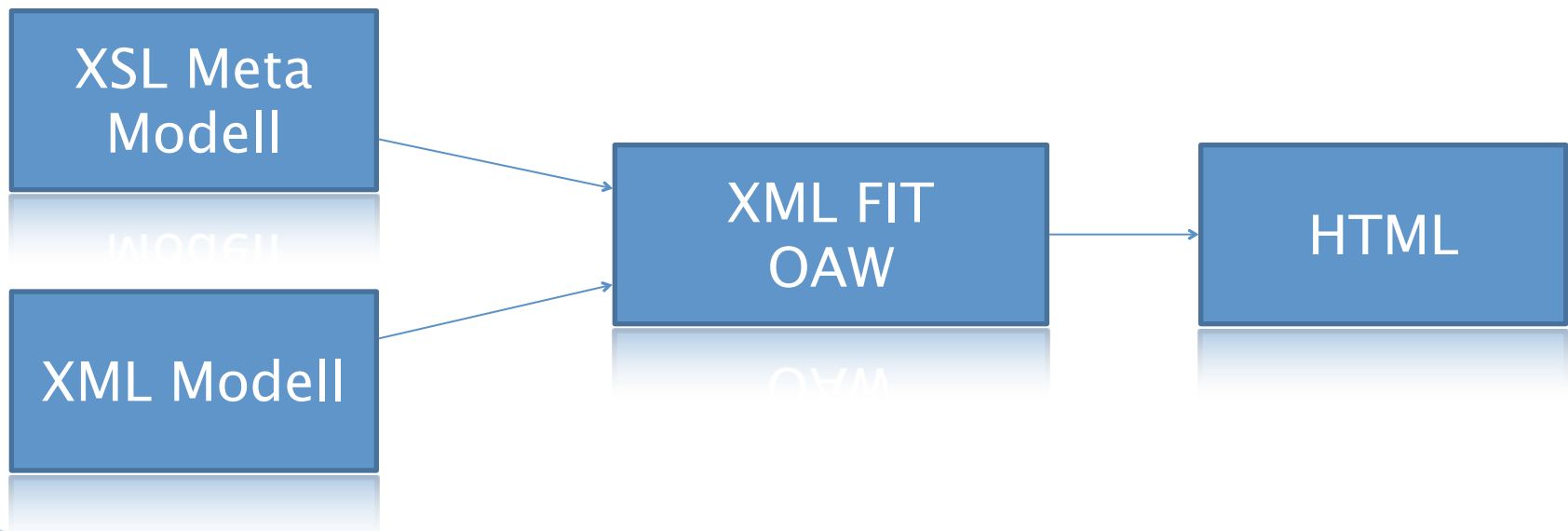


Einführung in XMLFit

- XMLFit einen XML basierte DSL für Fit Tests
 - DSL aus der die HTML Tests generiert werden
 - XML da gute Tool-Unterstützung
 - Aus Testdaten mehrere Fit Tests generieren über Variablen und DatenSets
 - Simpler Ansatz zur Testerzeugung
 - Dateibasierter Ansatz – XMLFit Tests können mittel SVN oder CVS verwaltet werden...
 - ...

XMLFit Design

- **Version 0.0.1**
 - XSLT basierende Transformation
- **Version 0.0.2**
 - **OAW basierende Transformation**



Einführung in XMLFit

- Einfaches Beispiel Zinsrechner:

fit.SeleniumFixture		
open	/JugsBase/zinsrechner.jsp	
type	betrag	5.0
type	zinssatz	5.0
type	laufzeit	10.0
clickAndWait	submit	
assertTextPresent	25.94 Euro	

Einführung in XMLFit

- Einfaches Beispiel Zinsrechner:

```
<actionFixture type="fit.SeleniumFixture">
  <command name="open">
    <target>/JugsBase/zinsrechner.jsp</target>
  </command>
  <command name="type">
    <target>betrag</target>
    <value>5.0</value>
  </command>
  ...
  <command name="assertTextPresent">
    <target>25.94 Euro</target>
  </command>
</actionFixture>
```

Einführung in XMLFit

- Einfaches Beispiel Zinsrechner:

```
...  
<test id="001" name="Zinsen">  
    <testStep file="example-zinsrechner.xml" />  
</test>  
...
```

Einführung in XMLFit

- Beispiel Zinsrechner mit getrennten Testdaten:

```
<actionFixture type="fit.SeleniumFixture">
...
    <command name="type">
        <target>betrag</target>
        <value>${betrag}</value>
    </command>
...
</actionFixture>
```

- Beispiel Zinsrechner mit getrennten Testdaten:

```
<test id="002" name="Zinsen">
  <testStep file="example-zinsrechner2.xml">
    <property name="betrag" value="10"/>
    <property name="zinssatz" value="5"/>
    <property name="laufzeit" value="10"/>
    <property name="summe" value="24"/>
  </testStep>
  <testStep file="example-zinsrechner2.xml">
    <property name="betrag" value="20"/>
    <property name="zinssatz" value="50"/>
    <property name="laufzeit" value="110"/>
    <property name="summe" value="241"/>
  </testStep>
</test>
```

Einführung in XMLFit

- Beispiel Zinsrechner mit getrennten Testdaten:

```
...  
<test id="002" name="Zinsen">  
  <testStep  
    file="example-zinsrechner2.xml"  
    data="example-data-zins.xml"  
  />  
</test>  
...
```

Einführung in XMLFit

- Beispiel Zinsrechner mit getrennten Testdaten:

```
<data>  
  <dataSet>  
    <betrag>10</betrag>  
    <laufzeit>10</laufzeit>  
    <zinssatz>11</zinssatz>  
    <summe>110</summe>  
  </dataSet>  
</data>
```


Einführung in XMLFit

- Mehrere Tests als Tabellen Sequenzen aus Daten generieren

```
<test id="004" name="Zinsen">
  <testStep
    file="example-zinsrechner2.xml"
    data="example-data-zins.xml" />
  <testStep file="example-zinsrechner2.xml">
    <property name="betrag" value="10"/>
    <property name="zinssatz" value="5"/>
    <property name="laufzeit" value="10"/>
    <property name="summe" value="24"/>
  </testStep>
</test>
```

Agenda

- Teil I – Einführung FIT
 - Testen mit Fit Tabellen
 - Fit Tabellen und Applikation verknüpfen
- Teil II – Webapplikation testen Fit und Selenium
 - Einführung in Selenium
 - Selenium in Fit Tests nutzen (Selenium Fixture)
- Teil III – Einführung in XMLFit
 - Fitnessse – Testen mittels Wiki und Fit
 - XMLFit – Fit Tests mit XML DSL entwickeln
- Übungsaufgabe

Einführung in Übungsaufgabe

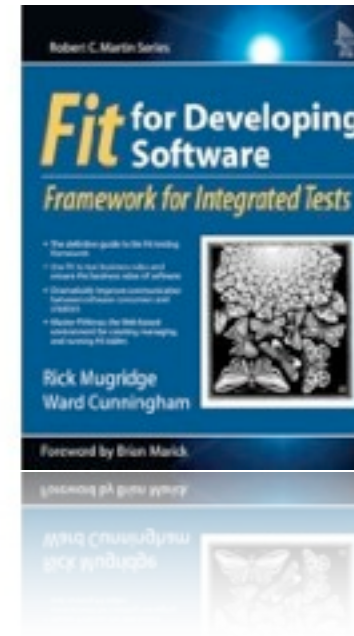
- Eclipse Projekt einrichten mit Maven
„mvn eclipse:eclipse“
- XMLFit Tests oder Fit Tests ausführen mit
„mvn test“
- Firefox Selenium IDE einrichten
- Selenium RC Server starten
- ...

Fragen & Diskussion

?

Referenzen

- Fit for Developing Software – Rick Mugridge und Ward Cunningham



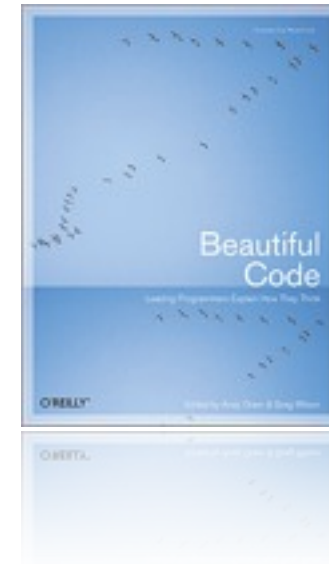
Referenzen

- xUnit Test Patterns (Refactoring Test Code)– Gerard Meszaros



Referenzen

- Beautiful Code – Michael Feathers: Framework for Integrated Test: Beauty through Fragility, et al.



Werkzeuge

Links zu Werkzeugen:

FIT – <http://fit.c2.com/>

Selenium – <http://seleniumhq.org/>

JUnit 4.X – <http://www.junit.org/>

Fitnessse – <http://fitnessse.org/>

OAW – <http://www.openarchitectureware.org/>

NVU – WYSIWYG HTML Editor: <http://net2.com/nvu/download.html>

XMLFIT – <http://xmlfit.sourceforge.net/>

Apache Maven – <http://maven.apache.org/>

Cobertura – <http://cobertura.sourceforge.net/>

EclEmma – <http://www.eclemma.org/>

Eclipse – <http://www.eclipse.org>