

# Seminar 1919 Themenliste Sommer 2018

Daniela Keller, Jörg Hagemann

5. März 2018

Die zu den jeweiligen Themen angegebene Literatur soll einen ersten Einstieg in das Thema ermöglichen. Es ist notwendig, weitere Literatur zu Rate zu ziehen. Die Literaturliste am Ende der Ausarbeitung sollte mindestens zehn Einträge enthalten. Sollte das bei bestimmten Themen nicht möglich sein, wird um Absprache mit den Betreuenden gebeten. Die angegebene Literatur kann verwendet werden, muss aber nicht.

## **T1: JUnit v5.0 und TestNG**

Diese Ausarbeitung soll sich mit dieser neuen JUnit-Version beschäftigen, die Besonderheiten herausarbeiten und bewerten.

TestNG, ein Framework zum Testen von Java-Programmen, hat als Grundlage Konzepte aus JUnit und NUnit, ergänzt um neue Funktionalitäten. Die Ausarbeitung soll die Besonderheiten hervorheben, das Framework soll ausprobiert werden, seine Vor- und -Nachteile beschrieben werden.

### **Literatur:**

<http://www.codeaffine.com/2016/02/18/junit-5-first-look/>

<http://junit.org/junit5/>: Hieraus stammt der folgende Text:

'Unlike previous versions of JUnit, JUnit 5 is composed of several different modules from three different sub-projects.

*JUnit 5 = JUnit Platform + JUnit Jupiter + JUnit Vintage*

The JUnit Platform serves as a foundation for launching testing frameworks on the JVM. It also defines the TestEngine API for developing a testing framework that runs on the platform. Furthermore, the platform provides a Console Launcher to launch the platform from the command line and build plugins for Gradle and Maven as well as a JUnit 4 based Runner for running any TestEngine on the platform. JUnit Jupiter is the combination of the new programming model and extension model for writing tests and extensions in JUnit 5. The Jupiter sub-project provides a TestEngine for running Jupiter based tests on the platform.

JUnit Vintage provides a TestEngine for running JUnit 3 and JUnit 4 based tests on the platform.'

<http://blog.takipi.com/junit-vs-testng-which-testing-framework-should-you-choose/>

<http://testng.org/doc/>

<http://www.asjava.com/testng/testng-tutorials/>

<https://www.ibm.com/developerworks/library/j-cq08296/>

**T2: Mocking-Framework für Java: Mockito, PowerMock**

In der Ausarbeitung soll behandelt werden, was man unter Mocking versteht und wofür Mocking-Frameworks sinnvoll verwendet werden können. Es sollen die Mocking-Frameworks Mockito und PowerMock vorgestellt und ihre wesentlichen Eigenschaften beschrieben werden. Am Ende soll die Arbeit mit den Frameworks bewertet werden.

**Literatur:**

- Mocks Aren't Stubs, Martin Fowler, 2007  
<https://martinfowler.com/articles/mocksArentStubs.html>
- TestDouble, Martin Fowler, <https://martinfowler.com/bliki/TestDouble.html>
- Wer braucht eigentlich ein Mocking-Framework? Erläuterung und Powerpoint-Folien, s. [/https://entwinklertag.de/karlsruhe/2016/wer-braucht-eigentlich](https://entwinklertag.de/karlsruhe/2016/wer-braucht-eigentlich)
- Robert C. Martin. The Little Mocker. 2014. Dialog über Mocks und Co mit Beispielen: <http://blog.8thlight.com/blog/>, dort unter 'Testen' entsprechende Artikel suchen
- <http://site.mockito.org/>
- sowie das Buch:  
**Mockito Cookbook** von Marcin Grzejszczak,  
June 2014, ISBN 139781783982745, Paperback, 284 pages

**Buchbeschreibung:**

'Mockito is a testing framework that allows you to easily create test doubles (mocks) of the system under test's collaborators. You can use mocks to better simulate the behavioral patterns of your software in isolation to receive feedback from tests as soon as possible.

This book consists of recipes that depict the usage of a vast majority of Mockito functionalities in real-life examples. It goes beyond the documentation and covers how Mockito works and shows the steps to write effective tests using Mockito. This is a focused guide with lots of practical recipes of business issues and presentation of the entire test of the system. This book shows the use of Mockito's popular unit testing frameworks such as JUnit, PowerMock, TestNG, and so on.'

- <https://github.com/powermock/powermock/wiki>, <http://powermock.github.io/>
- <https://blog.codecentric.de/2017/05/junit-tests-implementieren-mit-mockito-und-powermock/> Achtung: Link geteilt

- Buch: **Instant Mock Testing with PowerMock**, Deep Shah, October 2013

**Buchbeschreibung:**

Writing good unit tests can be hard in situations where we have to deal with static, final, or private methods. Sometimes, even a good design is sacrificed for the sake of testability. A mocking framework's ability to create mocks for almost any scenario enables us to focus on accurately programming for a business environment.

Starting with a basic example of how to create a mock and verify a method invocation Instant Testing with PowerMock then demonstrates various features of PowerMock using clear, practical,

step-by-step recipes. By the end of this book, you will be able to write unit tests for almost any scenario and suppress unwanted behaviour to effectively unit test your code.

Moving on from the basics, you will learn to write unit tests for static, final, and private methods, and write flexible unit tests using argument matchers. Following on from this, you will also learn how to mock a few methods of a class, and invoke real implementation of others using partial mocks or spies. This book will teach you the tricks of the trade that will enable you to write good unit tests for a wide range of scenarios.

### **T3: Was liefern JUnit-Erweiterungen wie Cactus (JUnit für Java EE), HttpUnit, DBUnit, Infinitest, DJUnit + Co?**

Die Ausarbeitung soll die verschiedenen Erweiterungen kurz vorstellen und die jeweiligen Besonderheiten herausarbeiten. Eine Erweiterung soll ausprobiert und bewertet werden. Es sollen mindestens drei Erweiterungen besprochen werden, welche aus der Liste unten, aber auch beliebig aus der Literatur ausgewählt werden können.

- Cactus als Junit für Java EE
- HttpUnit für Webentwicklung,
- DBUnit für Datenbankentwicklungen,
- Infintest für kontinuierliches Testen,
- DJUnit für die Berechnung von Code-Abdeckung
- PHPUnit

#### **Literatur:**

<http://jakarta.apache.org/cactus/>

<https://javaeeblog.wordpress.com/2016/06/15/java-ee-im-test-junit-dbunit-arquillian-jailer/>

[https://zaunberg.de/wp-content/uploads/2012/09/JM\\_Arquillian.pdf](https://zaunberg.de/wp-content/uploads/2012/09/JM_Arquillian.pdf)

<http://httpunit.sourceforge.net/>

<http://www.torsten-horn.de/techdocs/java-httpunit.htm>

<http://dbunit.sourceforge.net/>

<http://dbunit.sourceforge.net/howto.html>

<https://infinitest.github.io>

<http://realsearchgroup.org/SEMaterials/tutorials/djunit/>

<https://phpunit.de/>

### **T4: Spock - Test-und Spezifikationsframework für die Programmiersprachen Java und Groovy**

Die Ausarbeitung soll das Framework vorstellen, darlegen, warum man es benutzen sollte oder auch nicht, und insbesondere auf *Data Driven Testing* eingehen. Wenn es vom Umfang her passt, kann auch noch auf die Sprache Groovy eingegangen werden.

#### **Literatur:**

Peter Niederwieser, **The Spock Framework Team** version 1.1 ,

<http://spockframework.org/>:

'Spock is a testing and specification framework for Java and Groovy applications. What makes it stand out from the crowd is its beautiful and highly expressive specification language. Thanks to its JUnit runner, Spock is compatible with most IDEs, build tools, and continuous integration servers. Spock is inspired from JUnit, jMock, RSpec, Groovy, Scala, Vulcans, and other fascinating life forms.'

### **T5: RSpec: Behaviour Driven Development for Ruby. Making TDD Productive and Fun**

RSpec wurde in der Programmiersprache Ruby zum Testen von Ruby-Code geschrieben. Die grundlegende Idee dahinter ist *Test Driven Development*. RSpec enthält ein eigenes Mocking Framework. Es soll eines der populärsten Testtools für Ruby-Anwendungen sein.

Die Ausarbeitung soll sehr kurz die Programmiersprache Ruby vorstellen und sich dann intensiv mit RSpec beschäftigen. Das Tool muss ausprobiert werden. Eine Bewertung des Tools soll die Ausarbeitung abschließen.

#### **Literatur:**

- <http://rspec.info/>
- Myron Marston and Ian Dees: **Effective Testing with RSpec 3: Build Ruby Apps with Confidence.**

#### **Buchbeschreibung:**

'This definitive guide from RSpec's lead developer shows you how to use RSpec to drive more maintainable designs, specify and document expected behavior, and prevent regressions during refactoring. Build a project using RSpec to design, describe, and test the behavior of your code-whether you're new to testing tools or an experienced developer.'

### **T6: Tools zum automatisierten Testen von Webanwendungen: Selenium**

Die Ausarbeitung soll kurz erklären, was die Besonderheiten beim Testen von Webanwendungen sind. Dann soll das Tool Selenium ausführlich beschrieben werden. Dazu sollte es auch ausprobiert werden. Eine persönliche Bewertung soll die Arbeit abrunden.

#### **Literatur:**

[http://greiterweb.de/spw/test\\_werkzeuge.htm](http://greiterweb.de/spw/test_werkzeuge.htm)

<http://www.seleniumhq.org/>

und

Buch dazu: **Michael Kain: Selenium. Web-Applikationen automatisiert testen**, Open Source Press, ISBN 978-3-937514-57-4

#### **Buchbeschreibung Amazon:**

'Bei einer komplexen Webanwendung ist es nahezu unmöglich, alle Aktionen des potentiellen Anwenders manuell nachzuvollziehen und miteinander zu kombinieren. Automatisierte Tests können also viel Zeit sparen, die Qualität der Anwendung verbessern und dabei helfen, stupide Tätigkeiten zu vermeiden.'

Selenium ist eine Open-Source-Software, die solche Tests ermöglicht. Es ist systemunabhängig und kann sogar mittels SeleniumIDE in den Firefox-Browser eingebunden werden.

Dieses Buch beschreibt nicht nur die Benutzung von Selenium und der SeleniumIDE, sondern vermittelt auch Praxiswissen zum Planen und Durchführen von Tests. Damit richtet es sich auch an

Webentwickler, die über wenig Erfahrung mit automatisierten Testumgebungen verfügen.'

### **T7: Tools zum automatisierten Testen: HP QuickTest Professional (QTP 11.0)**

In dieser Ausarbeitung soll sich intensiv mit diesem Testwerkzeug beschäftigt werden. Laut Internetseite gibt es eine 30 Tage lang gültige Testversion. Das Tool soll verständlich beschrieben werden, seine Vor- und möglichen Nachteile herausgearbeitet werden. Eine persönliche Bewertung soll die Arbeit abschließen.

#### **Literatur:**

[http://greiterweb.de/spw/test\\_werkzeuge.htm](http://greiterweb.de/spw/test_werkzeuge.htm)

<http://www.learnqtp.com/qtp-11-available-for-download-hp-functional-testing-11/>

#### **Beschreibung des Tools:**

<https://www.learnqtp.com/what-is-qtp/>

'Unified Functional Testing (QuickTest Professional), popularly known by its acronym UFT is an automation testing tool originally from Mercury Interactive which was acquired by Hewlett Packard (HP) in 2006. UFT is primarily used for functional, regression and service testing. Using UFT, your HP QTP can automate user actions on a web or client based computer application and test the same actions for different users, different data set, on various Windows operating systems and/or different browsers. Automation using UFT if planned and executed in a proper manner can save considerable time and money.

With the release of version 11.50, QTP and Service Test became part of UFT 11.50 software. UFT is one of the most widely used automation testing tools in the market today with over 60% market share. Due to this reason, skilled UFT professionals are always in demand.'

### **T8: Tools zum automatisierten Testen: TestComplete**

Die Ausarbeitung soll kurz Data Driven Testing erläutern und sich dann intensiv mit dem Tool TestComplete beschäftigen. Es gibt die Möglichkeit, das Tool auszuprobieren, was unbedingt gemacht werden sollte.

#### **Literatur:**

[http://greiterweb.de/spw/test\\_werkzeuge.htm](http://greiterweb.de/spw/test_werkzeuge.htm)

<https://smartbear.com/product/testcomplete/overview/>

<https://smartbear.com/product/testcomplete/features/>

#### **Beschreibung:**

Das Tool wird auf dieser Internetseite folgendermaßen vorgestellt:

'TestComplete has an open and flexible architecture that makes creating, maintaining, and executing automated tests across desktop, web, and mobile applications easy, speedy, and cost effective. Some of the key features of the TestComplete Platform that demonstrate its power include:

- for multiple scripting languages
- The ability to record robust automated tests without scripting knowledge
- Regression tests that don't fail when the UI changes
- Data-driven testing
- Custom plugins and extensions'

**T9: Automatisiertes Testen: LLVM-Compiler-Framework und das Tool Klee (KLEE LLVM Execution Engine) im Speziellen**

Für dieses Thema sollten Sie Kenntnisse in C und C++ haben und Interesse, sich in das für diese Sprachen sehr wichtiges Framework LLVM einzuarbeiten und sich eine spezielle Anwendung (KLEE) genauer anzusehen und für das Seminar aufzubereiten.

**Literatur:**

<https://klee.github.io/>

<http://klee.github.io/publications/> mit z.B. den PDFs

- KLEE: Unassisted and Automatic Generation of High-Coverage Tests for Complex Systems Programs  
Cristian Cadar, Daniel Dunbar, Dawson Engler
- Server-side Verification of Client Behavior in Online Games  
Darrell Bethea, Robert Cochran, Michael Reiter

**T10: Continuous Integration und Jenkins**

Werkzeuge aus einem Bereich der Software-Entwicklung, der sog. 'kontinuierlichen Integration' (Continuous Integration, CI), unterstützen den Prozess des Bauens und Testens einer Anwendung. Die die Software entwickeln, nutzen dazu ein gemeinsames Versionsmanagement, in das ihre jeweiligen Änderungen regelmäßig integriert werden. Die Änderungen sollten lauffähig sein, so dass die Anwendung automatisiert getestet und ggf. gebaut (Build, Nightly Builds) werden kann.

Ein Beispiel für ein Open Source Continuous Integration System ist das auf Java basierende Jenkins. Es bietet eine webbasierte Benutzeroberfläche und unterstützt zahlreiche Werkzeuge wie z. B. SVN, Ant, Maven und JUnit. Durch Erweiterung mit Plugins lässt sich Jenkins individuell an Projekte anpassen.

Ihr Seminarbeitrag beschreibt den Prozess der kontinuierlichen Integration und deren bekannteste Vertreter im Allgemeinen und das Werkzeug Jenkins im Besonderen. Außerdem stellen Sie den Ablauf des CI-Prozesses anhand mehrerer typischer Anwendungsbeispiele vor.

**Literatur:**

<http://coding24.de/index.php/2016/08/31/continuous-integration/>

<https://www.martinfowler.com/articles/continuousIntegration.html>

<https://jenkins.io/>

<http://home.edvsz.fh-osnabrueck.de/skleuker/CSI/Werkzeuge/Jenkins/>

<https://www.tutorialspoint.com/jenkins/>

**T11: Design by Contract in der Sprache Groovy: GContract**

Diese Ausarbeitung soll die interessanten Eigenschaften der Sprache Groovy vorstellen und insbesondere GContract beschreiben und ausprobieren. Eine persönliche Bewertung der Sprache und der Umsetzung des Design by Contract soll die Ausarbeitung abrunden.

**Literatur:**

<https://jaxenter.de/gcontracts-vertraglich-korrektes-programmieren-4452>

<https://github.com/andresteingress/gcontracts/wiki>:

'Welcome to GContracts - A Programming by Contract Extension for Groovy!

This project aims to provide contract annotations that support the specification of class-invariants, pre- and post-conditions on Groovy classes and interfaces. In addition, GContracts supports postconditions with the old and result variable and inheritance of all assertion types.'

**T12: Design by Contract in den Programmiersprachen Eiffel, Python und D - ein Überblick**

Für diese Ausarbeitung soll die passende Literatur gesucht und die Umsetzung von Design by Contract in diesen drei sehr verschiedenen Programmiersprachen dargestellt und verglichen werden.

**Literatur:**

<http://dlang.org/> für die Programmiersprache D

**T13: Mehrfachvererbung und Mixins**

Die meisten neueren Sprachen bieten keine Mehrfachvererbung an. Stattdessen arbeiten sie mit sogenannten Mixins. Beispiele dafür sind die Sprachen Scala, Dart und Ruby.

Was sind die Vorteile von Mixins, was die Herausforderungen?

In der Ausarbeitung sollten mindestens zwei Sprachen betrachtet werden, die Mixins verwenden.

**Literatur:**

<https://www.dartlang.org/>

<http://www.scala-lang.org/>

<https://www.ruby-lang.org/de/>

**T14: Domain Specific Languages: Spoofax**

In der Seminararbeit soll das Tool Spoofax vorgestellt werden, mit dessen Hilfe DSLs erstellt werden können. Hierfür wird erwartet, dass nicht nur eine hilfreiche Beschreibung des Tools erfolgt, sondern das Tool auch ausprobiert wird, eine eigene kleine DSL erstellt und die Arbeit mit dem Tool bewertet wird.

**Literatur:**

<http://www.voelter.de/dslbook/markusvoelter-dslengineering-1.0.pdf>

**T15: .Net Compiler mit Roslyn**

Dieses Thema ist interessant für Studierende, die sich tiefer in die inneren Geheimnisse von Programmiersprachen einarbeiten möchten. Wozu man so etwas wie Roslyn braucht, zeigt exemplarisch die unten aufgeführte Abschlussarbeit von Ricardo Niepel.

**Literatur:**

Scheibenweise: Microsoft öffnet .Net-Compiler mit Roslyn

<http://www.heise.de/ix/artikel/Scheibenweise-1475294.html>

Abschlussarbeit Ricardo Niepel, Constrained based diagnostics and code fixes

<http://www.fernuni-hagen.de/ps/arbeiten/>

Die Programmiersprache Ruby wird hier behandelt, weil sie die Voraussetzung zum Verständnis des Frameworks Ruby-on-Rails ist. Sie selbst ist kein Framework.

**T16: Ruby - Grundlagen**

Die Ausarbeitung soll die Sprache Ruby vorstellen. Dabei soll auf die drei verschiedenen Arten von Vererbung, das Duck Typing und die Möglichkeiten, es abzusichern (Exceptions, Introspection etc.) und funktionale Programmierung eingegangen werden. Die Sprache soll ausprobiert werden, die Beispiele - in Anlehnung an die Literatur - selbst erstellt werden. Hier sollen insbesondere die Themenbereiche

- Vererbung,
- Ducktyping und die Möglichkeiten, es abzusichern (Exceptions, Introspection etc.),
- aspektorientierte Programmierung mit z.B. AspectR
- kontextorientierte Programmierung durch ContextR

behandelt werden.

**Literatur:**

<https://www.ruby-lang.org/de/>

<https://www.ruby-lang.org/de/documentation/>

<https://www.ruby-lang.org/en/news/2017/03/22/ruby-2-4-1-released/>

<https://entwickler.de/online/development/metaprogrammierung-mit-ruby-125311.html>

<http://aspectr.sourceforge.net/>

Masterarbeit zu kontextorientierter Programmierung:

[https://github.com/phenomenal/rphenomenal/blob/master/public/The\\_Phenomenal\\_Gem\\_Poncelet\\_Vigneron\\_2012.pdf](https://github.com/phenomenal/rphenomenal/blob/master/public/The_Phenomenal_Gem_Poncelet_Vigneron_2012.pdf)

**T17: Ruby on Rails**

Die Ausarbeitung soll eine Einführung in Ruby on Rails liefern. Dabei soll sie auf Grundprinzipien wie

- Don't repeat yourself
- Convention over Configuration
- Scaffolding

eingehen, das Schichtenmodell mit Model, Controller, View betrachten und besondere Eigenschaften oder Features der verschiedenen Versionen darstellen. Interessant ist insbesondere in



Version 5.0 das Test-Framework Test Runner.

**Literatur:**

Ruby und Ruby on Rails: Webanwendungen, Version 5.0 mit eigenem Test-Framework Test Runner, s.

<http://rubyonrails.org/>, <https://rubyonrails.de/>

<https://www.railstutorial.org/book>

[http://guides.rubyonrails.org/5\\_0\\_release\\_notes.html](http://guides.rubyonrails.org/5_0_release_notes.html)

**T18: Scala und Webanwendungen mit dem Play!-Framework**

Die Ausarbeitung soll die Sprachelemente von Scala vorstellen, die für Webanwendungen wichtig sind. Des Weiteren soll eine kurze Beschreibung des Play!-Frameworks verdeutlichen, was dieses Framework leistet und wofür man dieses Framework verwenden sollte.

**Literatur:**

<https://www.innoq.com/de/articles/2012/08/webanwendungen-mit-play/>

<https://www.playframework.com/>

<https://www.playframework.com/documentation/2.6.x/Home>

**Bücher:**

1. **Mastering Play Framework for Scala** by Shiti Saxena , Publisher: Packt Publishing, Release Date: May 2015, Pages: 274

<http://shop.oreilly.com/product/9781783983803.do>

'Play Framework is an open source web application framework that is written in Java and Scala. It follows the Model-View-Controller architectural pattern and enables the user to employ Scala for application development, while keeping key properties and features of Play Framework intact. Starting off by building a basic application with minimal features, you get a detailed insight into handling data transactions and designing models in Play. Next, you venture into the concepts of Actors and WebSockets, the process of manipulating data streams, and testing and debugging an application in Play. Finally, you gain an insight into extending the framework by writing custom modules or plugins in Play. Each chapter has a troubleshooting section that helps you out by discussing the causes of, and solutions to, some commonly faced issues.'

2. <https://www.manning.com/books/play-for-scala>

'Play is a Scala web framework with built-in advantages:

Scala's strong type system helps deliver bug-free code, and the Akka framework helps achieve hassle-free concurrency and peak performance. Play builds on the web's stateless nature for excellent scalability, and because it is event-based and nonblocking, you'll find it to be great for near real-time applications. Play for Scala teaches you to build Scala-based web applications using Play.

It gets you going with a comprehensive overview example. It then explores each facet of a typical Play application by walking through sample code snippets and adding features to a running example. Along the way, you'll deepen your knowledge of Scala and learn to work with tools like Akka. Written for readers familiar with Scala and web-based application architectures. No knowledge of Play is assumed.

### **T19: Unterschiede zwischen den Php Frameworks von Zend und Symfony**

Im Bereich der Programmierung von Internetanwendungen mit Php führt kaum ein Weg am Zend oder Symfony Web Application Framework vorbei. Beide verstehen sich dabei weniger als Framework im klassischen Sinne, sondern mehr als Ansammlung von Komponenten, die bedarfsgemäß ausgewählt und verwendet werden können. Eine der schwierigsten Aufgaben der Webentwicklung ist das Erstellen von anspruchsvollen Anwendungen in möglichst kurzer Zeit. Beide Frameworks versprechen durch ein großes Angebot verschiedenster Werkzeuge, dieser Aufgabe gerecht zu werden.

Ihr Seminarbeitrag stellt beide Frameworks vor und wägt deren Vor- und Nachteile ab. Sie gehen insbesondere der Frage nach, ob und unter welchen Bedingungen beide Frameworks ihr Versprechen hoher Produktivität tatsächlich entfalten können.

#### **Literatur:**

<https://framework.zend.com/>

<https://symfony.com/>

### **T20: Eigenes Thema**

Sie dürfen auch ein eigenes Thema vorschlagen, was zu diesem Seminar passt. Bei dem Thema bitte eine Beschreibung angeben und Literaturhinweise.