



Hochschule
Kaiserslautern
University of
Applied Sciences



Hochschule Kaiserslautern
- FACHBEREICH INFORMATIK UND MICROSYSTEMTECHNIK -

Der Titel der Arbeit

Abschlussarbeit zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

vorgelegt von

Vorname Nachname

12345

Betreuer Hochschule: Prof. Dr.

Betreuer PENTASYS: B. Sc.

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Dies ist ein Zitat.

verstanden, scheinen nun doch vorueber zu sein. Dies ist der Text sein.
siehe: <http://janeden.net/die-praeambel>

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vi
1 Einführung	1
2 Grundlagen	2
3 Was ist Reactive Programming	3
3.1 Reactive Programming - Ein neues Programmierparadigma?	3
3.1.1 Was versteht man unter Programmierparadigmen	3
3.1.2 Vergleich: Reactive Programming und Objekt orientierte Programmierung	3
3.2 Für was steht das Reactive im Kontext der Programmierung	3
3.2.1 Differenzierung zwischen Reactive Proramming und Reactive Systems . .	4
3.3 Reactive Programming vs. Functional Reactive Programming	4
3.4 Überblick über bekannte Frameworks und ihre Eigenschaften	4
3.4.1 Reactivex.io	4
3.4.2 Allgemeine Übersicht	4
3.4.3 Übersicht spezielle für die Entwicklung mit Java	4
3.5 Testen von reaktivem Code mit dem JUnit Framework	4
4 Einführung in Reactive Programming mit RxJava	5
4.1 Wie funktioniert Reactive Programming	5
4.1.1 Synchronität	5
4.1.2 Parallelisierung	5
4.1.3 Push vs. Pull	5
4.2 Rx.Observable	5
4.3 Rx.Observer	5
4.3.1 Rx.Subscriber	6
4.4 Operationen und Transformationen	6
4.4.1 Exkursion: Streams API Java 8	6
4.4.2 Operation filter()	6

4.4.3	Transformation map()	6
4.4.4	Transformation flatMap()	6
4.4.5	Operation merge()	6
4.4.6	Operation zip()	6
5	Beispiel: Implementierung eines Systemmonitors	7
5.1	API für Systemwerte	7
5.1.1	Beschreibung Klasse 1	7
5.1.2	Beschreibung Klasse 2	7
5.1.3	Beschreibung Klasse 3	7
5.1.4	Beschreibung Klasse 4	7
5.2	Client für API: GUI zur Repräsentation der Systemwerte	7
5.2.1	Beschreibung Klasse 1	7
5.2.2	Beschreibung Klasse 2	7
6	Fazit	8
6.1	Unterpunkt	8
7	Beispiele	9
7.1	Schriftarten	9
7.1.1	Symbole	9
7.2	Abbildungen	9
7.3	Tabellen	10
7.4	Verweise	10
7.4.1	Pageref und Ref	10
7.5	Listing	11
	Literaturverzeichnis	I

Abbildungsverzeichnis

7.1	Beispiel Bild; Quelle ist png	10
-----	---	----

Tabellenverzeichnis

7.1	Beispiel Beschriftung einer Tabelle	10
-----	---	----

Listingverzeichnis

7.1	Das Listing zeigt Java Quellcode	11
-----	--	----

Kapitel 1

Einführung

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Kapitel 2

Grundlagen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Kapitel 3

Was ist Reactive Programming

Historische Einführung. Wann und von wem wurde das erste Mal von RP gesprochen, wie war die weitere Entwicklung? Kapitelbeschreibung: Was folgt.

3.1 Reactive Programming - Ein neues Programmierparadigma?

Wie gliedert man einen Programmierstil ein? ¹

3.1.1 Was versteht man unter Programmierparadigmen

Erklärung was sind Paradigmen und wieso werden sie definiert.

3.1.2 Vergleich: Reactive Programming und Objekt orientierte Programmierung

Eventuell bessere mit Funktionaler Programmierung zu vergleichen. Muss noch genauer betrachtet werden. Soll die Unterschiede zu den gängigen, bekannten Methoden aufzeigen.

3.2 Für was steht das Reactive im Kontext der Programmierung

Was bedeutet der Begriff Reactive im eigentlichen Sinne, wie genau findet man dass in der Programmierung wieder?

¹[BCv⁺13]

3.2.1 Differenzierung zwischen Reactive Programming und Reactive Systems

Das Manifest bezieht sich eigentlich auf die Systeme. Wo kommt nun RP in Spiel?

3.3 Reactive Programming vs. Functional Reactive Programming

FRP findet nun mal in Funktionalen Programmiersprachen statt. Java ist jedoch OO und mit Java 8 und dem Rx Frameworks werden funktionale Eigenschaften in der OO Sprache eingebracht. Unterschiede müssen noch genau belegt² werden.

3.4 Überblick über bekannte Frameworks und ihre Eigenschaften

Überblick quer über die gängigen Programmiersprachen.

3.4.1 Reactivex.io

Kurze Erläuterung zu der Entstehung von Reactive Extensions

3.4.2 Allgemeine Übersicht

Rx Frameworks zu den jeweiligen Sprachen. Frameworks wie z.B. Akka³.

3.4.3 Übersicht speziell für die Entwicklung mit Java

RxJava. Reactive Streams Konvention. Java 9 Api Änderung bzgl. Reactive Streams.

Framework für JavaFX - RxJavaFX

Einführung und Eigenschaften erläutern

3.5 Testen von reaktivem Code mit dem JUnit Framework

Noch nichts genaues. Muss noch geschaut werden wie die Funktionalität von JUnit RP abdeckt.

²[Jan16]

³[D16]

Kapitel 4

Einführung in Reactive Programming mit RxJava

Beschreibung wieso RxJava. Beschreibung was beschrieben wird.

4.1 Wie funktioniert Reactive Programming

Einleitung zum Aufbau: Klassenübersicht des Frameworks mit Erklärung.

4.1.1 Synchronität

Sync vs. Async - was bringt RP in dieser Hinsicht

4.1.2 Parallelisierung

Concurrency vs. Parallelism - was tritt wie wann auf bzw. kann wie wann angewandt werden

4.1.3 Push vs. Pull

Wichtigster Unterschied. Observable als Gegenpart zu Iterable - somit Push vs. Pull Vergleich.

4.2 Rx.Observable

Interface Übersicht. Nutzen und Anwendung anhand von Beispiel. Hot vs. Cold

4.3 Rx.Observer

Was kann Observer -> Interface Übersicht

4.3.1 Rx.Subscriber

Was ist speziell am Subscriber -> Interface Übersicht

4.4 Operationen und Transformationen

Erläuterung von den Stadien der Operation von Beginn über Mitte bis Ende.

4.4.1 Exkursion: Streams API Java 8

Beschreibung was Streams darstellen, wie sich Observables im Vergleich verhalten

4.4.2 Operation filter()

Beispiel und Perlenbild. Einsatz beschreiben

4.4.3 Transformation map()

Beispiel und Perlenbild. Einsatz beschreiben

4.4.4 Transformation flatMap()

Beispiel und Perlenbild. Einsatz beschreiben

4.4.5 Operation merge()

Beispiel und Perlenbild. Einsatz beschreiben

4.4.6 Operation zip()

Beispiel und Perlenbild. Einsatz beschreiben Eventuell noch mehr Operationen

Kapitel 5

Beispiel: Implementierung eines Systemmonitors

Beschreibung der Funktionen der Anwendung. Überblick über Projekt/Klassenstruktur. Verwendete Tools und Versionen.

5.1 API für Systemwerte

Framework für die Systemwerte kurz erläutern. Grobes Vorgehen beschreiben wie man es in etwa Umsetzen kann.

5.1.1 Beschreibung Klasse 1

5.1.2 Beschreibung Klasse 2

5.1.3 Beschreibung Klasse 3

5.1.4 Beschreibung Klasse 4

5.2 Client für API: GUI zur Repräsentation der Systemwerte

5.2.1 Beschreibung Klasse 1

5.2.2 Beschreibung Klasse 2

Kapitel 6

Fazit

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est

6.1 Unterpunkt

Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Kapitel 7

Beispiele

Dieses Kapitel soll viele alltaegliche Beispiele¹ abdecken um einen L^AT_EX Dokument zu setzen

7.1 Schriftarten

- Kursiv: *Das ist ein Beispiel*
- Unterstreichen: Das ist ein Beispiel
- Fettschrift: **Das ist ein Beispiel**
 - Kombination aus dreien: ***Das ist ein Beispiel***
- Serifen: Das ist ein Beispiel
- Schreibmaschinen Schrift: Das ist ein Beispiel
- Kleine Grossbuchstassen: DAS IST EIN BEISPIEL
- Ausfuehrungszeichen: “Das ist ein Beispiel”
- asld

7.1.1 Symbole

1. Deutsche Umlaute: Ä, Ö, Ü, ä, ö, ü, ß
2. Sammlung von Sonderzeichen http://de.wikibooks.org/wiki/LaTeX-Kompendium:_Sonderzeich

7.2 Abbildungen

Wie folgt bindet man Abbildungen ein:

¹Fussnote

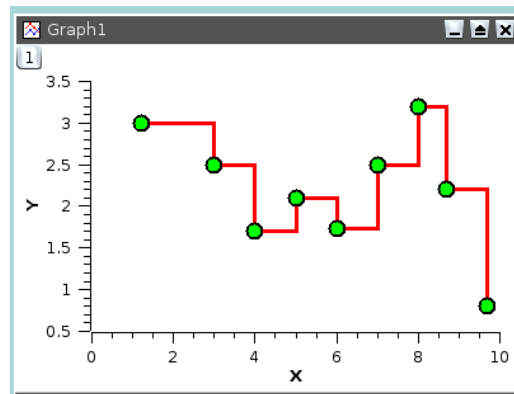


Abbildung 7.1: Beispiel Bild; Quelle ist png

7.3 Tabellen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren².

Stadium	Substratfreie Kontrolle	Probenansatz	
	Farbe	Farbe	Bewertung
Alpha1	farblos	braun	+++
Beta2	farblos	farblos	-

2. Beispiel

RANG	NAME	RATING
1	Garry Kasparov	2817
2	Viswanathan Anand	2774
3	Wladimir Kramnik	2764

Tabelle 7.1: Beispiel Beschriftung einer Tabelle

7.4 Verweise

Hier werden Verweise auf verschiedene Elemente erstellt [?]

7.4.1 Pageref und Ref

Diese Textstelle ist sehr interessant.

Hier wird auf die Textstelle 7.4.1 verwiesen,

²[Dan14]

die sich auf der Seite 10 befindet.

Verweis auf Listing 7.1 auf Seite 11

Verweis auf Abbildung 7.1 auf Seite 10

Verweis auf Tabelle 7.1 auf Seite 10

7.5 Listing

```
1  /* Java Hallo World Beispiel */  
  
   public class HelloWorld {  
       public static void main(String [] args) {  
5         System.out.println("Hello, World");  
       }  
   }
```

Listing 7.1: Das Listing zeigt Java Quellcode

Literaturverzeichnis

- [BCv⁺13] Engineer Bainomugisha, Andoni Lombide Carreton, Tom van Cutsem, Stijn Mostinckx, and Wolfgang de Meuter. A survey on reactive programming. *ACM Computing Surveys*, 45(4):1–34, 2013. doi:10.1145/2501654.2501666.
- [D16] Dávid Karnok. Operator-fusion (part 1), 2016. URL: <https://akarnokd.blogspot.de/2016/03/operator-fusion-part-1.html>.
- [Dan14] Dan Lew. Grokking rxjava, part 2: Operator, operator, 2014. URL: <http://blog.danlew.net/2014/09/22/grokking-rxjava-part-2>.
- [Jan16] Jan Carsten Lohmüller. Reactive programming – mehr als nur streams und lambdas, 2016. URL: <https://www.informatik-aktuell.de/entwicklung/programmiersprachen/reactive-programming-mehr-als-nur-streams-und-lambdas.html>.