Beispiel fi¿½r eine Abschlussarbeit

Manfred Brill

10. Mi¿½rz 2010

Inhaltsverzeichnis

1 Einleitung

Ein Beispiel für die Hauptdatei einer Diplomarbeit und die Aufteilung der einzelnen Kapitel in einzelne Dateien, die mit `\input` in die Hauptdatei `beispiel.tex` integriert werden.

Als Literaturdatenbank wird die Datei `arbeit.bib` verwendet. Die Abbildung ist sowohl im PNG- als auch im EPS-Format vorhanden. Das Beispiel ist also sowohl mit `latex` → `dvips` → `ps2pdf` als auch direkt mit `pdflatex` kompilierbar.

2 Was ist Reactive Programming

Historische EinfÃ¼hrung. Wann und von wem wurde das erste Mal von RP gesprochen, wie war die weitere Entwicklung? Kapitelbeschreibung: Was folgt.

2.1 Reactive Programming - Ein neues Programmierparadigma?

Wie gliedert man einen Programmierstil ein?

2.1.1 Was versteht man unter Programmierparadigmen

ErklÃ¤rung was sind Paradigmen und wieso werden sie definiert.

2.1.2 Vergleich: Reactive Programming und Objekt orientierte Programmierung

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

2.2 FÃ¼r 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?

2.2.1 Differenzierung zwischen Reactive Proramming und Reactive Systems

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

2.3 Reactive Programming vs. Functional Reactive Programming

FRP findet nun mal in Funktionalen Programmiersprachen statt. Java is jedoch OO und mit Java 8 und dem Rx Frameworks werden funktionale Eigenschaften in der OO Sprache eingebracht. Unterschiede mÃ¼ssen noch genau belegt werden.

2.4 Ãberblick Ã¼ber bekannte Frameworks und ihre Eigenschaften

Ãberblick quer Ã¼ber die gÃ¤ngigen Programmiersprachen.

2.4.1 Reactivex.io

Kurze ErlÃ¤uterung zu der Entstehung von Reactive Extensions

2.4.2 Allgemeine Ãbersicht

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

2.4.3 Ãbersicht spezielle fÃ¼r die Entwicklung mit Java

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

Framework fÃ¼r JavaFX - RxJavaFX

EinfÃ¼hrung und Eigenschaften erlÃ¤utern

2.5 Testen von reaktivem Code mit dem JUnit Framework

Noch nichts genaues. Muss noch geschaut werden wie die FunktionalitÃ¤t von JUnit RP abdeckt.

3 EinfÃ¼hrung in Reactive Programming mit RxJava

Beschreibung wieso RxJava. Beschreibung was beschrieben wird.

3.1 Wie funktioniert Reactive Programming

Einleitung zum Aufbau: KlassenÃ¼bersicht des Frameworks mit ErklÃ¤rung.

3.1.1 SyncronitÃ¤t

Sync vs. Async - was bringt RP in dieser Hinsicht

3.1.2 Parallelisierung

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

3.1.3 Push vs. Pull

Wichtigster Unteschied. Observable als Gegenpart zu Interable - somit Push vs. Pull Vergleich.

3.2 Rx.Observable

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

3.3 Rx.Observer

Was kann Observer -¿ Interface Ãbersicht

3.3.1 Rx.Subscriber

Was ist speziell am Subscriber -¿ Interface Ãbersicht

3.4 Operationen und Transformationen

ErlÃ¤uterung von den Stadien der Operation von Beginn Ã¼ber Mitte bis Ende.

3.4.1 Exkursion: Streams API Java 8

Beschreibung was Streams darstellen, wie sich Observables im Vergleich verhalten

3.4.2 Operation filter()

Beispiel und Perlenbild. Einsatz beschreiben

3.4.3 Transformation map()

Beispiel und Perlenbild. Einsatz beschreiben

3.4.4 Transformation flatMap()

Beispiel und Perlenbild. Einsatz beschreiben

3.4.5 Operation merge()

Beispiel und Perlenbild. Einsatz beschreiben

3.4.6 Operation zip()

Beispiel und Perlenbild. Einsatz beschreiben Eventuell noch mehr Operationen

4 Beispiel: Implementierung eines Systemmonitors

Beschreibung der Funktionen der Anwendung. Ãberblick Ã¼ber Projekt/Klassenstruktur. Verwendete Tools und Versionen.

4.1 API fÃ¼r Systemwerte

Framework fÃ¼r die Systemwerte kurz erlÃ¤utern. Grobes Vorgehen beschreiben wie man es in etwa Umsetzen kann.

4.1.1 Beschreibung Klasse 1

4.1.2 Beschreibung Klasse 2

4.1.3 Beschreibung Klasse 3

4.1.4 Beschreibung Klasse 4

4.2 Client fÃ¼r API: GUI zur ReprÃ¤sentation der Systemwerte

4.2.1 Beschreibung Klasse 1

4.2.2 Beschreibung Klasse 2

Literaturverzeichnis