

Ulf Fildebrandt

Software modular bauen

**Architektur von langlebigen Softwaresystemen –
Grundlagen und Anwendung mit OSGi und Java**



dpunkt.verlag

Ulf Fildebrandt
E-Mail: ulffildebrandt@aol.com

Lektorat: Dr. Michael Barabas
Copy Editing: Ursula Zimpfer, Herrenberg
Herstellung: Nadine Thiele
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-86490-019-8

1. Auflage 2012
Copyright © 2012 dpunkt.verlag GmbH
Ringstraße 19B
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Für meinen Vater

Inhaltsverzeichnis

1	Einleitung	1
1.1	Über dieses Buch	2
1.2	Motivation	4
1.3	Nachstellen der Beispiele	6
1.4	Regeln	8
2	Grundlagen	11
2.1	Definitionen	11
2.2	Grundthese	12
2.3	Prinzipien und Konzepte	16
2.4	Modularity Patterns	21
2.5	Vorgehen zur Definition modularer Architektur	25
2.6	Modularitätsmodell (Modularity Maturity Model)	29
3	Coding-Architektur: Erweiterbarkeit	33
3.1	Hintergrund von Coding Best Practices	33
3.2	Beschreibung des Anwendungsbeispiels	36
3.3	Factory Pattern	40
3.4	Decision Map	45
3.5	Single Responsibility	49
3.6	Nachstellen des Beispiels	51
3.7	Exkurs: Adapter und Facade	51
3.8	Zusammenfassung	57

4	Komponentenarchitektur: Grundlagen	59
4.1	Hintergrund eines modularen Laufzeitsystems	59
4.2	Umsetzung der Grundkonzepte am Beispiel OSGi	71
4.3	Auswirkungen eines modularen Laufzeitsystems auf Patterns	80
4.4	Umsetzung von Modularität in verschiedenen Laufzeitumgebungen	85
4.5	Modularisierung durch Refactoring	89
4.6	Zusammenfassung	96
5	Komponentenarchitektur: Entkopplung	99
5.1	Hintergrund modularisierter Software	99
5.2	Grundlagen von Declarative Services	104
5.3	Entkopplung in der Beispielanwendung	106
5.4	Exkurs: Zwei weitere SOLID-Prinzipien	115
5.5	Hierarchien in der Beispielanwendung	119
5.6	Schnittstellen in Modulen	121
5.7	Zusammenfassung	122
6	Systemarchitektur: Schichten	125
6.1	Hintergrund von Schichten in der Architektur	125
6.2	Entkoppelte Komponenten in der Beispielanwendung	128
6.3	Gruppierung von Bundles	134
6.4	Schichtenarchitektur mit OSGi	137
6.5	Exkurs: Testbarkeit	148
6.6	Zusammenfassung	152
7	Systemarchitektur: Erweiterbarkeit	153
7.1	Hintergrund von Schnittstellen	153
7.2	Schnittstellen in der Beispielanwendung	160
7.3	Erweiterung der Schnittstellen	166
7.4	Erweiterbarkeit mit OSGi	172
7.5	Ersetzbarkeit von Implementierungen	174
7.6	Kohäsion in der Beispielanwendung	176
7.7	Zusammenfassung	178

8	Systemarchitektur: Wiederverwendbarkeit entkoppelter Komponenten	181
8.1	Hintergrund entkoppelter wiederverwendbarer Komponenten	181
8.2	Entkopplung von externen Abhängigkeiten	186
8.3	Gemeinsam genutzte Aspekte	193
8.4	Zusammenfassung	197
9	Systemarchitektur: Funktionale Entkopplung	199
9.1	Hintergrund funktionaler Entkopplung	199
9.2	Parallele Verarbeitung in der Beispielanwendung	208
9.3	Versionierung in der Beispielanwendung	219
9.4	Zusammenfassung	232
10	Komponentenarchitektur: Frameworks	233
10.1	Hintergrund der Frameworks	234
10.2	Umsetzung der Beispielanwendung mit CDI	247
10.3	Umsetzung der Beispielanwendung mit Spring	253
10.4	Umsetzung der Beispielanwendung mit Maven	257
10.5	Nachstellen des Beispiels	258
10.6	Exkurs: Verwendung von OSGi	259
10.7	Zusammenfassung	260
11	Systemarchitektur: Product Line Engineering	263
11.1	Hintergrund zu Product Line Engineering	263
11.2	Anwendung von PLE auf die Beispielanwendung	266
11.3	Mehrere Produkte	273
11.4	Nachstellen des Beispiels	276
11.5	Zusammenfassung	277
12	Systemarchitektur: Repository	279
12.1	Hintergrund von Repositories	279
12.2	Ein Repository für OSGi	281
12.3	Das p2-Repository für die Beispielanwendung	284
12.4	Verschiedene Einsatzmöglichkeiten	288
12.5	Maven-Repository	291
12.6	Zusammenfassung	292

13	Schlusswort	295
13.1	Zusammenfassung	295
13.2	Ausblick	299
A	Appendix A: Tools zur Architekturvalidierung	301
A.1	Korrektheit	301
A.2	Metriken	302
A.3	Redundanz	303
B	Appendix B: Projekt Jigsaw	305
B.1	Designprinzipien	305
B.2	Definitionen	306
B.3	Moduldeklaration	306
B.4	Vergleich mit OSGi	309
C	Appendix C: OSGi in der Praxis	311
C.1	Lifecycle von Bundles	311
C.2	Statusinformationen eines OSGi-Laufzeitsystems	313
	Referenzen	315
	Index	319