

Eduard Hildebrandt

Aspektorientierte Programmierung mit Java
gezeigt am Beispiel einer konkreten
Anwendung

Diplomarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



FACHHOCHSCHULE LANDSHUT
Fachbereich Informatik

Aspektororientierte Programmierung mit Java
gezeigt am Beispiel einer konkreten Anwendung

Diplomarbeit

zur Erlangung des Grades eines
Diplom Informatikers (FH)

erstellt von
Eduard Hildebrandt

Eingereicht: 10. Januar 2005

Widmung

*Diese Diplomarbeit ist meinen Eltern gewidmet,
die immer für mich da sind.*

I. Eidesstattliche Erklärung

(gemäß § 31, Abs. 7 RaPO)

Hiermit erkläre ich, dass ich die Arbeit selbstständig verfasst, noch nicht anderweitig für Prüfungszwecke vorgelegt, keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt sowie wörtliche oder sinngemäße Zitate als solche gekennzeichnet habe.

Landshut, den 10.01.2005

(Eduard Hildebrandt)

II. Kurzfassung

Diese Diplomarbeit untersucht die Praxistauglichkeit der aspektorientierten Programmierung (AOP) mit Java am Beispiel einer konkreten J2EE Anwendung. Nach einer kurzen Einführung in die Grundzüge der aspektorientierten Programmierung, werden die verschiedenen Einsatzgebiete, wie Logging, Pooling, Qualitätssicherung oder Testen von Software vorgestellt. Dabei wird die AOP Lösung immer mit einer herkömmlichen Java Implementierung verglichen und die Vor- und Nachteile der beiden Implementierungen untersucht. Als Ergebnis dieser Arbeit soll geprüft werden, ob AOP inzwischen bereit für den Praxiseinsatz ist und der Nutzen sowie Probleme dieser Technologie untersucht werden.

III. Danksagung

Bei den meisten Diplomarbeiten steht auf der Titelseite in der Regel nur ein einziger Name als Autor der Arbeit. Dies ist eine arge Täuschung! Jede Diplomarbeit entsteht aus einer Kollaboration hingebungsvoller und talentierter Menschen. Beim Schreiben dieser Arbeit haben mir eine Vielzahl großzügiger und kluger Leute mit Rat und Tat zur Seite gestanden. Besonderen tiefen Dank empfinde ich für:

Meine Eltern, denen diese Diplomarbeit gewidmet ist. Sie haben mir das Studium der Informatik überhaupt erst ermöglicht. Sie haben stets zu mir gehalten und mich immer unterstützt. Ich danke für ihre Liebe und Vertrauen. Sie sind die beiden Riesen, auf deren Schultern ich gestanden bin.

Frau Prof. Dr. Gudrun Schiedermeier, die mir meine erste Java Lektion während des Informatikstudiums gegeben hatte. Sie begleitete diese Diplomarbeit als Erstkorrektorin und gab mir die Freiheit ohne Zwänge und Auflagen eine neue Technologie zu erforschen. Stets war sie erreichbar und hatte immer gute Vorschläge, wie ich meine Arbeit verbessern kann. Dafür möchte mich herzlich bedanken.

Frau Prof. Dr. Monika Messerer, die mir die Themen Softwareengineering und Design Patterns im Laufe des Studiums beigebracht hatte. Sie hat sich ebenfalls für diese Thematik als Zweitkorrektorin begeistern können.

Der Firma Kölsch & Altmann, wo ich die Möglichkeit hatte, meine Arbeit in einer sehr guten Arbeitsumgebung zu gestalten. Selten habe ich ein Unternehmen gesehen, in dem ein solch familiäres Arbeitsklima herrscht. Mein besonderer Dank gilt hier dem Geschäftsführer Dr. Werner Altmann, der mir diese Gelegenheit überhaupt erst ermöglichte und meine Arbeit stets mit großem Interesse verfolgte. Ein großer Dank gilt auch meinen beiden Betreuen Georg Süß und Klaus Dirlwanger, die mir immer mit Rat und Tat zur Seite gestanden haben.

Meinem besten Freund Matthias Schaufler, der immer für mich da ist. Er hat viel Zeit investiert diese Arbeit auf Rechtschreib- und Grammatikfehler zu überprüfen. Für seine Mühen möchte ich mich besonders bedanken, da er sich eigentlich nicht für Informatik interessiert und daher auch relativ wenig vom Inhalt verstanden hat. Umso mehr konnte er durch seine Vorschläge die Ausdrucksweise verbessern.

Meine Studienkollegien Waldemar Mertke und Philipp Morgenthaler, die sich die Zeit genommen haben, meine Diplomarbeit durchzulesen und viel konstruktive Kritik gegeben haben.

Allen anderen Leuten, die ich hier nicht explizit erwähnt habe, die jedoch in vielen verschiedenen Dialogen durch Ihre Ansichten und Vorschläge ebenfalls zum Erfolg dieser Arbeit beigetragen haben.

IV. Inhaltsverzeichnis

I. Eidesstattliche Erklärung	3
II. Kurzfassung	4
III. Danksagung	5
IV. Inhaltsverzeichnis	6
VI. Abbildungsverzeichnis	8
Kapitel 1: Einführung	9
1.1 Einleitung.....	9
1.2 Ziel der Arbeit	10
1.3 Voraussetzungen	10
1.4 Motivation.....	10
1.5 Aspektororientierte Programmierung	12
1.6 Weben von Aspekten.....	14
1.6.1 statisches Weben.....	14
1.6.7 dynamisches Weben	14
1.7. AspectJ	14
1.7.1 Schnittpunkte (engl. pointcuts)	15
1.7.2 Advices.....	16
1.7.3 Aspekte	16
1.7.4 Introductions	17
1.7.5 Compiler Regeln	17
1.8. „Hello World“ mit AspectJ.....	18
1.9. Zusammenfassung	19
Kapitel 2: Die Anwenung - iPad	21
2.1 Anforderungen	21
2.2 Design.....	23
2.2.1 Klassen Design	23
2.2.2 Datenbank Design.....	23
2.2.3 Grafische Oberfläche	24
2.2.4 Architektur.....	25
2.3 Implementierung	27
2.3.1 Allgemeines	27
2.3.2 Technologien.....	27
2.3.2.1 Hibernate.....	27
2.3.2.2 Axis	28
2.3.2.3 Lucene	28
2.3.2.4 Eclipse RCP.....	29
2.3.2.5 JDIC	31
2.3.3 Probleme und Erfahrungen	31
2.3.3.1 SWT.....	31
2.3.3.2 Performance	32
2.4 Bewertung des Designs und Architektur.....	32
2.5 Zusammenfassung	33
Kapitel 3: Tracing und Logging	34
3.1 Konventionelles Logging	34
3.2 Logging mit AspectJ	36
3.2.1 Protokollieren von Methodenaufrufen	36
3.2.2 Protokollieren von Feldzugriffen.....	39
3.2.3 Protokollieren von Exceptions	40
3.3 Spezielle Logging Anwendungen.....	42
3.3.1 Protokollieren von SQL Statements	42
3.3.2 Protokollieren von Sessions	42
3.3.3 Protokollieren von Webservice Requests	43
3.4 Einsatz in der Beispielanwendung.....	44
3.5 Zusammenfassung	44

Kapitel 4: Qualitätssicherung	45
4.1 Best-Practices Richtlinien.....	45
4.2 EJB Richtlinien.....	46
4.3 Swing Richtlinien.....	47
4.4 Architektur Richtlinien.....	48
4.5 Benutzerdefinierte Richtlinien.....	49
4.6 Richtlinien für Namenskonventionen.....	50
4.7 Weitere Tools.....	51
4.8 Einsatz in der Beispielanwendung.....	51
4.9 Zusammenfassung.....	52
Kapitel 5: Pooling und Caching	53
5.1 Thread Pool.....	54
5.1.1 Java Implementierung.....	54
5.1.2 AspectJ Implementierung.....	56
5.1.3 Bewertung.....	57
5.2 XSLT Cache.....	58
5.2.1 Java Implementierung.....	58
5.2.2 AspectJ Implementierung.....	60
5.2.3 Bewertung.....	60
5.3 Einsatz in der Beispielanwendung.....	61
Kapitel 6: Design Patterns	62
6.1 Singleton Pattern.....	62
6.1.1 Java Implementierung.....	62
6.1.2 AspectJ Implementierung.....	62
6.1.3 Bewertung.....	64
6.2 Observer Pattern.....	64
6.2.1 Java Implementierung.....	65
6.2.2 AspectJ Implementierung.....	65
6.2.3 Bewertung.....	67
6.3 Zusammenfassung.....	67
Kapitel 7: Testen mit AspectJ	68
7.1 Prüfungen privater Variablen.....	68
7.2 Austausch von Methoden und Objekten.....	69
7.3 Virtuelle Mock Objekte.....	70
7.4 Unit Test Coverage.....	71
Kapitel 8: Fazit	73
8.1 Bewertung der Technologie.....	73
8.3 Ausblick in die Zukunft.....	74
8.4 Zusammenfassung.....	75
VII. Literaturverzeichnis	76
VIII. Abkürzungsverzeichnis	78
IX. Glossar	79
X. Anhänge	84
Anhang A: Eingesetzte Hilfsmittel.....	84
Anhang B: Inhalt der CD-ROM.....	85
Anhang D: Java AOP Implementierungen.....	87
Anhang E: AspectJ Plugins für Entwicklungsumgebungen.....	88

VI. Abbildungsverzeichnis

Abbildung 1: Parsen einer XML Datei	10
Abbildung 2: Logging.....	11
Abbildung 3: Quellcode der Klasse Konto.....	11
Abbildung 4: Quellcode der Klasse Konto incl. technischer Belange.....	12
Abbildung 5: Weben von Aspekten	13
Abbildung 6: AOP als Lichtstrahl.....	13
Abbildung 7: Use-Case-Diagramm Resource-Management-System	21
Abbildung 8: Klassendiagramm mit zentralen Klassen der Anwendung.....	23
Abbildung 9: Datenbank Design der Anwendung	24
Abbildung 10: Paper Prototyp der Benutzeroberfläche	24
Abbildung 11: Screenshot der Benutzeroberfläche	25
Abbildung 12: Architektur Übersicht	26
Abbildung 13: Eclipse IDE	30
Abbildung 14: Eclipse Architektur.....	30
Abbildung 15: AspectJ Laufzeit-Bibliothek.....	38
Abbildung 16: Vier Schichten Architektur	48
Abbildung 17: Sequenzdiagramm - Erzeugung und Benutzung von Ressourcen.....	53
Abbildung 18: Sequenzdiagramm – Verwendung eines Pools.....	54
Abbildung 19: Klassendiagramm Singleton Pattern.....	62
Abbildung 20: Klassendiagramm für Grafikprogramm	64
Abbildung 21: UML Diagramm für Mock Klassen.....	70
Abbildung 22: Downloads von AspectJ.....	74