

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals  
in den Bereichen Softwareentwicklung,  
Internettechnologie und IT-Management aktuell  
und kompetent relevantes Fachwissen über  
Technologien und Produkte zur Entwicklung  
und Anwendung moderner Informationstechnologien.

Jochen Seemann  
Jürgen Wolff von Gudenberg

# Software-Entwurf mit UML 2

Objektorientierte Modellierung  
mit Beispielen in Java

2. Auflage  
Mit 252 Abbildungen und CD-ROM

 Springer

Jochen Seemann  
DSL Tools Team  
Microsoft  
Redmond  
jochen@seemanns.net

Jürgen Wolff von Gudenberg  
Lehrstuhl für Informatik II  
Universität Würzburg  
Am Hubland  
97074 Würzburg  
wolff@informatik.uni-wuerzburg.de

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

ISSN 1439-5428  
ISBN-10 3-540-30949-7 Springer Berlin Heidelberg New York  
ISBN-13 978-3-540-30949-9 Springer Berlin Heidelberg New York  
ISBN-10 3-540-64103-3 1. Auflage Springer Berlin Heidelberg New York

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer ist nicht Urheber der Daten und Programme. Weder Springer noch die Autoren übernehmen die Haftung für die CD-ROM und das Buch, einschließlich ihrer Qualität, Handels- und Anwendungseignung. In keinem Fall übernehmen Springer oder die Autoren Haftung für direkte, indirekte, zufällige oder Folgeschäden, die sich aus der Nutzung der CD-ROM oder des Buches ergeben.

Springer ist ein Unternehmen von Springer Science+Business Media  
[springer.de](http://springer.de)

© Springer-Verlag Berlin Heidelberg 2000, 2006  
Printed in Germany

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften. Text und Abbildungen wurden mit größter Sorgfalt erarbeitet. Verlag und Autor können jedoch für eventuell verbliebene fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Satz: Druckfertige Daten der Autoren  
Herstellung: LE-TeX, Jelonek, Schmidt & Vöckler GbR, Leipzig  
Umschlaggestaltung: KünkelLopka Werbeagentur, Heidelberg  
Gedruckt auf säurefreiem Papier 33/3100 YL - 5 4 3 2 1 0

# Vorwort

Der objektorientierte Software-Entwurf ist weiterhin auf dem Vormarsch. Nachdem sich Programmiersprachen wie C++ und Java rasch verbreitet hatten, rückte die objektorientierte Analyse und der objektorientierte Entwurf in den Mittelpunkt des Interesses. Eine Vielzahl von Entwicklungsmethoden wurde in den 90er Jahren publiziert. Hinter all diesen Ansätzen steht der Wunsch, die Erstellung von Software von der bloßen Programmierung in einen systematischen, nachvollziehbaren Herstellungsprozess zu verwandeln. Weil fast jeder, der eine objektorientierte Programmiersprache einsetzt, Diagramme zeichnet, die seine Software dokumentieren, standen Entwurfsdiagramme schon immer im Zentrum der objektorientierten Methoden.

Im Zeichen zunehmender Komplexität, dem Trend zur Wiederverwendung von Software einerseits und der verteilten Entwicklung andererseits wurde der Wunsch nach einer einheitlichen Diagramm-Sprache für die Modellierung laut. Die Modellierungssprache UML (Unified Modeling Language), die sich innerhalb kurzer Zeit als Standard etabliert hat, stellt sich dieser Herausforderung. Darüberhinaus regte ihre Definition auch umfangreiche wissenschaftliche Aktivitäten an, und ihr Gebrauch wird immer mehr in Vorlesungen gelehrt.

UML ist eine Modellierungssprache, die für die Dokumentation der Analyse, des Entwurfs und der Implementierung objektorientierter Software entworfen wurde. Zu diesem Zweck vereinigt die UML eine Vielzahl von Beschreibungstechniken und unterstützt verschiedene Modelle. Sie beinhaltet kein Vorgehensmodell zur Software-Entwicklung, gibt aber den Rahmen eines Entwurfsprozesses vor.

Wir stellen im ersten Teil des Buches die visuelle, grafische Sprache UML, also die verschiedenen Untersprachen oder Diagrammarten, ähnlich wie eine Programmiersprache vor.

Das zentrale Anliegen des zweiten Teils ist der sinnvolle Einsatz dieser Mittel, die Kombination mehrerer Diagrammarten, um unser letztendliches Ziel, die kontrollierte und dokumentierte Erstellung von Software zu erreichen. So widmen wir uns in diesem Teil des Buches der Umsetzung eines UML-Modells in ausführbaren Code. Als Zielsprache verwenden wir Java.

Wir stellen ein einfaches Vorgehensmodell vor, das vorschreibt, wie man durch den Einsatz verschiedener UML-Diagramme Software entwirft. Die-

se Vorgehensweise wird verwendet, um ein vollständiges Beispiel, den Entwurf und die Implementierung einer Tabellenkalkulation, zu beschreiben. Die UML-Diagramme und das komplette Java Programm sind auf der beiliegenden CD enthalten.

Dieses Buch wendet sich in gleicher Weise an den Anwender wie an den Lernenden. Wünschenswert, aber nicht zwingend, sind Grundkenntnisse einer objektorientierten Programmiersprache wie C++ oder Java. Das Buch soll den Schritt zum objektorientierten Software-Entwurf unterstützen, indem es Umsteigern eine Einführung vermittelt und die teilweise neuen Beschreibungstechniken erläutert. Durch seinen übersichtlichen, aussagekräftigen Referenzteil am Schluss hilft das Buch auch Profis beim täglichen Einsatz der UML. Diese Zusammenfassung der UML ist zusätzlich als Hypertext auf der CD zu finden. Studenten und Neulinge werden das Buch sicher von Kapitel 1 bis Kapitel 15 lesen, während erfahrene Programmierer, die vor allem der Einsatz von UML zur Dokumentation der Software interessiert, vielleicht bei Kapitel 10 oder 11 anfangen werden und dann die einzelnen Kapitel des ersten Teiles nach Bedarf hinzuziehen.

Das Buch ist entstanden aus Vorlesungen an der Universität Würzburg sowie Schulungen der Firma 3SOFT in Erlangen. Im Verlauf der Fertigstellung des Buches wurde von der Object Management Group (OMG) eine Erweiterung des vorliegenden UML-Standards angeregt, deren Entwicklung sich bis in den Juli 1999 erstreckte. Diese neue Version 1.3 wurde vollständig in Buch und Referenz eingearbeitet.

Wir danken allen Kollegen, Studenten und Kursteilnehmern, die durch ihre Diskussionsbeiträge und Fragen zur Entstehung und Gestaltung des Buches beigetragen haben. Insbesondere sind hier Dr. Jürgen Schmied und Michael Lerch zu nennen, sowie Peter Fleischmann, der uns bei der Layout-„TeX“-nik unterstützte.

Den Mitarbeitern des Springer-Verlages danken wir für die Geduld, die sie für die in Terminnöte geratenen Autoren aufbrachten. Unser besonderer Dank gilt ferner unseren Familien, die durch ihre wohlwollende Unterstützung wesentlich zum Gelingen des Buches beitrugen.

Würzburg, im August 1999

Jochen Seemann  
Jürgen Wolff von Gudenberg

# Vorwort zur zweiten Auflage

Der Siegeszug der UML setzte sich in den vergangenen 5 Jahren fort. In diesem Sog wurde auch das vorliegende Buch so gut nachgefragt, dass eine zweite Auflage fällig wurde. Wir danken den interessierten Lesern.

Die Geschichte der 2. Auflage ähnelt dem Märchen vom Hase und vom Igel. Immer wenn sich die zwei Autoren zusammengesetzt hatten und meinten fertig zu sein, meldete sich der Hase UML mit einer neuen Version zurück.

Die Versionen 1.4 und 1.5 brachten neben vielen Klarstellungen und Änderungen der Semantik einige auch für ein einführendes Werk wichtige neue Sprachelemente. So erfolgte eine weiter gehende Angleichung an die Programmiersprache Java.

Aber schon vor der Fertigstellung der Version 1.5 hatte sich ein Prozess der totalen Neugestaltung der UML in Gang gesetzt, der im Jahre 2003 mit den ersten Entwürfen an die Öffentlichkeit trat. Neben einer neuen Fundierung auf das Metamodell wurden vor Allem die Sequenz- und Interaktionsdiagramme erheblich erweitert. Dem Trend zum Komponentenbasierten Programmieren wurde durch detailliertere Komponentendiagramme Rechnung getragen.

So musste das gesamte Buch neu geschrieben werden. Nach langen Diskussionen haben wir uns entschlossen, im Aufbau und in der Wahl der Beispiele der alten Vorlage treu zu bleiben. So wird in diesem Buch erstens die Syntax der UML 2.0 dargestellt, zweitens ein inzwischen auch in der Praxis bewährter Entwicklungsprozess vorgestellt und drittens die Umsetzung eines umfangreichen Beispiels in Java behandelt. Eine ausführliche UML Referenz im Anhang rundet das Buch ab.

Dank an alle Studenten und Leser, die uns auf Fehler oder Ungereimtheiten in der ersten Auflage hinwiesen, und an die, die bei der Neuauflage mithalfen, namentlich Holger Eichelberger und Gregor Fischer.

Würzburg, im November 2005

Jochen Seemann  
Jürgen Wolff von Gudenberg

# Inhaltsverzeichnis

---

## Teil I UML als Entwurfssprache

---

<b>1</b>	<b>Modellierung von Software-Systemen</b> .....	1
1.1	Entstehung der UML .....	2
1.2	Zum Aufbau des Buches .....	4
1.3	Modelle, Sichten und Diagramme .....	6
1.4	Das statische Modell .....	10
1.5	Das dynamische Modell .....	11
<b>2</b>	<b>Das Use-Case-Diagramm</b> .....	15
2.1	Anwendungsfälle .....	16
2.2	Das Anwendungsfalldiagramm .....	18
2.3	Verfeinerung von Anwendungsfällen .....	19
2.4	Beziehungen in Use-Case-Diagrammen .....	20
2.5	Zusammenfassung .....	24
<b>3</b>	<b>Das Aktivitätsdiagramm</b> .....	27
3.1	Abläufe und Vorgänge .....	28
3.2	Aufteilung des Kontrollflusses .....	30
3.3	Zuteilung der Verantwortung .....	32
3.4	Objektfluss .....	32
3.5	Aktivitäten und Aktionen .....	35
3.6	Algorithmen .....	35
3.7	Zusammenfassung .....	38
<b>4</b>	<b>Das Klassendiagramm</b> .....	43
4.1	Klassen .....	44
4.2	Attribute und Operationen .....	45
4.3	Verantwortlichkeiten .....	48
4.4	Objekte .....	49
4.5	Verknüpfungen .....	51
4.6	Assoziationen .....	51
4.6.1	Rollen und Richtungen .....	53
4.6.2	Aggregation und Komposition .....	57
4.6.3	Multiplizität .....	59



4.6.4	Objektselektoren . . . . .	60
4.6.5	Assoziation als Klasse . . . . .	61
4.6.6	Mehrstellige Assoziationen . . . . .	62
4.7	Generalisierung . . . . .	65
4.7.1	Verallgemeinerung – Spezialisierung . . . . .	65
4.7.2	Abstraktion . . . . .	66
4.7.3	Klassifikation . . . . .	67
4.8	Klassen und Schnittstellen . . . . .	69
4.9	Geschachtelte Klassen . . . . .	71
4.10	Schablonen . . . . .	71
4.11	Erläuternder Text . . . . .	72
4.11.1	Bedingungen . . . . .	73
4.11.2	Kommentare . . . . .	75
4.11.3	Eigenschaftslisten . . . . .	75
4.12	Stereotypen . . . . .	76
4.13	Zusammenfassung . . . . .	77
<b>5</b>	<b>Das Sequenzdiagramm . . . . .</b>	<b>79</b>
5.1	Nachrichtenaustausch . . . . .	80
5.2	Aktivitätszonen . . . . .	81
5.3	Asynchrone Nachrichten . . . . .	83
5.4	Erzeugung und Zerstörung von Objekten . . . . .	84
5.5	Regelung des Interaktionsablaufs . . . . .	85
5.6	Zeitachse . . . . .	91
5.7	Zusammenfassung . . . . .	92
<b>6</b>	<b>Weitere Interaktionsdiagramme . . . . .</b>	<b>95</b>
6.1	Das Kommunikationsdiagramm . . . . .	96
6.1.1	Objekte und Nachrichten . . . . .	96
6.1.2	Verknüpfungen . . . . .	97
6.1.3	Sequenznummern . . . . .	99
6.1.4	Erweiterung . . . . .	100
6.2	Das Timing-Diagramm . . . . .	102
6.2.1	Zeitverlauf . . . . .	102
6.2.2	Wertverlauf . . . . .	103
6.3	Das Interaktionsübersichtsdiagramm . . . . .	103
<b>7</b>	<b>Das Zustandsdiagramm . . . . .</b>	<b>105</b>
7.1	Zustandsautomaten . . . . .	106
7.2	Zustände und Ereignisse . . . . .	107
7.3	Verzweigungen . . . . .	109
7.4	Atomare und andauernde Aktivitäten . . . . .	111
7.5	Hierarchische Zustandsdiagramme . . . . .	113
7.5.1	Unterzustände . . . . .	113

7.5.2	History-Zustand . . . . .	114
7.5.3	Nebenläufige Zustände . . . . .	116
7.6	Protokollzustandsautomaten . . . . .	117
7.7	Zusammenfassung . . . . .	118
<b>8</b>	<b>Die Komponentendiagramme . . . . .</b>	<b>121</b>
8.1	Das Komponentendiagramm . . . . .	122
8.1.1	Logische Komponenten . . . . .	122
8.1.2	Physische Komponenten . . . . .	124
8.2	Das Installationsdiagramm . . . . .	126
8.2.1	Software-Architekturen . . . . .	127
8.3	Das Kompositionsdiagramm . . . . .	127
8.3.1	Strukturierte Klassen und Komponenten . . . . .	127
8.3.2	Anschlüsse oder Ports . . . . .	130
8.4	Das Kooperationsdiagramm . . . . .	131
8.4.1	Erweiterung von Klassendiagrammen . . . . .	132
8.4.2	Kooperationsmuster . . . . .	134
<b>9</b>	<b>Das Paketdiagramm . . . . .</b>	<b>137</b>
9.1	Pakete und Abhängigkeiten . . . . .	138
9.2	Zusammenwirken von Paketen . . . . .	140
9.3	Modelle . . . . .	144

---

## Teil II Anwendung der UML

---

<b>10</b>	<b>Ein Vorgehensmodell für den Software-Entwurf . . . . .</b>	<b>147</b>
10.1	Anforderungsermittlung . . . . .	148
10.2	Analyse . . . . .	152
10.2.1	Objekte finden . . . . .	152
10.2.2	Objekte strukturieren . . . . .	154
10.2.3	Verantwortlichkeiten verteilen . . . . .	157
10.3	Entwurf . . . . .	158
10.3.1	Systemarchitektur festlegen . . . . .	158
10.3.2	Klassenentwurf . . . . .	160
10.3.3	Schnittstellen spezifizieren . . . . .	161
10.3.4	Detailentwurf . . . . .	163
10.4	Implementierung . . . . .	163
10.4.1	Klassendefinitionen . . . . .	163
10.4.2	Methoden . . . . .	163
10.5	Bemerkungen . . . . .	165
10.6	Komponentenbasierte Implementierung . . . . .	167
10.7	Modularisierung . . . . .	168
10.7.1	Verfeinerung von Modellelementen . . . . .	169

10.7.2	Vergrößerung	170
10.7.3	Frühe Zerlegung des Modells	170
10.8	Das Vorgehensmodell – kurz gefasst	171
10.9	Einsatzgebiete der Diagramme	172
<b>11</b>	<b>UML und Java</b>	<b>175</b>
11.1	Klassendefinitionen	177
11.1.1	Attribute	177
11.1.2	Methoden	178
11.2	Beziehungen zwischen Klassen	180
11.2.1	Generalisierung	180
11.2.2	Assoziationen	182
11.2.3	Aggregation und Komposition	189
11.2.4	Vorgehen	190
11.3	Methodenrumpfe	190
11.3.1	Implementierung aus Kommunikationsdiagrammen	191
11.3.2	Implementierung aus Zustandsdiagrammen	194
11.3.3	Bedingungen	195
11.4	Pakete	196
11.5	Java Klassenbibliotheken	197
<b>12</b>	<b>Entwurfsmuster</b>	<b>203</b>
12.1	Einführung und Begriffsklärung	204
12.2	Das Kompositum-Muster	205
12.3	Das Beobachter-Muster	210
12.3.1	Die Model-View-Controller Architektur	214
12.4	Das Adapter-Muster	215
12.5	Das Kommando-Prozessor-Muster	219
12.6	Das Status-Muster	221
12.7	Der Asynchrone Methodenaufruf	227
12.8	Software-Entwurf mit Entwurfsmustern	229
<b>13</b>	<b>Fallstudie: Eine Tabellenkalkulation</b>	<b>233</b>
13.1	Einführung	234
13.1.1	Tabellenkalkulation	234
13.1.2	Entwicklungsvorgehen	234
13.2	Anforderungsermittlung	236
13.2.1	Szenarien in Form von Aktivitätsdiagrammen	237
13.2.2	Weitere Anforderungen	242
13.3	Analyse und Entwurf	243
13.3.1	Festlegen der Software-Architektur	243
13.3.2	Die Tabelle	245
13.3.3	Der Formel-Parser	254
13.3.4	Die Benutzeroberfläche	258

13.3.5 Kopplung der Subsysteme .....	264
13.4 Implementierung in Java .....	272

---

## Teil III Formale Grundlagen der UML

---

<b>14 Erweiterungsmechanismen</b> .....	277
14.1 Präzisierung .....	278
14.2 Zusätzliche Information .....	278
14.3 Vereinbarung von Stereotypen .....	279
14.4 Profile .....	281
14.5 Die OCL .....	281
14.5.1 Überblick .....	281
14.5.2 Einführung der OCL an einem Beispiel .....	282
<b>15 Das UML-Metamodell</b> .....	287
15.1 Modell und Metamodell .....	288
15.2 Die 4-Schichten Architektur von UML .....	289
15.3 Zusammenfassung .....	291

---

## Teil IV Anhang

---

<b>A Die UML Referenz</b> .....	295
A.1 Gemeinsame Elemente aller Diagramme .....	296
A.2 Use-Case-Diagramm .....	297
A.3 Aktivitätsdiagramm .....	300
A.4 Klassendiagramm .....	307
A.5 Sequenzdiagramm .....	317
A.6 Weitere Interaktionsdiagramme .....	323
A.6.1 Kommunikationsdiagramm .....	323
A.6.2 Timing-Diagramm .....	326
A.6.3 Interaktionsübersichtsdiagramm .....	329
A.7 Zustandsdiagramm .....	330
A.8 Komponentendiagramme .....	337
A.8.1 Komponentendiagramm .....	337
A.8.2 Installationsdiagramm .....	340
A.8.3 Kompositionsdiagramm .....	342
A.8.4 Kooperationsdiagramm .....	343
A.9 Paketdiagramm .....	346
A.10 Die OCL-Syntax .....	349
A.11 Vordefinierte Stereotypen .....	350
A.12 Vordefinierte Bedingungen .....	351

<b>B Inhalt der CD-ROM</b> .....	353
<b>Literatur</b> .....	355
<b>Index</b> .....	357