



Objektorientierte Softwareentwicklung

von

Bernd Oestereich

unter Mitarbeit von Stefan Bremer

9., aktualisierte und erweiterte Auflage

Oldenbourg Verlag München

Inhaltsüberblick

1 Einführung	13
Die besonderen Merkmale der objektorientierten Softwareentwicklung, die Historie der UML und objektorientierter Methoden sowie die Unterschiede zu den alten Methoden werden in diesem Kapitel dargestellt.	
2 Grundkonzepte der Objektorientierung	31
Hier finden Sie eine einfache Einführung in die Grundbegriffe und -konzepte der Objektorientierung.	
3 Methodik: Analyse und Design	87
Anhand eines durchgängigen Beispiels werden aufeinander aufbauend die einzelnen Schritte der objektorientierten Analyse und des Designs erläutert. Die praktische Anwendung der einzelnen UML-Konzepte wird gezeigt. Zahlreiche Querverweise in das UML-Grundlagenkapitel erleichtern das Verständnis.	
Analyse	89
Design	171
4 UML-Grundlagen (Notation und Semantik)	239
Die Notation und Semantik der einzelnen UML-Konzepte wird detailliert und kritisch erläutert, begleitet von zahlreichen Tipps und Tricks für die Praxis.	
Anwendungsfalldiagramm	244
Klassendiagramm (Strukturelemente)	273
Klassendiagramm (Beziehungselemente)	298
Subsystem-, Paket- und Komponentendiagramm	323
Einsatz- und Verteilungsdiagramm	332
Aktivitätsdiagramm	335
Zustandsdiagramm	350
Kommunikationsdiagramm	357
Sequenzdiagramm	361
Zeitdiagramm	367
Zusicherungen	369
5 Anhang	381
Glossar	382
Übersetzungen	396
Literatur	399
Index	405

Inhaltsverzeichnis

	Einführung	13
1.1	Objektorientierte Softwareentwicklung	15
1.2	Die Historie	17
1.3	Notationen, Methoden, Vorgehensmodelle und Co.	19
1.4	Die Unified Modeling Language	21
1.5	Ganzheitliche Herangehensweise	24
1.6	Weiterführende Literatur	26
	Grundkonzepte der Objektorientierung	31
2.1	Grundidee der Objektorientierung	33
2.2	Klassen, Objekte, Instanzen, Exemplare & Co.	34
2.3	Attribute, Operationen, Zusicherungen, Beziehungen	36
2.4	Objektidentität	38
2.5	Verantwortlichkeiten	39
2.6	Demeter Gesetz	41
2.7	Taxonomie und Vererbung	44
2.7.1	Strukturierung von Eigenschaften.....	46
2.7.2	Vererbung: Restriktionen und Probleme.....	50
• 2.8	Abstrakte Klassen	53
f 2.9	Assoziationen	54
T 2.10	Aggregationen	55
2.11	Nachrichtenaustausch	58
2.12	Sammlungen (Collections)	62
i 2.13	Polymorphie	64
2.14	Persistenz	67
2.15	Klassifizierung von Klassen	70
2.15.1	Entitätsklasse «entity».....	71
2.15.2	Steuerungsklasse «control».....	72
2.15.3	Schnittstellenklasse «interface».....	73
: 2.15.4	Schnittstellenobjekt «boundary».....	73
2.15.5	Typ «type».....	74
ï2.15.6	Primitive Klasse «primitive».....	75

2.15.7	Datentyp, Datenstruktur «dataType».....	75
2.15.8	Aufzählung «enumeration».....	76
2.15.9	Konfigurationswert «configvalues».....	77
2.16	Entwurfsmuster	78
2.17	Komponenten	81
2.18	Serviceorientierte Architekturen	84
2.19	Weiterführende Literatur	86
3	Methodik ^{^^^^}	
3.1	Analyse	89
3.1.1	Systemidee und Zielsetzung entwickeln.....	91
3.1.2	Interessenhalter identifizieren.....	96
3.1.3	Interessen der Interessenhalter identifizieren.....	102
3.1.4	Systemkontext aufnehmen.....	104
3.1.5	Geschäftsanwendungsfälle identifizieren.....	106
3.1.6	Anwendungsfälle essenziell beschreiben.....	112
3.1.7	Materialsammlung und-Studie.....	120
3.1.8	Systemanwendungsfälle identifizieren.....	122
3.1.9	Fachklassen, Aufzählungen und Konfigurationswerte identifizieren.....	130
3.1.10	Fachliches Glossar anlegen.....	134
3.1.11	Systemablaufmodelle entwickeln.....	142
3.1.12	Systemanwendungsfallmodell erstellen.....	153
3.1.13	Übrige fachliche Anforderungen und Regeln beschreiben.....	158
3.1.14	Systemschnittstelle beschreiben.....	163
3.1.15	Exploratives Schnittstellen-Prototyping.....	167
3.2	Design	171
3.2.1	Von der Analyse zum Design.....	173
3.2.2	Schichtenmodell definieren.....	175
3.2.3	Verteilungsmodell definieren.....	180
3.2.4	Fachliches Subsystemmodell definieren.....	181
3.2.5	Ablaufverantwortlichkeiten festlegen.....	185
3.2.6	Komponentenspezifische Klassenmodelle entwickeln.....	187
3.2.7	Komponentenschnittstellen entwerfen.....	190
3.2.8	Zustandsmodelle (weiter-) entwickeln.....	193
3.2.9	Objektfluss modellieren.....	195
3.2.10	Interaktionsmodelle entwickeln.....	198
3.2.11	Attribute definieren.....	201
3.2.12	Dialoge spezifizieren.....	205
3.2.13	Design-Diskurs.....	208
3.2.14	Testgetriebene Entwicklung.....	219
3.2.15	Ablauforientierte Komponententests entwickeln.....	228
3.2.16	Klassentests entwickeln.....	232
3.2.17	Testmethoden.....	235

4	UML-Grundlagen.....	239
4.1	Einleitung.....	241
4.1.1	Diagramme im Überblick.....	243
4.2	Anwendungsfälle und Anforderungen.....	244
4.2.1	Anwendungsfalldiagramm.....	245
4.2.2	Systemkontextdiagramm.....	247
4.2.3	Realisierung von Anwendungsfällen.....	248
4.2.4	Spezialisierung von Anwendungsfällen.....	248
4.2.5	Enthältbeziehung, Erweiterungsbeziehung.....	250
4.2.6	Assoziation in Anwendungsfalldiagrammen.....	251
4.2.7	Anwendungsfall.....	252
4.2.8	Geschäftsanwendungsfall.....	258
4.2.9	Systemanwendungsfall.....	260
4.2.10	Sekundärer Anwendungsfall.....	261
4.2.11	Abstrakter Anwendungsfall.....	262
4.2.12	Anforderung, Feature & Co.....	264
4.2.13	Akteur.....	269
4.2.14	Anwendungsfallsszenario.....	271
4.3	Klassendiagramm, Strukturelemente.....	273
4.3.1	Klasse.....	274
4.3.2	Parametrisierbare Klasse.....	276
4.3.3	Abstrakte Klasse.....	278
4.3.4	Aktive Klasse.....	279
4.3.5	Objekt.....	280
4.3.6	Attribut.....	281
4.3.7	Operation.....	285
4.3.8	Verantwortlichkeit.....	287
4.3.9	Enumeration.....	288
4.3.10	Schnittstellen.....	289
4.3.11	Eigenschaftswert.....	293
4.3.12	Stereotyp.....	294
4.3.13	Notiz.....	297
4.4	Beziehungselemente.....	298
4.4.1	Generalisierung, Spezialisierung.....	299
4.4.2	Assoziation.....	303
4.4.3	Gerichtete Assoziation.....	306
4.4.4	Attributierte Assoziation.....	308
4.4.5	Qualifizierte Assoziation.....	311
4.4.6	Mehrgliedrige Assoziation.....	313
4.4.7	Aggregation.....	314
4.4.8	Komposition.....	316
4.4.9	Abhängigkeitsbeziehung.....	318
4.5	Weitere Strukturdiagramme.....	322
4.5.1	Objektdiagramm.....	322

