

Inhalt

Vorwort	17
1 Einführung	19
1.1 Einleitung und allgemeine Hinweise	19
1.1.1 Für wen ist dieses Buch gedacht?	19
1.1.2 Muster erkennen	20
1.1.3 Muster im weiteren Sinne	20
1.1.4 Programmiersprachen und Frameworks	21
1.1.5 Babylon I: Deutsch vs. Englisch	21
1.1.6 Babylon II: Verwendung von Begriffen	22
1.1.7 UML	22
1.1.8 Auf den Schultern von Riesen	23
1.1.9 Der Aufbau eines Entwurfsmusters im Buch	24
1.1.10 Die Implementierung und die verwendete Programmiersprache	24
1.1.11 Herstellerspezifische Technologien	25
1.2 Was sind Entwurfsmuster und was sind sie nicht?	25
1.2.1 Was Entwurfsmuster sind	26
1.2.2 Was Entwurfsmuster nicht sind	27
1.3 Der OO-Werkzeugkasten	30
1.3.1 Schnittstellen-Implementierung vs. Klassen-Vererbung	30
1.3.2 Is-A vs. Has-A	32
1.4 Ein kleines Beispiel aus der Praxis	33
1.4.1 Die Anforderung	33
1.4.2 Und dann kommt die Änderung der Anforderung	34
1.4.3 Der Versuch, das Ganze doch noch irgendwie hinzubekommen	35
1.4.4 Entwurfsmuster als Lösung	36
1.5 Überlegungen zum Einsatz	37
1.5.1 Erkenne das Muster	37
1.5.2 Dokumentation und Bezeichnung	38
1.5.3 Anti-Patterns	39
1.5.4 Verhältnismäßigkeit	39
1.6 Wegweiser	40
1.6.1 Erzeugungsmuster (Creational Design Patterns)	41
1.6.2 Strukturmuster (Structural Design Patterns)	42

1.6.3	Verhaltensmuster (Behavioral Design Patterns)	43
1.6.4	Architekturmuster (Architectural Design Patterns)	45
1.6.5	Datenmuster (Data Design Patterns)	45
1.6.6	GUI-Muster	47
1.6.7	Design- und Entwicklungsprinzipien (Design Principles)	47

2 Erzeugungsmuster

49

2.1	Fabrikmethode	51
2.1.1	Steckbrief	51
2.1.2	Beschreibung	51
2.1.3	Anwendungsfälle	53
2.1.4	Implementierung	54
2.1.5	Weitere Überlegungen und Alternativen	57
2.2	Singleton	64
2.2.1	Steckbrief	64
2.2.2	Beschreibung	65
2.2.3	Anwendungsfälle	66
2.2.4	Implementierung	67
2.2.5	Weitere Überlegungen und Alternativen	69
2.2.6	Vollständige Implementierung	73
2.3	Multiton	74
2.3.1	Steckbrief	74
2.3.2	Beschreibung	74
2.3.3	Anwendungsfälle	75
2.3.4	Implementierung	75
2.3.5	Weitere Überlegungen und Alternativen	77
2.3.6	Vollständige Implementierung	77
2.4	Abstrakte Fabrik	78
2.4.1	Steckbrief	78
2.4.2	Beschreibung	79
2.4.3	Anwendungsfälle	81
2.4.4	Implementierung	83
2.4.5	Weitere Überlegungen und Alternativen	89
2.5	Erbauer	92
2.5.1	Steckbrief	93
2.5.2	Beschreibung	93
2.5.3	Anwendungsfälle	94

2.5.4	Implementierung	95
2.5.5	Weitere Überlegungen und Alternativen	101
2.6	Prototyp	101
2.6.1	Steckbrief	102
2.6.2	Beschreibung	102
2.6.3	Anwendungsfälle	103
2.6.4	Implementierung	104
2.6.5	Weitere Überlegungen und Alternativen	109
3	Strukturmuster	113
3.1	Adapter	114
3.1.1	Steckbrief	114
3.1.2	Beschreibung	114
3.1.3	Anwendungsfälle	117
3.1.4	Implementierung	119
3.1.5	Weitere Überlegungen und Alternativen	128
3.2	Brücke	129
3.2.1	Steckbrief	130
3.2.2	Beschreibung	130
3.2.3	Anwendungsfälle	132
3.2.4	Implementierung	135
3.2.5	Weitere Überlegungen und Alternativen	140
3.3	Kompositum	142
3.3.1	Steckbrief	142
3.3.2	Beschreibung	142
3.3.3	Anwendungsfälle	145
3.3.4	Implementierung	147
3.3.5	Weitere Überlegungen und Alternativen	152
3.4	Dekorierer	155
3.4.1	Steckbrief	155
3.4.2	Beschreibung	155
3.4.3	Anwendungsfälle	157
3.4.4	Implementierung	160
3.4.5	Weitere Überlegungen und Alternativen	167
3.5	Fassade	169
3.5.1	Steckbrief	169
3.5.2	Beschreibung	169

- 3.5.3 Anwendungsfälle 171
 - 3.5.4 Implementierung 172
 - 3.5.5 Weitere Überlegungen und Alternativen 178
- 3.6 Fliegengewicht 182**
 - 3.6.1 Steckbrief 183
 - 3.6.2 Beschreibung 183
 - 3.6.3 Anwendungsfälle 187
 - 3.6.4 Implementierung 189
 - 3.6.5 Weitere Überlegungen und Alternativen 194
- 3.7 Proxy 198**
 - 3.7.1 Steckbrief 198
 - 3.7.2 Beschreibung 198
 - 3.7.3 Anwendungsfälle 200
 - 3.7.4 Implementierung 204
 - 3.7.5 Weitere Überlegungen und Alternativen 208

4 Verhaltensmuster

211

- 4.1 Zuständigkeitskette 211**
 - 4.1.1 Steckbrief 211
 - 4.1.2 Beschreibung 211
 - 4.1.3 Anwendungsfälle 213
 - 4.1.4 Implementierung 214
 - 4.1.5 Weitere Überlegungen und Alternativen 220
- 4.2 Befehl 222**
 - 4.2.1 Steckbrief 222
 - 4.2.2 Beschreibung 223
 - 4.2.3 Anwendungsfälle 226
 - 4.2.4 Implementierung 228
 - 4.2.5 Weitere Überlegungen und Alternativen 237
- 4.3 Interceptor 240**
 - 4.3.1 Steckbrief 241
 - 4.3.2 Beschreibung 241
 - 4.3.3 Anwendungsfälle 245
 - 4.3.4 Implementierung 246
 - 4.3.5 Weitere Überlegungen und Alternativen 251

4.4	Interpreter	253
4.4.1	Steckbrief	253
4.4.2	Beschreibung	253
4.4.3	Anwendungsfälle	258
4.4.4	Implementierung	259
4.4.5	Weitere Überlegungen und Alternativen	270
4.5	Iterator	271
4.5.1	Steckbrief	272
4.5.2	Beschreibung	272
4.5.3	Anwendungsfälle	277
4.5.4	Implementierung	278
4.5.5	Weitere Überlegungen und Alternativen	283
4.6	Vermittler	287
4.6.1	Steckbrief	287
4.6.2	Beschreibung	288
4.6.3	Anwendungsfälle	291
4.6.4	Implementierung	293
4.6.5	Weitere Überlegungen und Alternativen	301
4.7	Memento	303
4.7.1	Steckbrief	303
4.7.2	Beschreibung	303
4.7.3	Anwendungsfälle	305
4.7.4	Implementierung	307
4.7.5	Weitere Überlegungen und Alternativen	312
4.8	Beobachter	313
4.8.1	Steckbrief	314
4.8.2	Beschreibung	314
4.8.3	Anwendungsfälle	318
4.8.4	Implementierung	319
4.8.5	Weitere Überlegungen und Alternativen	323
4.9	Zustand	331
4.9.1	Steckbrief	331
4.9.2	Beschreibung	331
4.9.3	Anwendungsfälle	335
4.9.4	Implementierung	336
4.9.5	Weitere Überlegungen und Alternativen	341
4.10	Strategie	343
4.10.1	Steckbrief	343
4.10.2	Beschreibung	343

- 4.10.3 Anwendungsfälle 345
 - 4.10.4 Implementierung 346
 - 4.10.5 Weitere Überlegungen und Alternativen 350
- 4.11 Schablonenmethode 351**
 - 4.11.1 Steckbrief 352
 - 4.11.2 Beschreibung 352
 - 4.11.3 Anwendungsfälle 354
 - 4.11.4 Implementierung 355
 - 4.11.5 Weitere Überlegungen und Alternativen 357
- 4.12 Besucher 358**
 - 4.12.1 Steckbrief 358
 - 4.12.2 Beschreibung 358
 - 4.12.3 Anwendungsfälle 364
 - 4.12.4 Implementierung 365
 - 4.12.5 Weitere Überlegungen und Alternativen 370

5 Muster verteilter Architekturen 375

- 5.1 Kleine Architekturmusterkunde mit Historie 376**
 - 5.1.1 Am Anfang war der Monolith 376
 - 5.1.2 DLLs (Dynamic Link Libraries) 377
 - 5.1.3 Shared DLLs 378
 - 5.1.4 Client-Server 379
 - 5.1.5 Serviceorientierte Architekturen (SOA) 381
 - 5.1.6 Peer to Peer 383
- 5.2 The (8) Fallacies of Distributed Computing 384**
 - 5.2.1 Irrtum No. 1: Das Netzwerk arbeitet zuverlässig 384
 - 5.2.2 Irrtum No. 2: Die Latenzzeit ist 0 385
 - 5.2.3 Irrtum No. 3: Die Bandbreite ist unbegrenzt 388
 - 5.2.4 Irrtum No. 4: Das Netzwerk ist sicher 388
 - 5.2.5 Irrtum No. 5: Die Netzwerktopologie ändert sich nicht 389
 - 5.2.6 Irrtum No. 6: Es gibt nur einen Administrator 390
 - 5.2.7 Irrtum No. 7: Der Datentransport ist umsonst 391
 - 5.2.8 Irrtum No. 8: Das Netzwerk ist homogen 391
- 5.3 Serviceorientierte Architekturen 391**
 - 5.3.1 Definition und Merkmale von SOA und Services 392
 - 5.3.2 Kommunikationsmuster 396
 - 5.3.3 Weitere Überlegungen 399

5.4	Event Sourcing	406
5.4.1	Steckbrief	406
5.4.2	Beschreibung	406
5.4.3	Anwendungsfälle	412
5.4.4	Weitere Überlegungen und Alternativen	412
5.5	Command Query Responsibility Segregation (CQRS)	415
5.5.1	Steckbrief	416
5.5.2	Beschreibung	416
5.5.3	Anwendungsfälle	425
5.5.4	Implementierung	426
5.5.5	Weitere Überlegungen und Alternativen	428
6	Datenmuster	431
<hr/>		
6.1	Unit of Work	432
6.1.1	Steckbrief	432
6.1.2	Beschreibung	432
6.1.3	Anwendungsfälle	434
6.1.4	Implementierung	436
6.1.5	Weitere Überlegungen und Alternativen	439
6.2	Transaktionen	443
6.2.1	Teilnehmer an einer Transaktion	443
6.2.2	Tugenden	444
6.2.3	ACID	445
6.2.4	Ergebnisse einer Transaktion: Commit und Rollback	446
6.2.5	Parallele Transaktionen	447
6.2.6	Mögliche Probleme paralleler Transaktionen	448
6.2.7	Isolation Levels	454
6.2.8	Verteilte Transaktionen	457
6.3	Datentransferobjekt (DTO)	458
6.3.1	Steckbrief	459
6.3.2	Beschreibung	459
6.3.3	Anwendungsfälle	460
6.3.4	Implementierung	461
6.3.5	Weitere Überlegungen und Alternativen	462
6.4	Table Data Gateway	467
6.4.1	Steckbrief	467
6.4.2	Beschreibung	468

6.4.3	Anwendungsfälle	470
6.4.4	Implementierung	471
6.4.5	Weitere Überlegungen und Alternativen	473
6.5	Row Data Gateway	475
6.5.1	Steckbrief	475
6.5.2	Beschreibung	476
6.5.3	Anwendungsfälle	477
6.5.4	Implementierung	477
6.5.5	Weitere Überlegungen und Alternativen	480
6.6	Identity Map bzw. Identity Function	482
6.6.1	Steckbrief	482
6.6.2	Beschreibung	482
6.6.3	Anwendungsfälle	484
6.6.4	Implementierung	485
6.6.5	Weitere Überlegungen und Alternativen	486
6.7	Optimistisches Sperren	487
6.7.1	Steckbrief	487
6.7.2	Beschreibung	487
6.7.3	Anwendungsfälle	491
6.7.4	Implementierung	492
6.7.5	Weitere Überlegungen und Alternativen	492
6.8	Pessimistisches Sperren	495
6.8.1	Steckbrief	495
6.8.2	Beschreibung	495
6.8.3	Anwendungsfälle	501
6.8.4	Implementierung	501
6.8.5	Weitere Überlegungen und Alternativen	506
6.9	Vererbung	507
6.9.1	Klassenhierarchie	508
6.9.2	Eine Tabelle für alle Klassen	509
6.9.3	Eine Tabelle für jede Klasse	511
6.9.4	Eine Tabelle für jede konkrete Klasse	515
7	GUI-Muster	519
7.1	Model View Controller (MVC)	519
7.1.1	Steckbrief	520
7.1.2	Beschreibung	520

7.1.3	Anwendungsfälle	526
7.1.4	Implementierung	527
7.1.5	Weitere Überlegungen und Alternativen	538
7.2	Model View Presenter (MVP)	541
7.2.1	Steckbrief	541
7.2.2	Beschreibung	542
7.2.3	Anwendungsfälle	546
7.2.4	Weitere Überlegungen und Alternativen	547
7.3	Model View ViewModel (MVVM)	549
7.3.1	Steckbrief	549
7.3.2	Beschreibung	549
7.3.3	Anwendungsfälle	552
7.3.4	Implementierung	552
7.3.5	Weitere Überlegungen und Alternativen	562
8	Design- und Entwicklungsprinzipien	563
<hr/>		
8.1	Merkmale schlechten Designs	563
8.1.1	Starre, unflexible Software	564
8.1.2	Fragile Software	564
8.1.3	Nicht wiederverwendbare Software	565
8.1.4	Unnötige Wiederholungen	565
8.1.5	Viskosität	566
8.1.6	Unnötige Komplexität	566
8.1.7	Undurchsichtige Software (opaque Software)	567
8.2	SOLID	567
8.2.1	S – Single-Responsibility-Prinzip	568
8.2.2	O – Open-Closed-Prinzip	571
8.2.3	L – Liskovsches Substitutionsprinzip (LSP)	575
8.2.4	I – Interface-Segregation-Prinzip	579
8.2.5	D – Dependency-Inversion-Prinzip	581
8.3	Das agile Manifest	584
8.3.1	Agilität allenthalben	584
8.3.2	Die Werte	586
8.3.3	Prinzipien	588
8.3.4	You Ain't Gonna Need It (YAGNI)	589

- 8.4 Designprinzipien** 591
 - 8.4.1 Keep It Short and Simple (KISS) 592
 - 8.4.2 Don't Repeat Yourself (DRY) 593
 - 8.4.3 Separation of Concerns (SoC) 595
 - 8.4.4 Das Gesetz von Demeter 596
 - 8.4.5 Design by Contract 597
 - 8.4.6 Das Selbstdokumentationsprinzip 599
 - 8.4.7 Single-Choice-Prinzip 599
- 8.5 Design Smells und Anti-Patterns zu Abstraktionen** 601
 - 8.5.1 Zu viel Abstraktion 603
 - 8.5.2 Zu wenig Abstraktion 605
 - 8.5.3 Unvollständige Abstraktion 607
 - 8.5.4 Abstraktionen ohne ausreichende Abgrenzungen 608
 - 8.5.5 Weitere Probleme mit Abstraktionen 609
- 8.6 Design Smells und Anti-Patterns zur Kapselung** 613
 - 8.6.1 Fehlende oder lückenhafte Kapselung 614
 - 8.6.2 Über Modularität 616
 - 8.6.3 Zyklische Benutzungsbeziehungen 616
 - 8.6.4 Zu viele Beziehungen 619
- 8.7 Design Smells und Anti-Patterns zu Hierarchien** 620
 - 8.7.1 Zu komplexe bzw. unnötige Hierarchie 620
 - 8.7.2 Zu flache bzw. fehlende Hierarchie 623
 - 8.7.3 Rebellierende Hierarchien 625
 - 8.7.4 Zyklische Hierarchien 628
 - 8.7.5 Unechte Hierarchien 628
 - 8.7.6 Redundante Hierarchien bzw. mehrere Pfade in Hierarchien 630
- Literaturverzeichnis 633
- Index 637