



# Index



## A

- Abstract Factory-Muster 156
  - Siehe auch* Factory-Muster
- Adapter-Muster
  - Definition 243
  - Einführung 237
  - Enten-Sticker 245
  - Enumeration/Iterator-Adapter 248
  - Erläuterung 241
  - Kamingespräche 247, 252 – 253
  - Klassenadapter 244
  - Klassendiagramm 243
  - Objektadapter 244
  - Übung 251
  - Vorteile 242
    - zusammengesetzte Muster 504
- Alexander, Christopher 602
- Algorithmen kapseln 286, 289
- Allheilmittel 607
- Anfragen kapseln 206
- Antimuster 606 – 607
  - Allheilmittel 607
- Anwendungsmuster 604
- Architekturmuster 604

## B

- bereichsspezifische Muster 604
- besser leben mit Mustern 578
- Bridge-Muster 612 – 613
- Builder-Muster 614 – 615

## C

- CD-Cover-Viewer 463
- Chain of Responsibility-Muster 616 – 617
- Code-Magnet 69, 179, 245, 350
- Command-Muster
  - Anfragen protokollieren 229
  - Anfrage-Warteschlangen 228
  - Befehl-Objekt 203
  - Definition 206 – 207
  - den Aufrufer laden 201
  - Einführung 196
  - Klassendiagramm 207
  - Makro-Befehle 224
  - Null-Objekt 214
  - Rückgängig 216, 220, 227
- Composite-Muster
  - Definition 356
  - Interview 376 – 377
  - Klassendiagramm 358
  - Sicherheit 367
  - Sicherheit vs. Transparenz 515

Standardverhalten 360  
 Transparenz 367, 375  
 und Iterator-Muster 368  
 zusammengesetzte Muster 363

## D

Decorator-Muster  
 Definition 91  
 Einführung 88  
 Gespräch am Arbeitsplatz 93  
 in Java I/O 100 – 101  
 Interview 104  
 Kamingespräch 252 – 253  
 Klassendiagramm 91  
 Nachteile 101, 104  
 Strukturmuster 591  
 und Proxy-Muster 472 – 473  
 zusammengesetzte Muster 506  
 Demeters Gesetz *siehe* Prinzip der Umkehrung der  
 Abhängigkeiten  
 Denken in Mustern 594 – 595  
 Design-Puzzle 25, 133, 279, 395, 468, 542  
 Design-Werkzeugkasten 32, 74, 105, 162, 186, 230, 270,  
 311, 380, 423, 491, 560, 608  
 DJ-View 534

## E

ECOOP 603  
 Einfache Fabrik 117  
 Eins-zu-Viele-Beziehung 51 – 52  
 Elvis 526  
 eng gebunden 53  
 Entwurfsmuster

Abstract Factory-Muster 156  
 Adapter-Muster 243  
 Bridge-Muster 612 – 613  
 Builder-Muster 614 – 615  
 Chain of Responsibility-Muster 616 – 617  
 Command-Muster 206  
 Composite-Muster 356  
 Decorator-Muster 91  
 Definition 579, 581  
 eigene entdecken 586 – 587  
 Einfache Fabrik 114  
 Facade-Muster 264  
 Factory Method-Muster 134  
 Flyweight-Muster 618 – 619  
 Interpreter-Muster 620 – 621  
 Iterator-Muster 336  
 Kategorien 589, 592 – 593  
 Klassenmuster 591  
 Mediator-Muster 622 – 623  
 Memento-Muster 624 – 625  
 Null-Objekt 214  
 Objektmuster 591  
 Observer-Muster 51  
 organisieren 589  
 Prototype-Muster 626 – 627  
 Proxy-Muster 460  
 Singleton-Muster 177  
 State-Muster 410  
 Strategy-Muster 24  
 Template Method-Muster 289  
 Verwendung 29  
 Visitor-Muster 628 – 629  
 Vorteile 599

vs. Bibliotheken 29  
 vs. Frameworks 29  
 Entwurfsmuster für Benutzerschnittstellen 605  
 Entwurfsprinzipien *siehe* objektorientierte Entwurfsprinzipien

## F

Facade-Muster  
 Definition 264  
 Einführung 258  
 Klassendiagramm 264  
 und Prinzip der Verschwiegenheit 269  
 Vorteile 260

Factory Method-Muster 134  
*Siehe auch* Factory-Muster

Factory-Muster  
 Abstract Factory  
 Definition 156  
 Einführung 153  
 Interview 158 – 159  
 Klassendiagramm 156 – 157  
 und Factory Method 158 – 161  
 zusammengesetzte Muster 508

Einfache Fabrik  
 Definition 117  
 Einführung 114

Factory Method  
 Definition 134  
 Einführung 120, 131 – 132  
 Interview 158 – 159  
 Klassendiagramm 134  
 und Abstract Factory 160 – 161  
 Unter der Lupe 125  
 Vorteile 135

Familie von Algorithmen *siehe* Strategy-Muster  
 Familie von Produkten 145  
 Fernsteuerung 193, 209  
 Flyweight-Muster 618 – 619  
 Friedman, Dan 171

## G

Gamma, Erich 601  
 Gang of Four *siehe* Viererbande  
 gemeinsames Vokabular 26 – 28, 599 – 600  
 Geschäftsprozessmuster 605  
 Gespräch am Arbeitsplatz 55, 93, 195, 208, 387, 397, 433, 583 – 584  
 globaler Zugriffspunkt 177

## H

Halten Sie es einfach 594  
 HAS-A 23  
 Heimautomatisierung oder Krise 192  
 Heimkino 255  
 Helm, Richard 601  
 Herausgeber/Abonnent 45  
 Hollywood-Prinzip 296  
 und Prinzip der Umkehrung der Abhängigkeiten 298  
 Hot-or-Not 475

## I

Interpreter-Muster 620 – 621  
 IST-EIN 23  
 Iterationen kapseln 323  
 Iterator-Muster  
 Code-Magnet 350  
 Definition 336

Einführung 325  
 externer Iterator 338  
 for/in 349  
 interner Iterator 338  
 java.util.Iterator 332  
 Klassendiagramm 337  
 Null-Iterator 372  
 Objekte entfernen 332  
 polymorphe Iteration 338  
 Übung 327  
 und Collections 347 – 349  
 und Composite-Muster 368  
 und Enumeration 338  
 und Hashtable 343, 348  
 Vorteile 330

## J

Johnson, Ralph 601

## K

Kamingespräche 62, 247, 252, 308, 418, 472 – 473  
 kapseln, was veränderlich ist 8 – 9, 75, 136, 397, 612  
 Kaugummiautomat-Überwachung 431  
 Kaukugel & Co KG 386  
 Klassenexplosion 81  
 Kohäsion 339 – 340  
 koller koller 239  
 Komposition 23, 85, 93, 247, 309  
 Komposition der Vererbung vorziehen 23, 75  
 Kreuzworträtsel 33, 76, 163, 187, 231, 271, 310, 378, 490  
 Kurzdrama 48, 478

## L

Lernprinzipien dieser Reihe xxviii  
 lose Bindung 53

## M

machen Sie Ihren Kopf mustergültig 597  
 Mächte 582  
 Mediator-Muster 622 – 623  
 Meister und Schüler 23, 30, 85, 136, 592, 596  
 Memento-Muster 624 – 625  
 Methodenaufrufe kapseln 206  
 Model 2 549  
     *Siehe auch* Model-View-Controller  
     und Entwurfsmuster 557 – 558  
 Model-View-Controller  
     Adapter-Muster 546  
     Code-Fertiggericht 564 – 576  
     Composite-Muster 532, 559  
     Einführung 529  
     Lied 526  
     Mediator-Muster 559  
     Observer-Muster 532  
     Strategy-Muster 532, 545  
     und das Web 549  
     und Entwurfsmuster 532  
     unter der Lupe 530  
 Muster Ehrenhalber 117, 214  
 Muster im wirklichen Leben 299, 488 – 489  
 Muster unter der Lupe 104, 158, 174, 337 – 378  
 Musterkatalog 581, 583, 585  
 Muster-Zoo 604  
 MVC *siehe* Model-View-Controller

## N

Null-Objekt 214, 372

## O

Objekte erzeugen 134

Objekte umhüllen 88, 242, 252, 260, 473, 508  
*Siehe auch* Adapter-Muster, Decorator-Muster,  
 Facade-Muster, Proxy-Muster

Objekterstellung kapseln 114, 136

Objektkonstruktion kapseln 614 – 615

objektorientierte Entwurfsprinzipien 9, 30 – 31  
 auf eine Schnittstelle programmieren, nicht auf eine  
 Implementierung 11, 243, 335

das Veränderliche kapseln 9, 111

die Komposition der Vererbung vorziehen 23, 243,  
 397

eine Klasse, eine Verantwortlichkeit 185, 336, 339,  
 367

Hollywood-Prinzip 296

nach lose gebundenen Entwürfen für interagierende  
 Objekte streben 53

Offen/Geschlossen-Prinzip 86 – 87, 407

Prinzip der Umkehrung der Abhängigkeiten 139 143

Prinzip der Verschwiegenheit 265

Observable 64, 71

Observer-Muster

Code-Magnet 69

Definition 51 – 52

Einführung 44

Gespräch am Arbeitsplatz 55

Herausgeben (push) 63

Herausziehen (pull) 63

in Swing 72 – 73

Java-Unterstützung 64

Kamingespräch 62

Klassendiagramm 52

Kurzdrama 48

zusammengesetzte Muster 516

Offen/Geschlossen-Prinzip 86 – 87

Oreo-Keks 526

## P

Partnervermittlung in Objekthausen 475

Pfannkuchenhaus Objekthausen 316, 628

Pizzeria 112

Portland Pattern Repository 603

Prinzip der einen Verantwortlichkeit 339  
*Siehe auch* objektorientierte Entwurfsprinzipien:  
 eine Klasse, eine Verantwortlichkeit

Prinzip der Umkehrung der Abhängigkeiten 139 – 143  
 und Hollywood-Prinzip 298

Prinzip der Verschwiegenheit 265 – 268  
 Nachteile 267

Programmieren auf Implementierungen 12, 17, 71

Programmieren auf Schnittstellen 12

Programmieren auf Schnittstellen, nicht auf Implemen-  
 tierungen 11, 75

Prototyp-Muster 626 – 627

Proxy-Muster

Cache-Proxy 471

Code-Fertiggericht 495

Definition 460

dynamischer Proxy 474, 479, 486  
 und RMI 486

java.lang.reflect.Proxy 474

Kamingespräch 472 – 473

Klassendiagramm 461

Proxy-Zoo 488 – 489

## Q – Z

Remote-Proxy 434  
Schutzproxy 474, 477  
Übung 482  
und Adapter-Muster 471  
und Decorator-Muster 471 – 473  
Varianten 471  
virtueller Proxy 462  
    Image-Proxy 464  
Punkt für Punkt 32, 74, 105, 162, 186, 230, 270, 311,  
    380, 423, 491, 560, 608

## Q

Qualität *siehe* Qualität ohne Namen  
Qualität ohne Namen *siehe* Qualität

## R

Refactoring 354, 359  
Remote Method Invocation *siehe* RMI  
Remote-Proxy 434  
    *Siehe auch* Proxy-Muster  
Restaurant Objekthausen 26, 197, 316, 628  
RMI 436  
Rückgängig 216, 227

## S

Schnittstelle 12  
Schok-O-Holika 175  
SimEnte 2, 500  
Singleton-Muster  
    Definition 177  
    doppelt geprüftes Sperren 182  
    Interview 174  
    Klassendiagramm 177

Nachteile 184  
    und Garbage Collection 184  
    und globale Variablen 185  
    und Multithreading 180 – 182  
Unter der Lupe 173  
Vorteile 170, 184

Skeleton 440

Spitzen Sie Ihren Bleistift 5, 42, 54, 61, 94, 97, 99, 124,  
    137, 148, 176, 183, 205, 225, 242, 268, 284, 322,  
    342, 396, 400, 406, 409, 421, 483, 511, 518, 520,  
    589

State-Muster

    Definition 410  
    Einführung 398  
    Klassendiagramm 410  
    Nachteile 412, 417  
    und Strategy-Muster 411, 418 – 419  
    Zustände teilen 412

statische Fabrik 115

Sternback-Kaffee 80, 276

Strategy-Muster 24

    Familie von Algorithmen 22  
    Kamingespräch 308  
    und State-Muster 411, 418 – 419  
    und Template Method-Muster 308 – 309  
    Verhalten kapseln 22

Strukturmuster 605

Stub 440

## T

Teil/Ganzes-Hierarchie 356  
    *Siehe auch* Composite-Muster

Template Method-Muster

    Definition 289  
    Einführung 286

Hook 292, 295  
 Kamingespräch 308 – 309  
 Klassendiagramm 289  
 und Applet 307  
 und Hollywood-Prinzip 297  
 und java.util.Arrays 300  
 und Strategy-Muster 305, 308 – 309  
 und Swing 306  
 Unter der Lupe 290 – 291  
 Vorteile 288  
 Sokratisches Gespräch 171

## U

Umkehrung 141 – 142

## V

Veränderliches *siehe* kapseln, was veränderlich ist  
 Veränderung 339  
   identifizieren 53  
   Konstante der Software-Entwicklung 8  
   vorhersagen 14  
 Vererbung  
   Nachteile 5  
   vs. Komposition 93  
   Wiederverwendbarkeit 5 – 6  
 Verhalten kapseln 11  
 Vermittler 237  
 verzögerte Initialisierung 177

Viererbände (Gang of Four, GoF) 583, 601  
   Gamma, Erich 601  
   Helm, Richard 601  
   Johnson, Ralph 601  
   Vlissides, John 601  
 Visitor-Muster 628 – 629  
 Vlissides, John 601

## W

wann man keine Muster verwenden soll 596 – 598  
 Warum Enten? 500  
 Wer macht was? 202, 254, 298, 379, 422, 487, 588  
 Wetter-O-Rama 38  
 Wiederverwendung 13, 23, 85  
 Wunderwaffe 594

## Z

Zugriffskontrolle 460  
   *Siehe auch* Proxy-Muster  
 zusammengesetzte Muster 500  
   500, 522  
 Abstract Factory-Muster 508  
 Adapter-Muster 504  
 Composite-Muster 513  
 Decorator-Muster 506  
 Klassendiagramm 524  
 Observer-Muster 516  
 Zustände kapseln 399  
 Zustandsautomaten 388 – 389

# ✦ **Kolophon** ✦



**Alle Layouts im Innenteil wurden von Eric Freeman, Elisabeth Freeman, Kathy Sierra und Bert Bates gestaltet.** Von Kathy und Bert stammt das »Look and Feel« der amerikanischen Head First-Reihe. Das Buch wurde mit Adobe InDesign CS (einem unglaublich coolen Design-Tool, von dem wir gar nicht genug bekommen können) und Adobe Photoshop CS hergestellt. Für den Satz wurden die Schriftarten Uncle Stinky, Mister Frisky (nein, das ist *kein* Witz), Ann Satellite, Arial, Baskerville, Comic Sans, Corndog, Futura, Marker Felt, Myriad Pro, Residoo, Skippy Sharp, Savoye LET, Jokerman LET, Courier New und Woodrow benutzt.

Der Innenteil wurde ausschließlich auf Apple Macintosh-Rechnern gestaltet und hergestellt – bei Head First legen wir alle viel Wert auf »Think Different«. Der gesamte Original-Java-Code wurde mit James Goslings Lieblings-IDE *vi* erstellt, aber wir sollten wirklich einmal Erich Gammas Eclipse ausprobieren.

Unsere Energielieferanten an den langen, mit Schreiben ausgefüllten Tagen waren der Koffein-Kraftstoff von Honest Tea und Tejava, die reine Luft von Santa Fe und die großartigen Klänge von Banco de Gaia, Cocteau Twins, Buddha Bar I–VI, Delerium, Enigma, Mike Oldfield, Olive, Orb, Orbital, LTJ Bukem, Massive Attack, Steve Roach, Sasha and Digweed, Thievery Corporation, Zero 7 und Neil Finn (in all seinen Inkarnationen), außerdem Unmengen von Acid Trance und mehr Musik aus den 80ern, als Sie so genau wissen möchten.