

O'REILLY®

# Entwurfsmuster

von Kopf bis Fuß

Mit Design Patterns  
flexible objekt-  
orientierte Software  
erstellen

---

Eric Freeman &  
Elisabeth Robson

mit Kathy Sierra & Bert Bates

Übersetzung von Jørgen W. Lang



Übersetzung der  
2. englischen  
Auflage



Ein gehirnfrendliches Buch

# Entwurfsmuster von Kopf bis Fuß

Wäre es nicht wundervoll,  
wenn es ein Buch über Entwurfs-  
muster gäbe, das mehr Spaß macht  
als ein Besuch beim Zahnarzt  
und aufschlussreicher ist als ein  
Steuerformular? Ist wohl nur  
ein Traum ...



Eric Freeman  
Elisabeth Robson

mit  
Kathy Sierra  
Bert Bates

Deutsche Übersetzung von  
Jørgen W. Lang

**O'REILLY®**

Eric Freeman, Elisabeth Robson, Kathy Sierra und Bert Bates

Lektorat: Alexandra Follenius

Übersetzung: Jørgen W. Lang

Korrektur: Sibylle Feldmann, [www.richtiger-text.de](http://www.richtiger-text.de)

Satz: Ulrich Borstelmann, [www.borstelmann.de](http://www.borstelmann.de)

Herstellung: Stefanie Weidner

Umschlaggestaltung: Ellie Volckhausen, Michael Oréal, [www.oreal.de](http://www.oreal.de)

Druck und Bindung: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;

detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-96009-162-2

PDF 978-3-96010-503-9

ePub 978-3-96010-504-6

mobi 978-3-96010-505-3

3. Auflage 2022, Übersetzung der 2. englischen Auflage

Translation Copyright für die deutschsprachige Ausgabe © 2021 by dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Authorized German translation of the English edition of *Head First Design Patterns*, ISBN 978-1-492-07800-5 © 2021 Eric Freeman and Elisabeth Robson. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Dieses Buch erscheint in Kooperation mit O'Reilly Media, Inc. unter dem Imprint »O'REILLY«.

O'REILLY ist ein Markenzeichen und eine eingetragene Marke von O'Reilly Media, Inc. und wird mit Einwilligung des Eigentümers verwendet.

Bei der Erstellung dieses Buchs wurden keinerlei Enten oder Truthähne verletzt.

Die Mitglieder der »Gang of Four« haben uns die Erlaubnis erteilt, ihre Fotos in diesem Buch abzdrukken. Doch, sie sehen wirklich so gut aus.

*Hinweis:*

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt.

Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.



*Schreiben Sie uns:*

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: [komentar@oreilly.de](mailto:komentar@oreilly.de).

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag noch Übersetzer können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Papier  
plus<sup>+</sup>  
PDF.

Zu diesem Buch – sowie zu vielen weiteren O'Reilly-Büchern – können Sie auch das entsprechende E-Book im PDF-Format herunterladen. Werden Sie dazu einfach Mitglied bei [oreilly.plus](http://oreilly.plus)<sup>+</sup>:

[www.oreilly.plus](http://www.oreilly.plus)

Für die Gang of Four, deren Fachwissen und deren Erfahrung beim Beschreiben und Erklären von Entwurfsmustern die Welt des Softwaredesigns für immer verändert und das Leben von Entwicklern weltweit verbessert haben.

Aber mal im Ernst – *wann* kommt die zweite Auflage eures Buchs?  
Schließlich ist es erst ~~zehn~~ *fünfundzwanzig* Jahre her!

## Die Autor:innen von »Entwurfsmuster von Kopf bis Fuß«

↙ Elisabeth Robson



↙ Eric Freeman



**Eric** wurde von seiner Mitstreiterin bei der Entwicklung der Von-Kopf-bis-Fuß-Reihe, Kathy Sierra, bezeichnet als »eines dieser seltenen Individuen, die fließend die Sprache, Praxis und Kultur verschiedener Lebenswelten wie die der Hipster-Hacker, Unternehmensdirektoren, Ingenieure und Think-Tanks beherrschen«.

Von Haus aus ist Eric Informatiker, seinen Dokortitel erlangte er an der Yale University. Beruflich war er als CTO von Disney Online & Disney.com bei der Walt Disney Company tätig.

Aktuell ist er Kodirektor der Head-First-Reihe, die in Deutschland als Von-Kopf-bis-Fuß-Reihe bekannt ist. Bei der von ihm mitgegründeten Online-Lernplattform WickedlySmart widmet er seine Zeit der Erstellung von Print- und Video-Inhalten, die über die führenden Bildungskanäle vertrieben werden.

Eric's Bücher in der Von-Kopf-bis-Fuß-Reihe Bücher umfassen die Themen Entwurfsmuster, HTML & CSS, JavaScript- und HTML5-Programmierung sowie das Programmierenlernen.

Eric lebt in Austin, Texas.

**Elisabeth** ist Softwareentwicklerin, Autorin und IT-Trainerin. Seit ihren Studententagen an der Yale University, wo sie ihren Master in Informatik erworben hat, ist sie mit Leib und Seele Informatikerin.

Aktuell ist sie Mitbegründerin von WickedlySmart, wo sie Bücher, Artikel, Videos und mehr erstellt. Zuvor hat sie als Director of Special Projects bei O'Reilly Media Präsenz-Workshops und Onlinekurse zu einer Vielzahl von technischen Themen gestaltet und eine Leidenschaft für das Entwerfen von Lehrsystemen entwickelt, die Menschen helfen, Technologien zu verstehen.

Wenn sie nicht vor ihrem Computer sitzt, geht Elisabeth, stets die Kamera in Griffweite, in die freie Natur, um zu wandern oder Rad oder Kajak zu fahren, oder sie widmet sich der Gartenarbeit.

## Die Köpfe hinter dieser Reihe (und Mitverschwörer:innen bei diesem Buch)

Kathy Sierra



Bert Bates



**Kathy** interessiert sich für Lerntheorie seit ihrer Zeit als Spieleentwicklerin für Virgin, MGM und Amblin' und als Dozentin für New Media Authoring an der UCLA. Sie war Master-Java-Trainerin für Sun Microsystems, und sie gründete JavaRanch.com (jetzt CodeRanch.com), das in den Jahren 2003 und 2004 den Jolt Cola Productivity Award gewann.

Im Jahr 2015 erhielt sie den Electronic Frontier Foundation's Pioneer Award für ihre Arbeit zur Schaffung kompetenter Nutzer und den Aufbau nachhaltiger Gemeinschaften.

In jüngster Zeit konzentriert sich Kathy auf modernste Bewegungswissenschaften und das Coaching zum Erwerb von Fähigkeiten, bekannt als Ecological Dynamics oder »Eco-D«. Ihre Arbeit, bei der sie Eco-D für das Training von Pferden einsetzt, führt zu einem weitaus humaneren Ansatz in der Reitkunst, was die einen erfreut (und die anderen, traurigerweise, verwirrt). Die Pferde, die das Glück haben, dass ihre Besitzer Kathys Ansatz anwenden, sind zufriedener, autonomer, gesünder und sportlicher als ihre traditionell trainierten Artgenossen.

Sie können Kathy Sierra auf Instagram folgen: [@pantherflows](https://www.instagram.com/pantherflows).

Bevor **Bert** Autor wurde, war er Entwickler, spezialisiert auf KI der alten Schule (hauptsächlich Expertensysteme), Echtzeit-Betriebssysteme und komplexe Planungssysteme.

Im Jahr 2003 schrieben Bert und Kathy Head First Java und starteten die Head-First-Reihe. Seitdem hat er weitere Java-Bücher geschrieben und Sun Microsystems und Oracle bei vielen ihrer Java-Zertifizierungen beraten. Außerdem hat er Hunderte von Autor:innen und Lektor:innen darin geschult, Bücher zu schreiben, die gute Lerninhalte bieten.

Bert ist Go-Spieler und hat 2016 mit Entsetzen und Faszination zugehört, wie AlphaGo Lee Sedol besiegt hat. In letzter Zeit nutzt er Eco-D (Ecological Dynamics), um sein Golfspiel zu verbessern und seinen Papagei Bokch zu trainieren.

Bert und Kathy haben das Privileg, Beth und Eric seit 16 Jahren zu kennen, und die Head-First-Reihe hat das große Glück, sie zu den wichtigsten Mitwirkenden zu zählen.

Sie können Bert unter [CodeRanch.com](https://www.coderanch.com) eine Nachricht schicken.

## Über den Übersetzer dieses Buchs

**Jørgen W. Lang** lebt und arbeitet als freier Autor (»CSS Kochbuch«) und Übersetzer in Oldenburg/Niedersachsen. Mitte der Neunzigerjahre des vergangenen Jahrhunderts begann er, sich mit dem damals noch jungen World Wide Web und seinen Möglichkeiten zu beschäftigen. Pünktlich zum Jahrtausendwendejahr erschien seine erste Übersetzung für den O'Reilly Verlag. Mittlerweile ist der Umfang auf mehr als 10.000 Seiten angewachsen.

Mit großer Energie und Ausdauer bringt Jørgen seit fast schon zwei Jahrzehnten Webseiten bei, das zu tun, was von ihnen erwartet wird – unabhängig davon, auf welchem Gerät sie betrachtet werden (elektrische Zahnbürsten ausgenommen).

Das zweite Standbein von Jørgen Lang ist die Musik. Außerhalb der Welt der semantischen Elemente, Selektoren und Objekte hat er sich einen Namen als hervorragender Gitarrist, Sänger, Komponist und Arrangeur gemacht und kann auf eine Vielzahl veröffentlichter Alben und mehrere Hundert Konzerte in aller Welt (z. B. für die UNESCO in Seoul) zurückblicken.



## Zur deutschen Übersetzung

Eine Herausforderung bei der Übersetzung von »Head First Design Patterns« war, möglichst alle verwendeten Bilder und Metaphern ins Deutsche zu übertragen, bei den Fachausdrücken aber auf in der Praxis ungebräuchliche Übersetzungen zu verzichten. So werden wir im Buch zwar Fabriken bauen, aber unser Entwurfsmuster heißt Factory, wir werden Beobachter entsenden, das Entwurfsmuster aber Observer nennen. Wir hoffen, dass unsere deutschen Leserinnen und Leser die Entwurfsmuster so leichter vor Augen haben, aber dennoch wissen, wie der Profi diese Muster in der Praxis bezeichnet.

# Der Inhalt (im Überblick)

	Einführung	xxiii
1	Willkommen bei den Entwurfsmustern: <i>Einführung in Entwurfsmuster</i>	1
2	Ihre Objekte auf dem Laufenden halten: <i>Das Observer-Muster</i>	37
3	Objekte dekorieren: <i>Das Decorator-Muster</i>	79
4	In der OO-Bäckerei ...: <i>Das Factory-Muster</i>	109
5	Einmalige Objekte: <i>Das Singleton-Muster</i>	169
6	Aufrufe verkapseln: <i>Das Command-Muster</i>	191
7	Anpassungsfähigkeit beweisen: <i>Die Adapter- und Facade-Muster</i>	237
8	Algorithmen verkapseln: <i>Das Template Method-Muster</i>	277
9	Erfolgreiche Collections: <i>Die Iterator- und Composite-Muster</i>	317
10	Der (Zu-)Stand der Dinge: <i>Das State-Muster</i>	381
11	Objektzugriff kontrollieren: <i>Das Proxy-Muster</i>	425
12	Muster von Mustern: <i>Zusammengesetzte Muster</i>	493
13	Muster in der wahren Welt: <i>Schöner leben mit Mustern</i>	563
14	Anhang: <i>Übrig gebliebene Muster</i>	597

# Inhalt (jetzt ausführlich)

## Einführung

**Ihr mustergültiges Gehirn.** Sie versuchen, etwas zu lernen, und Ihr Hirn tut sein Bestes, damit das Gelernte nicht hängen bleibt. Es denkt nämlich: »Wir sollten lieber ordentlich Platz für wichtigere Dinge lassen, z. B. für das Wissen, welche Tiere einem gefährlich werden könnten, oder dass es eine ganz schlechte Idee ist, nackt Snowboard zu fahren.« Tja, wie schaffen wir es nun, Ihr Gehirn davon zu überzeugen, dass Ihr Leben davon abhängt, etwas über Entwurfsmuster zu wissen?

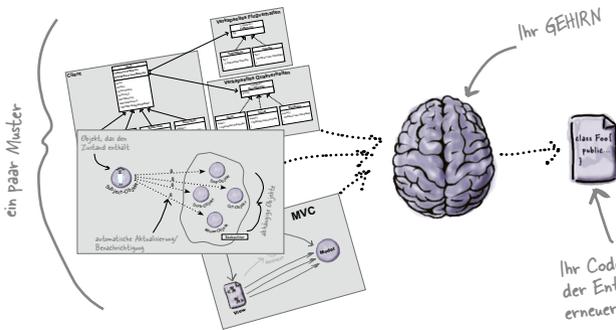
Für wen ist dieses Buch?	xxiv
Wir wissen, was Sie gerade denken	xxv
Und wir wissen, was Ihr Gehirn gerade denkt	xxv
Metakognition: Nachdenken übers Denken	xxvii
Machen Sie sich Ihr Hirn untertan	xxix
Fachgutachter	xxxii
Danksagungen	xxxiv

# 1

## Willkommen bei den Entwurfsmustern

**Irgendjemand hat Ihr Problem schon gelöst.** In diesem Kapitel lernen Sie, warum (und wie) Sie die Weisheit und die Lehren anderer Entwickler nutzen können, die die gleichen Designprobleme bereits hatten und die Reise überlebt haben. Bevor dieses Kapitel zu Ende ist, kümmern wir uns um die Verwendung und die Vorteile der Entwurfsmuster, sehen uns ein paar grundsätzliche objektorientierte (OO-)Designprinzipien an und gehen mit Ihnen zusammen ein Beispiel für die Funktionsweise von Entwurfsmustern durch. Die beste Möglichkeit, die Muster zu verwenden, ist, sie *in Ihr Gehirn zu laden* und dann die Stellen in Ihren Designs und bestehenden Programmen zu *erkennen*, an denen der Einsatz sinnvoll ist. Im Gegensatz zur Codewiederverwendung können Sie mit Entwurfsmustern die *Erfahrung* anderer Menschen wiederverwenden.

Vergiss nicht: Das Wissen um Konzepte wie Abstraktion, Vererbung und Polymorphismus macht dich noch nicht zu einer guten OO-Designerin. Eine Design-Meisterin überlegt, wie sie flexible Entwürfe erschaffen kann, die wartbar sind und mit Veränderungen umgehen können.



Es begann mit einer einfachen SimUDuck-App	2
Aber jetzt sollen die Enten FLIEGEN können	3
Aber irgendetwas ging furchtbar schief ...	4
Joe denkt über Vererbung nach ...	5
Wie wäre es mit einem Interface?	6
Was würden Sie an Joes Stelle tun?	7
Die einzige Konstante in der Softwareentwicklung	8
Das Problem eingrenzen	9
Veränderliches und Unveränderliches voneinander trennen	10
Das Entenverhalten entwerfen	11
Das Entenverhalten implementieren	13
Das Entenverhalten integrieren	15
Den Entencode testen	18
Verhalten dynamisch festlegen	20
Das große Ganze: Verkapseltes Verhalten	22
HAT-EIN ist besser als IST-EIN	23
Da wir gerade von Entwurfsmustern sprechen ...	24
Im Bistro um die Ecke aufgeschnappt ...	26
Im Büro nebenan aufgeschnappt ...	27
Die Stärke eines gemeinsamen Mustervokabulars	28
Wie setze ich Entwurfsmuster ein?	29
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	32

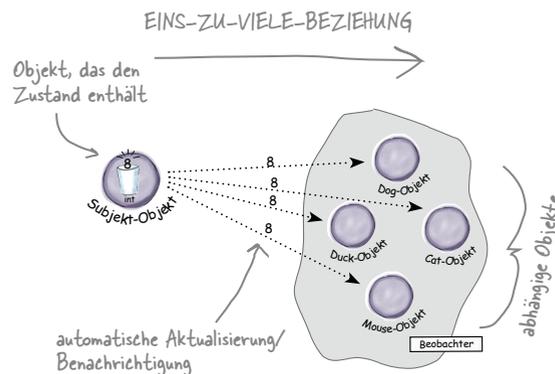
# Das Observer-Muster

## 2 Ihre Objekte auf dem Laufenden halten

Sie wollen interessante Ereignisse doch nicht verpassen, oder? Es gibt ein Muster, das unsere Objekte auf dem Laufenden hält, wenn etwas für sie Wichtiges passiert, und zwar eins der am häufigsten verwendeten und nützlichsten Entwurfsmuster überhaupt: das Observer-Muster. In diesem Kapitel sehen wir uns alle möglichen interessanten Eigenschaften dieses Musters an, wie *Eins-zu-viele-Beziehungen* und *lose Kopplungen*. Mit dem Observer-Muster im Gepäck sind Sie der Star jeder Muster-Party!



Die Wetterstation im Überblick	39
Willkommen zum Observer-Muster	44
Herausgeber + Abonnenten = Observer-Muster	45
Die Observer-Muster-Definition	51
Die Macht der losen Kopplung	54
Die Wetterstation entwerfen	57
Die Wetterstation implementieren	58
Die Wetterstation hochfahren	61
Das Observer-Muster in freier Wildbahn	65
Die lebensverändernde Applikation programmieren	66
Inzwischen bei Weather-O-Rama	69
Probefahrt für den neuen Code	71
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	72
Die Entwurfsprinzipien-Challenge	73

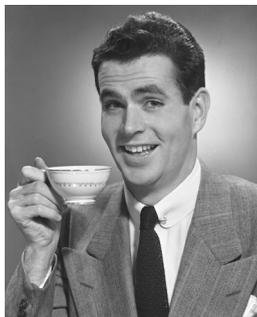


## Das Decorator-Muster

# 3 Objekte dekorieren

**Nennen wir dieses Kapitel einfach »Vererbst du noch, oder designst du schon?«.** Hier werfen wir einen weiteren Blick auf das typische Überstrapazieren von Vererbung. Sie werden lernen, wie Sie Ihre Klassen zur Laufzeit mit einer Form der Objektkomposition dekorieren können. Warum? Sobald Sie die Dekoration beherrschen, können Sie Ihren Objekten (oder denen anderer Leute) neue Verantwortung geben, *ohne hierfür den Code der zugrunde liegenden Klassen ändern zu müssen.*

Ich dachte immer, echte Männer benutzen grundsätzlich Subklassen. Bis ich die Macht der Erweiterung zur Laufzeit anstatt während der Kompilierung kennenlernte. Und jetzt sehen Sie mich an!



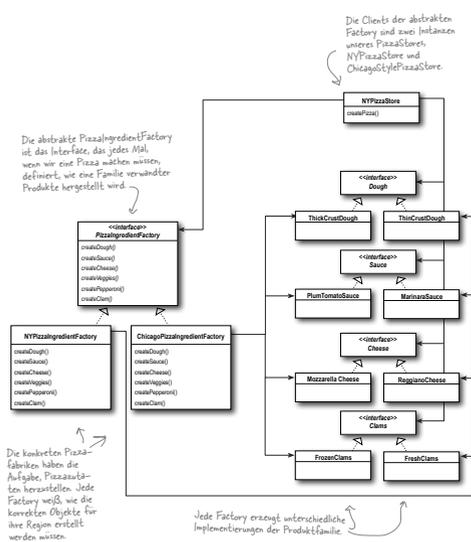
Willkommen bei Starbuzz Coffee	80
Das Offen/Geschlossen-Prinzip	86
Wir stellen vor: das Decorator-Muster	88
Eine Getränkebestellung mit Decoratoren aufbauen	89
Die Definition des Decorator-Musters	91
Unsere Getränke dekorieren	92
Den Starbuzz-Code schreiben	95
Getränke programmieren	96
Zutaten programmieren	97
Den Kaffee servieren	98
Decoratoren in freier Wildbahn: Java I/O	100
Die java.io-Klassen dekorieren	101
Einen eigenen Java-I/O-Decorator schreiben	102
Unseren neuen Java-I/O-Decorator testen	103
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	105

# Das Factory-Muster

## 4

### In der OO-Bäckerei ...

Machen Sie sich bereit, ein paar lose gekoppelte OO-Entwürfe zu backen. Zur Erstellung von Objekten gehört mehr, als einfach den `new`-Operator einzusetzen. Sie werden lernen, dass Instanziierung nicht in der Öffentlichkeit durchgeführt werden sollte und oft zu *Kopplungsproblemen* führen kann. Und *das* wollen wir nun wirklich nicht, oder? Finden Sie heraus, wie das Factory-Muster Sie vor peinlichen Abhängigkeiten bewahren können.



Das Veränderliche finden	112
Die Objekterstellung verkapseln	114
Eine einfache Pizzafabrik erstellen	115
Die einfache Fabrik definieren	117
Ein Framework für die Pizzeria	120
Den Subklassen die Entscheidung überlassen	121
Eine Fabrikmethode deklarieren	125
Jetzt ist es endlich Zeit, das Factory Method-Muster kennenzulernen	131
Ein paralleler Blick auf Hersteller und Produkte	132
Die Definition des Factory Method-Musters	134
Ein Blick auf Objektabhängigkeiten	138
Das Prinzip der Umkehrung der Abhängigkeiten	139
Das Prinzip anwenden	140
Zutatenfamilien	145
Die Zutatenfabriken bauen	146
Die Pizzas überarbeiten ...	149
Unsere Pizzerien überarbeiten	152
Was haben wir getan?	153
Die Definition des Abstract Factory-Musters	156
Factory Method und Abstract Factory im Vergleich	160
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	162

## Das Singleton-Muster

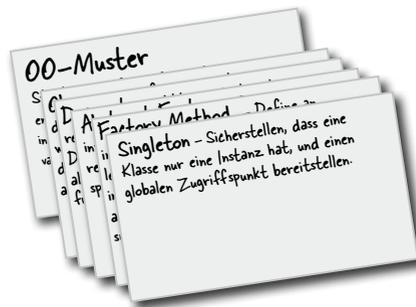
# 5

### Einmalige Objekte

Unser nächster Halt ist das Singleton-Muster, unsere Fahrkarte für die Erstellung einmaliger Objekte, von denen es immer nur eine Instanz gibt. Es wird Sie freuen, zu erfahren, dass das Singleton-Muster, bezogen auf sein Klassendiagramm, das einfachste Muster von allen ist. Tatsächlich enthält es nur eine einzige Klasse! Aber machen Sie es sich nicht zu bequem. Trotz des einfachen Klassendesigns müssen wir für seine Implementierung einige tiefgehende objektorientierte Überlegungen anstellen. Also, setzen Sie Ihre Denkmütze auf, und los geht's.



Die Implementierung des klassischen Singleton-Musters im Detail	173
Die Schokoladenfabrik	175
Definition des Singleton-Musters	177
Brüssel Houston, wir haben ein Problem ...	178
Mit Multithreading umgehen	180
Können wir das Multithreading verbessern?	181
Inzwischen in der Schokoladenfabrik ...	183
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	186



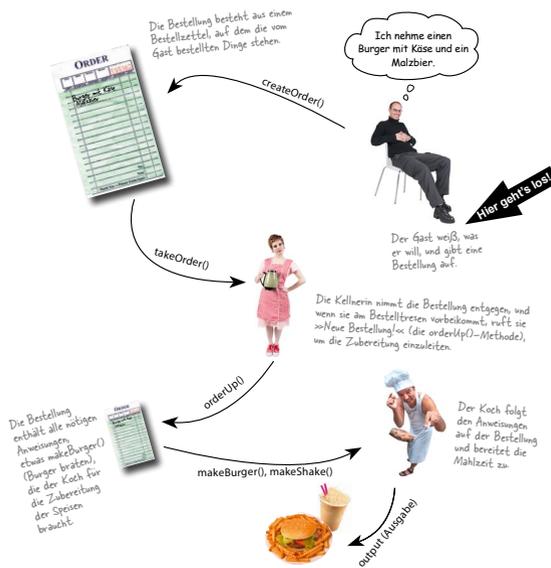
# Das Command-Muster

## 6

### Aufrufe verkapseln

In diesem Kapitel heben wir die Verkapselung auf ein ganz neues Niveau: Wir werden **Methodenaufrufe verkapseln**.

Ja, richtig gehört: Methodenaufrufe. Damit können wir Teile von Berechnungen »einfrieren«, wodurch sich das aufrufende Objekt nicht um die Details der Berechnung kümmern muss. Es nutzt einfach die eingefrorene Methode für die Erfüllung seiner Aufgabe. Mit diesen verkapselten Methodenaufrufen sind aber noch ganz andere schlaue Dinge möglich. Wir können sie beispielsweise zur Protokollierung nutzen oder sie wiederverwenden, um eine »Rückgängig«-Funktionalität zu implementieren.



Ein Blick auf die Drittherstellerklassen	194
Inzwischen im Bistro ...	197
Vom Bistro zum Command-Muster	201
Unser erstes Command-Objekt	203
Das Command-Objekt verwenden	204
Befehle den Plätzen zuweisen	209
Die Fernsteuerung implementieren	210
Die Befehle implementieren	211
Die Fernsteuerung auf Herz und Nieren testen	212
Zeit für die Dokumentation ...	215
Was machen wir hier?	217
Zeit, den Rückgängig-Knopf auf seine Qualität zu testen!	220
Zustände für die Implementierung der »Rückgängig«-Funktion verwenden	221
Die Deckenventilator-Befehle mit einer »Rückgängig«-Funktion versehen	222
Jede Fernsteuerung braucht einen Partymodus!	225
Einen Makro-Befehl benutzen	226
Viele Verwendungsmöglichkeiten für das Command-Muster: Warteschlangen für Befehle	229
Weitere Anwendungen des Command-Musters: Aufträge protokollieren	230
Das Command-Muster in der wahren Welt	231
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	233

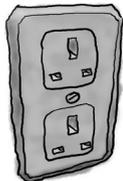
# Die Adapter- und Facade-Muster

## 7

### Anpassungsfähigkeit beweisen

In diesem Kapitel versuchen wir so unmögliche Dinge wie die **Quadratur des Kreises**. Klingt ausgeschlossen? Aber nicht mit Entwurfsmustern. Erinnern Sie sich noch an das Decorator-Muster? Wir haben Objekte verpackt, um sie mit Verantwortlichkeiten zu versehen. Diesmal **verpacken wir Objekte**, damit ihre Schnittstellen wie etwas aussehen, das sie nicht sind. So können wir Designs, die bestimmte Schnittstellen erwarten, an Klassen anpassen, die eine andere Schnittstelle implementieren. Und das ist noch nicht alles. Wenn wir schon dabei sind, sehen wir uns gleich noch ein anderes Muster an, das Objekte verpackt, um ihre Schnittstelle zu vereinfachen.

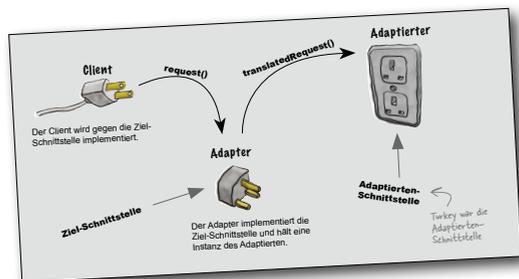
Steckdose in Großbritannien



Stromadapter



US-Standardstecker



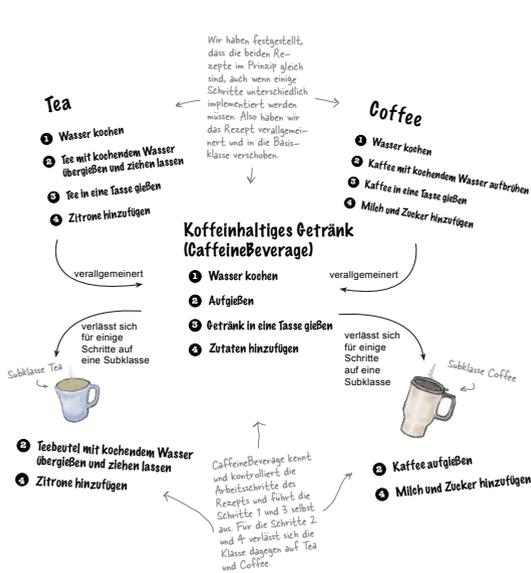
Überall Adapter	238
Objektorientierte Adapter	239
Probelauf für den Adapter	242
Das Adapter-Muster erklärt	243
Die Definition des Adapter-Musters	245
Objekt- und Klassen-Adapter	246
Adapter im echten Leben	250
Einen Enumerator an einen Iterator anpassen	251
Unser eigenes Heimkino	257
Einen Film ansehen (auf die harte Tour)	258
Licht, Kamera, Facade!	260
Die Heimkino-Facade konstruieren	263
Die vereinfachte Schnittstelle implementieren	264
Einen Film anschauen (auf die sanfte Tour)	265
Die Definition des Facade-Musters	266
Das Prinzip der Verschwiegenheit	267
Wie man KEINE Freunde gewinnt und KEINE Objekte beeinflusst	268
Das Facade-Muster und das Prinzip der Verschwiegenheit	271
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	272

# Das Template Method-Muster

## 8

### Algorithmen verkapseln

**Wir sind auf dem Verkapselungstrip. Was haben wir schon alles verkapselt? Objekterstellung, Methodenaufrufe, komplexe Schnittstellen, Enten, Pizzas – was kommt wohl als Nächstes?** In diesem Kapitel gehen wir der Verkapselung von Algorithmenteilen auf den Grund, damit Subklassen sich bei Bedarf jederzeit direkt in eine Berechnung einklinken können. Außerdem lernen wir ein Entwurfsprinzip kennen, das von Hollywood inspiriert ist. Na dann mal los ...



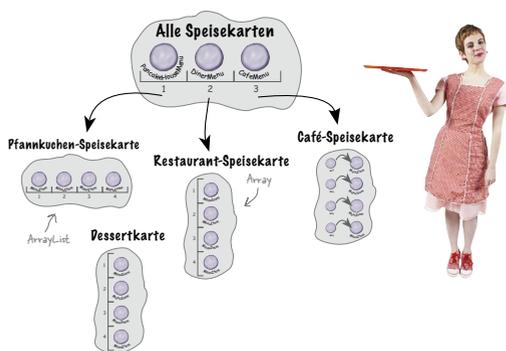
Zeit für mehr Koffein	278
Ein paar Kaffee- und Tee-Klassen zusammenrühren (in Java)	279
Kaffee und Tee abstrahieren	282
Das Design verbessern	283
prepareRecipe() abstrahieren	284
Was haben wir getan?	287
Willkommen beim Template Method-Muster	288
Was hat uns das Template Method-Muster gebracht?	290
Die Definition des Template Method-Musters	291
Eingehängt in die Template Method ...	294
Den Hook verwenden	295
Das Hollywood-Prinzip und das Template Method-Muster	299
Template-Methoden in freier Wildbahn	301
Mit dem Template Method-Muster sortieren	302
Was ist compareTo()?	303
Enten mit Enten vergleichen	304
Ein paar Enten sortieren	305
Das Making-of der Entensortiermaschine	306
Swinging mit Frames	308
Eigene Listen mit AbstractList	309
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	313

# Die Iterator- und Composite-Muster

## 9

### Erfolgreiche Collections

Es gibt viele Möglichkeiten, Objekte in einer Collection zu speichern. Zum Beispiel in einem Array, einem Stack, einer Liste oder einer HashMap – Sie haben die Wahl. Dabei hat jede Form ihre Vor- und Nachteile. Irgendwann werden Ihre Clients allerdings über diese Objekte iterieren wollen. Und wollen Sie ihnen dann Ihre Implementierung offenbaren? Hoffentlich nicht! Das wäre einfach nicht professionell. Aber keine Sorge. Ihre Karriere ist nicht gefährdet. In diesem Kapitel werden Sie sehen, wie Ihre Clients über Ihre Objekte iterieren können, ohne zu wissen, wie sie gespeichert sind. Außerdem lernen Sie, wie man Super Collections von Objekten erstellt, die mit einem einzigen Satz einige eindrucksvolle Datenstrukturen überspringen können. Und als wäre das noch nicht genug, werden Sie auch noch das eine oder andere über Objektverantwortlichkeit lernen.



Große Neuigkeiten: Das Restaurant und das Pfannkuchenhaus von Objectville fusionieren	318
Sehen wir uns die Gerichte an	319
Die Spezifikation implementieren: unser erster Versuch	323
Können wir die Iteration verkapseln?	325
Willkommen zum Iterator-Muster	327
DinerMenu mit einem Iterator versehen	328
Die Restaurant-Speisekarte mit einem Iterator überarbeiten	329
Den Kellnerin-Code aufmöbeln	330
Den Code testen	331
Unser aktueller Entwurf auf dem Prüfstand ...	333
Aufräumen mit java.util.Iterator	335
Die Definition des Iterator-Musters	338
Die Struktur des Iterator-Musters	339
Das Prinzip der einzelnen Verantwortlichkeit	340
Willkommen zu Javas Iterable-Interface	343
Javas erweiterte for-Schleife	344
Ein Blick auf die Speisekarte des Cafés	347
Iteratoren und Collections	353
Die Definition des Composite-Musters	360
Speisekarten mit dem Composite-Muster entwerfen	363
MenuComponent implementieren	364
Das Gericht (MenuItem) implementieren	365
Die Komposita-Speisekarte implementieren	366
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	376

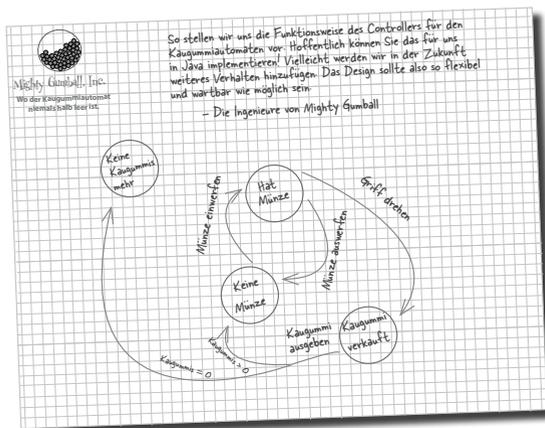
## Das State-Muster

# 10

### Der (Zu-)Stand der Dinge

**Eine kaum bekannte Tatsache: Die Strategy- und das State-Muster sind Zwillinge, die bei der Geburt getrennt wurden.**

Vielleicht denken Sie, dass beide ein ähnliches Leben führen. In Wirklichkeit hat Strategy jedoch ein unglaublich erfolgreiches Unternehmen rund um austauschbare Algorithmen aufgebaut, während State den vermutlich edleren Weg gewählt hat. Es hilft anderen Objekten, ihr Verhalten zu kontrollieren, indem es ihren inneren Zustand verändert. So unterschiedlich die Wege der beiden auch scheinen – hinter den Kulissen ist ihr Design fast identisch. Wie das sein kann und worum es beim State-Muster wirklich geht, werden wir herausfinden. Am Ende des Kapitels sehen wir uns dann an, welche Beziehung beide Muster tatsächlich zueinander haben.



Ein echter Java-Plombenzieher	382
Kurze Einführung in Zustandsautomaten	384
Den Code schreiben	386
Interner Testlauf	388
Sie haben es geahnt – ein Änderungswunsch!	390
ZUSTÄNDE wie bei Hempels unterm Sofa ...	392
Der neue Entwurf	394
Das Interface State und die Klassen definieren	395
Umbau des Kaugummiautomaten	398
Ein Blick auf die komplette Klasse GumballMachine ...	399
Weitere Zustände implementieren	400
Die Definition des State-Musters	406
Wir müssen uns wieder dem 1-von-10-Kaugummispiel widmen	409
Das Spiel fertigstellen	410
Demo für den CEO von Mighty Gumball, Inc.	411
Stimmt alles?	413
Das haben wir fast vergessen!	416
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	419



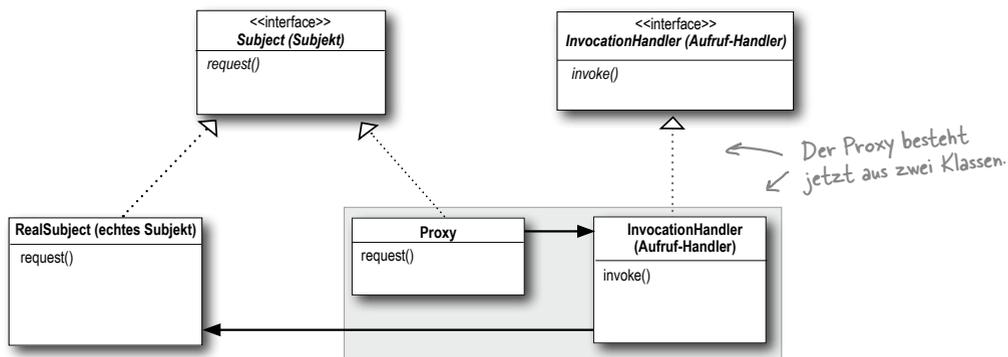
# 11 Objektzugriff kontrollieren

## Haben Sie schon einmal »guter Bulle, böser Bulle« gespielt?

Sie sind der gute Polizist und stellen alle Ihre Dienste auf eine nette und freundliche Weise bereit. Wenn Sie aber nicht wollen, dass jeder ungefragt Ihre Dienste nutzt, übernimmt der böse Polizist die Zugangskontrolle für Sie. Denn genau das tun Proxies (»Stellvertreter«): Sie kontrollieren und verwalten den Zugriff. Wie Sie sehen werden, gibt es viele Möglichkeiten, Proxies als Vertreter für andere Objekte zu nutzen. Proxies sind dafür bekannt, dass sie für die von ihnen vertretenen Objekte komplette Methodenaufrufe über das Internet abwickeln. Außerdem nehmen sie bekanntermaßen den Platz einiger ziemlich fauler Objekte ein.



Den Überwachungscode schreiben	427
Den Überwachungscode testen	428
Einführung in entfernte Methodenaufrufe	433
Den Kaugummiautomaten (GumballMachine) als entfernten Dienst einrichten	446
Bei der RMI-Registry anmelden ...	448
Die Definition des Proxy-Musters	455
Bereitmachen für den virtuellen Proxy	457
Den virtuellen Proxy für die Albencover entwerfen	459
Den Bild-Proxy schreiben	460
Partnervermittlung für Geeks in Objectville	470
Die Person implementieren	471
Fünf-Minuten-Drama: Subjekte schützen	473
Das große Ganze: Einen dynamischen Proxy für Person erstellen	474
Der Proxy-Zoo	482
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	485
Der Code für den Albumcover-Viewer	489

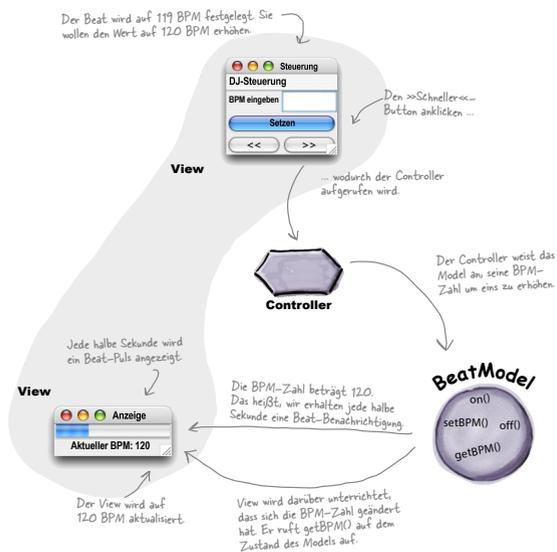


## Zusammengesetzte Muster

# 12

### Muster von Mustern

**Wer hätte gedacht, dass Muster auch zusammenarbeiten können?** Sie haben bereits die erbitterten Kamingespräche miterlebt (und dabei haben wir Ihnen die Seiten mit dem »Pattern Death Match« noch gar nicht gezeigt, die wir auf Druck des Verlegers wieder entfernen mussten). Wer hätte gedacht, dass Muster eigentlich sogar recht gut miteinander auskommen können? Ob Sie's glauben oder nicht – einige der mächtigsten OO-Entwürfe verwenden Kombinationen mehrerer Muster. Machen Sie sich bereit für die nächste Stufe Ihrer Entwurfsmuster-Fähigkeiten: zusammengesetzte Muster (»Compound-Muster«)



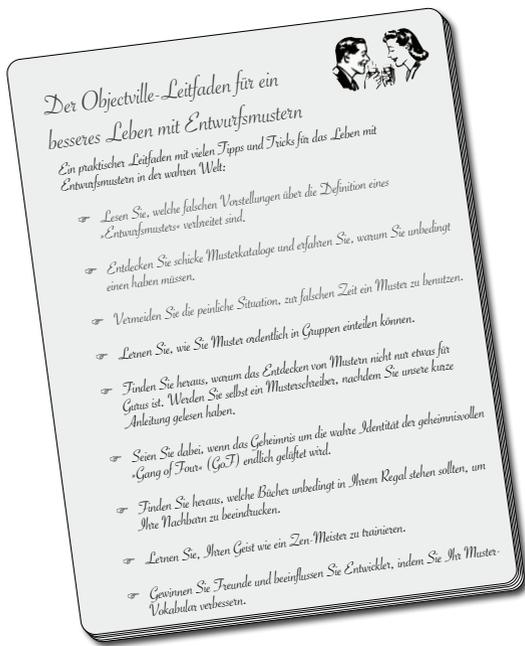
Mustergültige Zusammenarbeit	494
Ein Wiedersehen mit den Enten	495
Was haben wir getan?	517
Ein Blick aus der Vogelperspektive: das Klassendiagramm	518
Der König der zusammengesetzten Muster	520
Willkommen zu Model-View-Controller	523
Genauer hingesehen ...	524
MVC als eine Reihe von Mustern verstehen	526
MVC, um den Beat zu steuern	528
Erstellung der Einzelteile	531
Ein Blick auf die konkrete Klasse BeatModel	532
Der View	533
Den View implementieren	534
Und damit zum Controller	536
Die Einzelteile zusammensetzen ...	538
Strategy erforschen	539
Das Model adaptieren	540
Noch ein Probelauf ...	542
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	545

## Schöner leben mit Mustern

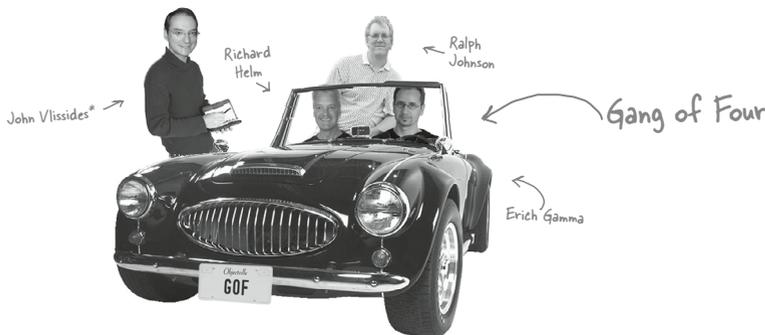
# 13

### Muster in der wahren Welt

Und damit sind Sie bereit für eine strahlende neue Welt voller Entwurfsmuster. Aber bevor Sie all die neuen Chancen nutzen, müssen wir uns noch um ein paar Details kümmern, die Ihnen draußen in der wahren Welt begegnen können, wo die Dinge etwas komplexer sind als hier in Objectville. Auf den folgenden Seiten haben wir für Sie einen kleinen Reiseführer (oder Leitfaden) vorbereitet, der Sie beim Übergang begleiten wird ...

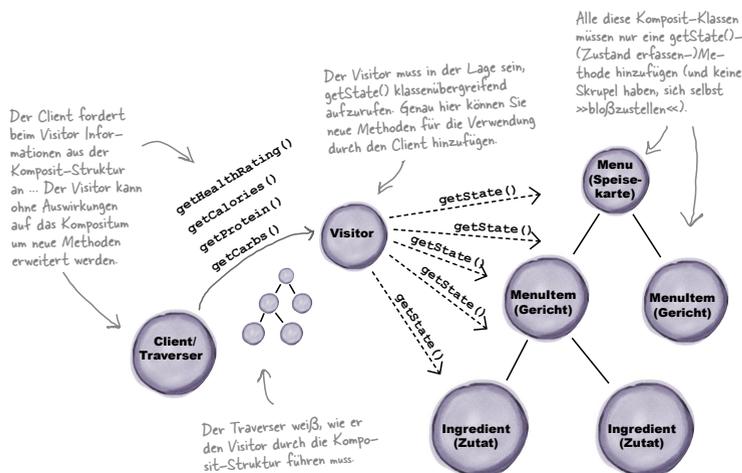


Definition von Entwurfsmustern	565
Ein genauerer Blick auf die Entwurfsmusterdefinition	567
Möge die Macht mit Ihnen sein	568
Sie wollen also eigene Entwurfsmuster schreiben	573
Entwurfsmuster ordnen	575
In Mustern denken	580
Denk-Muster	583
Vergessen Sie nicht die Macht des gemeinsamen Vokabulars	585
Spritztour durch Objectville mit der Gang of Four	587
Ihre Reise hat gerade erst begonnen!	588
Der Muster-Zoo	590
Mit Anti-Mustern das Böse auslöschen	592
Werkzeuge für Ihren Entwurfs-Werkzeugkasten	594
Abschied von Objectville ...	595



# 14 Anhang: Übrig gebliebene Muster

**Nicht jeder kann der Beliebteste sein.** Seit der Erstveröffentlichung von *Design Patterns: Elements of Reusable Object-Oriented Software* hat sich viel verändert. Entwickler haben diese Muster tausendfach verwendet. Die in diesem Anhang vorgestellten Muster sind Vollmitglieder der offiziellen GoF-Musterfamilie, werden aber nicht so oft genutzt wie die bisher gezeigten. Trotzdem sind sie auf ihre eigene Art großartig, und wenn die Situation es erfordert, können Sie sie einsetzen, ohne sich dafür zu schämen. In diesem Anhang wollen wir Ihnen einen Überblick darüber geben, worum es bei diesen Mustern geht.



Bridge	598
Builder	600
Chain of Responsibility	602
Flyweight	604
Interpreter	606
Mediator	608
Memento	610
Prototype	612
Visitor	614

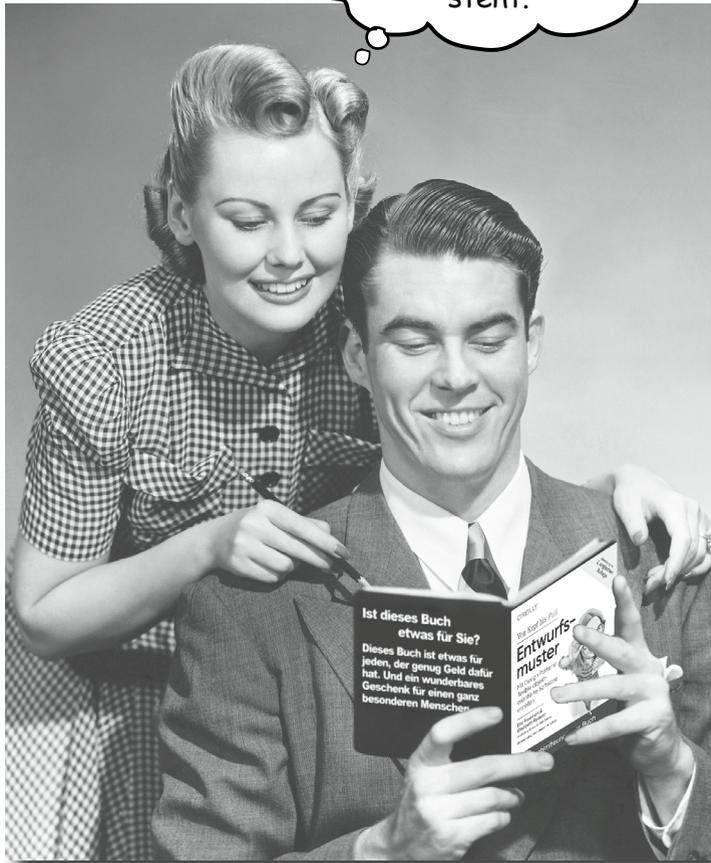
## i Index



Wie man dieses Buch benutzt

# **Einführung**

Kaum zu glauben,  
dass *so etwas* in  
einem Buch über  
Entwurfsmuster  
steht.



In diesem Abschnitt beantworten wir die brennende Frage:  
>>Warum STEHT so was in einem Buch über Entwurfsmuster?<<

## Für wen ist dieses Buch?

Wenn Sie alle diese Fragen mit »Ja« beantworten können ...

- ① Können Sie Java? (Sie müssen kein Guru sein.)
- ② Möchten Sie Entwurfsmuster **lernen, verstehen, behalten und anwenden**, einschließlich der OO-Entwurfsprinzipien, auf denen Entwurfsmuster basieren?
- ③ Ziehen Sie eine **anregende Unterhaltung beim Abendessen einer trockenen, langweiligen Vorlesung vor?**

Alle unsere Beispiele sind in Java geschrieben. Sie sollten die Grundkonzepte dieses Buchs aber auch verstehen, wenn Sie eine andere objektorientierte Sprache beherrschen.

... ist dieses Buch etwas für Sie.

## Wer sollte eher Abstand von diesem Buch nehmen?

Wenn Sie eine dieser Fragen mit »Ja« beantworten müssen ...

- ① Ist die **objektorientierte Programmierung** komplettes Neuland für Sie?
- ② Sind Sie ein Top-OO-Entwickler, der ein **Buch zum Nachschlagen** sucht?
- ③ Sind Sie ein Softwarearchitekt auf der Suche nach **Enterprise**-Entwurfsmustern?
- ④ Haben Sie **Angst, etwas Neues auszuprobieren?** Ist Ihnen eine Wurzelkanalbehandlung lieber, als Streifen kombiniert mit Karos zu tragen? Glauben Sie, dass ein Technikfachbuch, in dem Java-Komponenten vermenschlicht werden, nicht seriös sein kann?

... ist dieses Buch **nicht** das richtige für Sie.



[Anmerkung aus dem Marketing:  
Dieses Buch ist etwas für jeden,  
der eine Kreditkarte besitzt.  
Auch Barzahlung ist möglich.]

## Wir wissen, was Sie gerade denken

»Kann *das* wirklich ein seriöses Programmierlehrbuch sein?«

»Was sollen all die Abbildungen?«

»Kann ich das auf diese Weise wirklich *lernen*?«

## Und wir wissen, was Ihr Gehirn gerade denkt.

Ihr Gehirn lechzt nach Neuem. Es ist ständig dabei, Ihre Umgebung abzusuchen, und es *wartet* auf etwas Ungewöhnliches. So ist es nun einmal gebaut, und es hilft Ihnen zu überleben.

Also, was macht Ihr Gehirn mit all den gewöhnlichen, normalen Routine-sachen, denen Sie begegnen? Es tut alles in seiner Macht Stehende, damit es dadurch nicht bei seiner *eigentlichen* Arbeit gestört wird: Dinge zu erfassen, die wirklich *wichtig* sind. Es gibt sich nicht damit ab, Langweiliges zu speichern, sondern lässt dieses gar nicht erst durch den »Dies-ist-offensichtlich-nicht-wichtig«-Filter.

Woher *weiß* Ihr Gehirn denn, was wichtig ist? Nehmen Sie an, Sie machen einen Tagesausflug und ein Tiger springt vor Ihnen aus dem Gebüsch: Was passiert dabei in Ihrem Kopf und Ihrem Körper?

Neuronen feuern. Gefühle werden angekurbelt. *Chemische Substanzen durchfluten Sie.*

Und so weiß Ihr Gehirn:

### **Dies muss wichtig sein! Vergiss es nicht!**

Aber nun stellen Sie sich vor, Sie sind zu Hause oder in einer Bibliothek. In einer sicheren, warmen, tigerfreien Zone. Sie lernen. Bereiten sich auf eine Prüfung vor. Oder Sie versuchen, sich in irgendein schwieriges Thema einzuarbeiten, von dem Ihr Chef glaubt, Sie bräuchten dafür eine Woche oder höchstens zehn Tage.

Da ist nur ein Problem: Ihr Gehirn versucht, Ihnen einen großen Gefallen zu tun. Es versucht, dafür zu sorgen, dass diese *offensichtlich* unwichtigen Inhalte nicht knappe Ressourcen verstopfen. Ressourcen, die besser dafür verwendet würden, die wirklich *wichtigen* Dinge zu speichern. Wie Tiger. Wie die Gefahren des Feuers. Wie die Notwendigkeit, schnell das Browserfenster mit dem YouTube-Video zu einer Alien-Entführung zu verbergen, wenn Ihr Chef die Nase ins Büro steckt.

Und es gibt keine einfache Möglichkeit, Ihrem Gehirn zu sagen:

»Hey, Gehirn, vielen Dank, aber egal, wie langweilig dieses Buch auch ist und wie klein der Ausschlag auf meiner emotionalen Richterskala gerade ist, ich *will* wirklich, dass du diesen Kram behältst.«

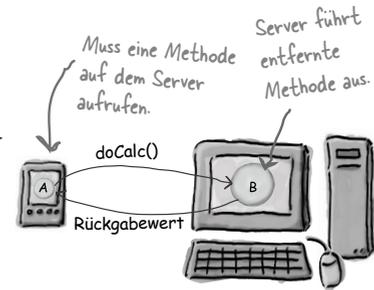


## Wir stellen uns unsere Leser:innen als aktiv Lernende vor.

Also, was ist nötig, damit Sie etwas lernen? Erst einmal müssen Sie es aufnehmen und dann dafür sorgen, dass Sie es nicht wieder vergessen. Es geht nicht darum, Fakten in Ihren Kopf zu schieben. Nach den neuesten Forschungsergebnissen der Kognitionswissenschaft, der Neurobiologie und der Lernpsychologie gehört zum Lernen viel mehr als nur Text auf einer Seite. Wir wissen, was Ihr Gehirn anmacht.

### Einige der Lernprinzipien dieser Buchreihe:

**Bilder einsetzen.** An Bilder kann man sich viel besser erinnern als an Worte allein und lernt so viel effektiver (bis zu 89 % Verbesserung bei Abrufbarkeits- und Lerntransferstudien). Außerdem werden die Dinge dadurch verständlicher. **Text in oder neben die Grafiken setzen**, auf die sie sich beziehen, anstatt darunter oder auf eine andere Seite. Die Leserinnen und Leser werden auf den Bildinhalt bezogene Probleme dann mit *doppelt* so hoher Wahrscheinlichkeit lösen können.



**Verwenden Sie einen gesprächsorientierten Stil mit persönlicher Ansprache.** Nach neueren Untersuchungen haben Studierende nach dem Lernen bei Tests um bis zu 40 % besser abgeschnitten, wenn der Inhalt Leser:innen direkt in der ersten Person und im lockeren Stil angesprochen hat statt in einem formalen Ton. Halten Sie keinen Vortrag, sondern erzählen Sie Geschichten. Benutzen Sie eine zwanglose Sprache. Nehmen Sie sich selbst nicht zu ernst. Würden Sie einer anregenden Unterhaltung beim Abendessen mehr Aufmerksamkeit schenken oder einem Vortrag?

Es ist wirklich ätzend, eine abstrakte Methode zu sein, so ganz ohne Körper ...



**Bringen Sie die Lernenden dazu, intensiver nachzudenken.** Mit anderen Worten: Falls Sie nicht aktiv Ihre Neuronen strapazieren, passiert in Ihrem Gehirn nicht viel. Ein Leser oder eine Leserin muss motiviert, begeistert und neugierig sein und angeregt werden, Probleme zu lösen, Schlüsse zu ziehen und sich neues Wissen anzueignen. Und dafür brauchen Sie Herausforderungen, Übungen, zum Nachdenken anregende Fragen und Tätigkeiten, die beide Seiten des Gehirns und mehrere Sinne einbeziehen.

Ergibt es einen Sinn zu sagen: Wanne IST-EIN Badezimmer? Badezimmer IST-EINE Wanne? Oder ist das eine HAT-EINE-Beziehung?



```
abstract void umherwandern();
```

Kein Body für die Methode! Am Schluss steht ein Semikolon.

**Ziehen Sie die Aufmerksamkeit der Lernenden auf sich – und behalten Sie sie.** Wir alle haben schon Erfahrungen dieser Art gemacht: »Ich will das wirklich lernen, aber ich kann einfach nicht über Seite 1 hinaus wach bleiben.« Ihr Gehirn passt auf, wenn Dinge ungewöhnlich, interessant, merkwürdig, auffällig, unerwartet sind. Ein neues, schwieriges, technisches Thema zu lernen, muss nicht langweilig sein. Wenn es das nicht ist, lernt Ihr Gehirn viel schneller.

**Sprechen Sie Gefühle an.** Wir wissen, dass Ihre Fähigkeit, sich an etwas zu erinnern, wesentlich von dessen emotionalem Gehalt abhängt. Sie erinnern sich an das, was Sie *bewegt*. Sie erinnern sich, wenn Sie etwas *fühlen*. Nein, wir erzählen keine herzerreißenden Geschichten über einen Jungen und seinen Hund. Was wir erzählen, ruft Überraschungs-, Neugier- und Was-soll-das?-Emotionen hervor und dieses Hochgefühl, das Sie beim Lösen eines Puzzles empfinden oder wenn Sie etwas lernen, das alle anderen schwierig finden. Oder wenn Sie merken, dass Sie etwas können, das dieser »Ich-bin-ein-besserer-Techniker-als-du«-Typ aus der Technikabteilung *nicht kann*.



## Metakognition: Nachdenken übers Denken

Wenn Sie wirklich lernen möchten, und zwar schneller und nachhaltiger, dann schenken Sie Ihrer Aufmerksamkeit Aufmerksamkeit. Denken Sie darüber nach, wie Sie denken. Lernen Sie, wie Sie lernen.

Die meisten von uns haben in ihrer Jugend keine Kurse in Metakognition oder Lerntheorie gehabt. Es wurde von uns *erwartet*, dass wir lernen, aber nur selten wurde uns auch *beigebracht*, wie man lernt.

Wir nehmen aber an, dass Sie wirklich etwas über Entwurfsmuster lernen möchten, wenn Sie dieses Buch in den Händen halten. Und wahrscheinlich möchten Sie nicht viel Zeit aufwenden. Und Sie wollen sich an das *erinnern*, was Sie lesen, und es anwenden können. Und deshalb müssen Sie es *verstehen*. Wenn Sie so viel wie möglich von diesem Buch profitieren wollen oder von irgendeinem anderen Buch oder einer anderen Lernerfahrung, übernehmen Sie Verantwortung für Ihr Gehirn. Ihr Gehirn im Zusammenhang mit diesem Lernstoff.

Der Trick besteht darin, Ihr Gehirn dazu zu bringen, neuen Lernstoff als etwas wirklich Wichtiges anzusehen. Als entscheidend für Ihr Wohlbefinden. So wichtig wie einen Tiger. Andernfalls stecken Sie in einem dauerhaften Kampf, in dem Ihr Gehirn sein Bestes gibt, um die neuen Inhalte davon abzuhalten, hängen zu bleiben.



### Wie bringen Sie also Ihr Gehirn dazu, Entwurfsmuster für so wichtig zu halten wie einen Tiger?

Da gibt es den langsamen, ermüdenden Weg oder den schnelleren, effektiveren Weg. Der langsame Weg geht über bloße Wiederholung. Natürlich ist Ihnen klar, dass Sie lernen und sich sogar an die langweiligsten Themen erinnern *können*, wenn Sie sich dieselbe Sache immer wieder einhämmern. Wenn Sie nur oft genug wiederholen, sagt Ihr Gehirn: »Er hat zwar nicht das *Gefühl*, dass das wichtig ist, aber er sieht sich dieselbe Sache *immer und immer wieder* an – dann muss sie wohl wichtig sein.«

Der schnellere Weg besteht darin, **alles zu tun, was die Gehirnaktivität erhöht**, vor allem verschiedene Arten von Gehirnaktivität. Eine wichtige Rolle dabei spielen die auf der vorhergehenden Seite erwähnten Dinge – alles Dinge, die nachweislich dabei helfen, dass Ihr Gehirn *für* Sie arbeitet. So hat sich z. B. in Untersuchungen gezeigt: Wenn Wörter *in* den Abbildungen stehen, die sie beschreiben (und nicht irgendwo anders auf der Seite, z. B. in einer Bildunterschrift oder im Text), versucht Ihr Gehirn, herauszufinden, wie die Wörter und das Bild zusammenhängen, und dadurch feuern mehr Neuronen. Und je mehr Neuronen feuern, umso größer ist die Chance, dass Ihr Gehirn mitbekommt: Bei dieser Sache lohnt es sich, aufzupassen, und vielleicht auch, sich daran zu erinnern.

Ein lockerer Sprachstil hilft, denn Menschen tendieren zu höherer Aufmerksamkeit, wenn ihnen bewusst ist, dass sie ein Gespräch führen – man erwartet dann ja von ihnen, dass sie dem Gespräch folgen und sich beteiligen. Das Erstaunliche daran ist: Es ist Ihrem Gehirn ziemlich egal, dass die »Unterhaltung« zwischen Ihnen und einem Buch stattfindet! Wenn der Schreibstil dagegen formal und trocken ist, hat Ihr Gehirn den gleichen Eindruck wie bei einem Vortrag, bei dem in einem Raum passive Zuhörer sitzen. Nicht nötig, wach zu bleiben.

Aber Abbildungen und ein lockerer Sprachstil sind erst der Anfang.

## Das haben WIR getan:

Wir haben **Bilder** verwendet, weil Ihr Gehirn auf visuelle Eindrücke eingestellt ist, nicht auf Text. Soweit es Ihr Gehirn betrifft, sagt ein Bild *wirklich* mehr als 1.024 Worte. Und dort, wo Text und Abbildungen zusammenwirken, haben wir den Text *in* die Bilder eingebettet, denn Ihr Gehirn arbeitet besser, wenn der Text *innerhalb* der Sache steht, auf die er sich bezieht, und nicht in einer Bildunterschrift oder irgendwo vergraben im Text.

Wir haben **Redundanz** eingesetzt, d. h. dasselbe auf *unterschiedliche* Art und mit verschiedenen Medientypen ausgedrückt, damit Sie es über *mehrere Sinne* aufnehmen. Das erhöht die Chance, dass die Inhalte an mehr als nur einer Stelle in Ihrem Gehirn verankert werden.

Wir haben Konzepte und Bilder in **unerwarteter** Weise eingesetzt, weil Ihr Gehirn auf Neuigkeiten programmiert ist. Und wir haben Bilder und Ideen mit zumindest *etwas emotionalem Charakter* verwendet, weil Ihr Gehirn darauf eingestellt ist, auf die Biochemie von Gefühlen zu achten. An alles, was ein *Gefühl* in Ihnen auslöst, können Sie sich mit höherer Wahrscheinlichkeit erinnern, selbst wenn dieses Gefühl nicht mehr ist als ein bisschen **Belustigung, Überraschung oder Interesse**.

Wir haben einen **umgangssprachlichen Stil** mit direkter Anrede benutzt, denn Ihr Gehirn ist von Natur aus aufmerksamer, wenn es Sie in einer Unterhaltung wähnt, als wenn es davon ausgeht, dass Sie passiv einer Präsentation zuhören – sogar dann, wenn Sie *lesen*.

Wir haben mehr als 100 **Aktivitäten** für Sie vorgesehen, denn Ihr Gehirn lernt und behält von Natur aus besser, wenn Sie Dinge *tun*, als wenn Sie nur darüber *lesen*. Und wir haben die Übungen zwar anspruchsvoll, aber doch lösbar gemacht, denn so ist es den meisten Lesern am liebsten.

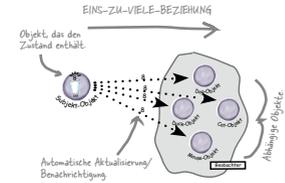
Wir haben **mehrere unterschiedliche Lernstile** eingesetzt, denn vielleicht bevorzugen *Sie* ein Schritt-für-Schritt-Vorgehen, während ein anderer erst einmal den groben Zusammenhang verstehen und ein Dritter einfach nur ein Codebeispiel sehen möchte. Aber ganz abgesehen von den jeweiligen Lernvorlieben profitiert *jeder* davon, wenn er die gleichen Inhalte in unterschiedlicher Form präsentiert bekommt.

Wir liefern Inhalte für **beide Seiten Ihres Gehirns**, denn je mehr Sie von Ihrem Gehirn einsetzen, umso wahrscheinlicher werden Sie lernen und behalten, und umso länger bleiben Sie konzentriert. Wenn Sie mit einer Seite des Gehirns arbeiten, bedeutet das häufig, dass sich die andere Seite des Gehirns ausruhen kann; so können Sie über einen längeren Zeitraum produktiver lernen.

Und wir haben **Geschichten** und Übungen aufgenommen, die **mehr als einen Blickwinkel repräsentieren**, denn Ihr Gehirn lernt von Natur aus intensiver, wenn es gezwungen ist, selbst zu analysieren und zu beurteilen.

Wir haben **Herausforderungen** eingefügt: in Form von Übungen und indem wir **Fragen** stellen, auf die es nicht immer eine eindeutige Antwort gibt, denn Ihr Gehirn ist darauf eingestellt, zu lernen und sich zu erinnern, wenn es an etwas *arbeiten* muss. Überlegen Sie: Ihren *Körper* bekommen Sie ja auch nicht in Form, wenn Sie die Menschen auf dem Sportplatz nur *beobachten*. Aber wir haben unser Bestes getan, um dafür zu sorgen, dass Sie – wenn Sie schon hart arbeiten – an den *richtigen* Dingen arbeiten. Dass Sie **nicht einen einzigen Dendriten darauf verschwenden**, ein schwer verständliches Beispiel zu verarbeiten oder einen schwierigen, mit Fachbegriffen gespickten oder übermäßig gedrängten Text zu analysieren.

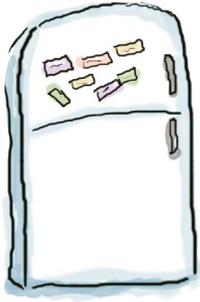
Wir haben **Menschen** eingesetzt. In Geschichten, Beispielen, Bildern usw. – denn *Sie sind* ein Mensch. Und Ihr Gehirn schenkt *Menschen* mehr Aufmerksamkeit als *Dingen*.



Die Meisterin  
der Muster

## Punkt für Punkt





Schneiden Sie dies aus und heften Sie es an Ihren Kühlschrank.

## Und das können SIE tun, um sich Ihr Gehirn untertan zu machen

So, wir haben unseren Teil der Arbeit geleistet. Der Rest liegt bei Ihnen. Diese Tipps sind ein Anfang; hören Sie auf Ihr Gehirn und finden Sie heraus, was bei Ihnen funktioniert und was nicht. Probieren Sie neue Wege aus.

### ① Immer langsam. Je mehr Sie verstehen, umso weniger müssen Sie auswendig lernen.

Lesen Sie nicht nur. Halten Sie inne und denken Sie nach. Wenn das Buch Sie etwas fragt, springen Sie nicht einfach zur Antwort. Stellen Sie sich vor, dass Sie das wirklich jemand *fragt*. Je gründlicher Sie Ihr Gehirn zum Nachdenken zwingen, umso größer ist die Chance, dass Sie lernen und behalten.

### ② Bearbeiten Sie die Übungen. Machen Sie sich selbst Notizen.

Wir haben sie entworfen, aber wenn wir sie auch für Sie lösen würden, wäre dass, als würde ein anderer Ihr Training für Sie absolvieren. Und sehen Sie sich die Übungen *nicht einfach nur an*. **Benutzen Sie einen Bleistift.** Es deutet vieles darauf hin, dass körperliche Aktivität beim Lernen den Lernerfolg erhöhen kann.

### ③ Lesen Sie die Abschnitte »Es gibt keine Dummen Fragen«.

Und zwar alle. Das sind keine Zusatzanmerkungen – **sie gehören zum Kerninhalt!** Überspringen Sie sie nicht.

### ④ Lesen Sie dies als Letztes vor dem Schlafengehen. Oder lesen Sie danach zumindest nichts Anspruchsvolles mehr.

Ein Teil des Lernprozesses (vor allem die Übertragung in das Langzeitgedächtnis) findet erst statt, *nachdem* Sie das Buch zur Seite gelegt haben. Ihr Gehirn braucht Zeit für sich, um weitere Verarbeitung zu leisten. Wenn Sie in dieser Zeit etwas Neues aufnehmen, geht ein Teil dessen, was Sie gerade gelernt haben, verloren.

### ⑤ Trinken Sie Wasser. Viel.

Ihr Gehirn arbeitet am besten in einem schönen Flüssigkeitsbad. Austrocknung (zu der es schon kommen kann, bevor Sie überhaupt Durst verspüren) beeinträchtigt die kognitive Funktion.

### ⑥ Reden Sie drüber. Laut.

Sprechen aktiviert einen anderen Teil des Gehirns. Wenn Sie etwas verstehen wollen oder Ihre Chancen verbessern möchten, sich später daran zu erinnern, sagen Sie es laut. Noch besser: Versuchen Sie, es jemand anderem laut zu erklären. Sie lernen dann schneller und haben vielleicht Ideen, auf die Sie beim bloßen Lesen nie gekommen wären.

### ⑦ Hören Sie auf Ihr Gehirn.

Achten Sie darauf, Ihr Gehirn nicht zu überladen. Wenn Sie merken, dass Sie etwas nur noch überfliegen oder dass Sie das gerade erst Gelesene vergessen haben, ist es Zeit für eine Pause. Ab einem bestimmten Punkt lernen Sie nicht mehr schneller, indem Sie mehr hineinzustopfen versuchen; das kann sogar den Lernprozess stören.

### ⑧ Aber bitte mit Gefühl!

Ihr Gehirn muss wissen, dass es *um etwas Wichtiges geht*. Lassen Sie sich in die Geschichten hineinziehen. Erfinden Sie eigene Bildunterschriften für die Fotos. Über einen schlechten Scherz zu stöhnen, ist *immer noch* besser, als gar nichts zu fühlen.

### ⑨ Entwerfen Sie etwas!

Wenden Sie das hier Gelernte auf einen Entwurf an, an dem Sie gerade arbeiten, oder gestalten Sie ein älteres Projekt damit um. Tun Sie irgendetwas, um weitere Erfahrungen zu sammeln, die über die Übungen und Aktivitäten in diesem Buch hinausgehen. Sie brauchen dazu nur einen Bleistift und ein Problem, das es zu lösen gilt ... ein Problem, das von einem oder mehreren Entwurfsmustern profitieren würde.