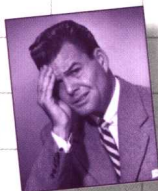


深入浅出设计模式 (影印版)

# Head First Design Patterns

Avoid those  
embarrassing  
coupling mistakes



Learn why everything  
your friends know about Factory  
pattern is  
probably wrong



Discover the secrets  
of the Patterns Guru

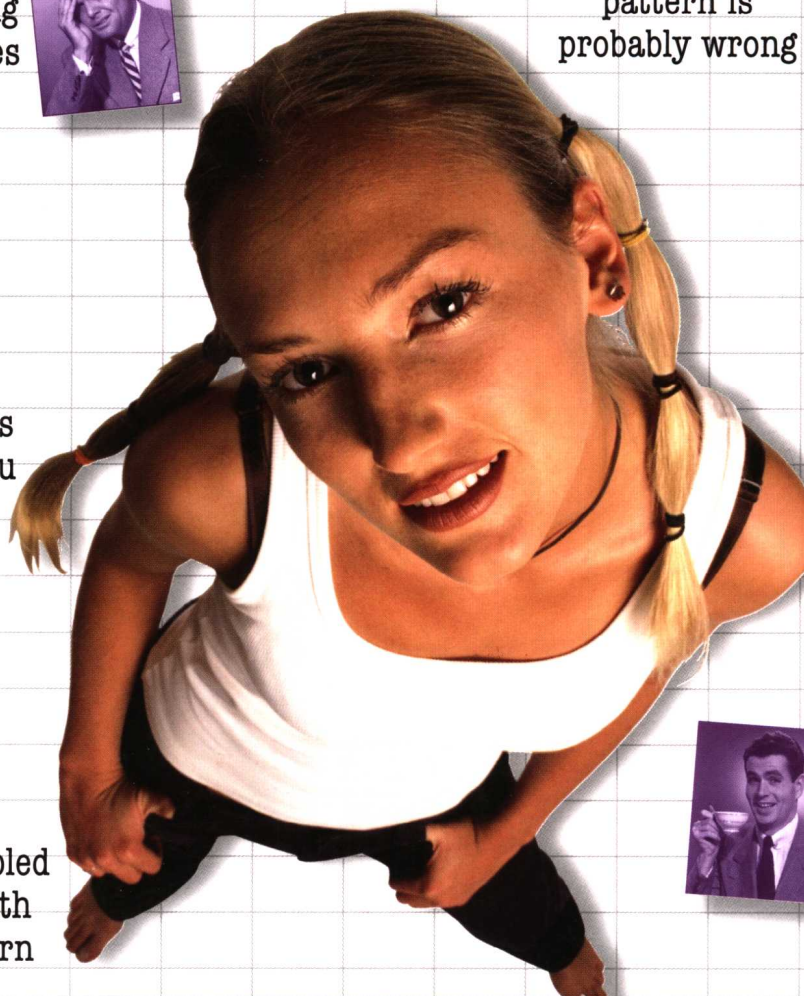


Find out how  
Starbuzz Coffee doubled  
their stock price with  
the Decorator pattern

Load the patterns  
that matter straight  
into your brain



See why Jim's  
love life improved  
when he cut down  
his inheritance



O'REILLY® 東南大學 出版社

Eric Freeman & Elisabeth Freeman  
with Kathy Sierra & Bert Bates 著

# 深入浅出设计模式 (影印版)

## Head First Design Patterns

Wouldn't it be dreamy if  
there was a Design Patterns  
book that was more fun than going  
to the dentist, and more revealing  
than an IRS form? It's probably  
just a fantasy...



江苏工业学院图书馆  
Elisabeth Freeman  
藏书章  
with  
Kathy Sierra  
Bert Bates

**O'REILLY®**

Beijing • Cambridge • Köln • Paris • Sebastopol • Taipei • Tokyo

O'Reilly Media, Inc. 授权东南大学出版社出版

东南大学出版社

## 图书在版编目 (CIP) 数据

深入浅出设计模式 / (美) 弗里曼 (Freeman, E.) 等著.

影印本. - 南京: 东南大学出版社, 2005.10

书名原文: Head First Design Patterns

ISBN 7-5641-0165-2

I. 深... II. 弗... III. 软件设计-英文 IV. TP311.5

中国版本图书馆 CIP 数据核字 (2005) 第 103058 号

江苏省版权局著作权合同登记

图字: 10-2005-202 号

©2004 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2005. Authorized reprint of the original English edition, 2004 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2004。

英文影印版由东南大学出版社出版 2005。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

书 名 / 深入浅出设计模式 (影印版)

书 号 / ISBN 7-5641-0165-2

责任编辑 / 张烨

封面设计 / Ellie Volckhausen, 张健

出版发行 / 东南大学出版社 (press.seu.edu.cn)

地 址 / 南京四牌楼 2 号 (邮政编码 210096)

印 刷 / 扬中市印刷有限公司

开 本 / 787 毫米 × 980 毫米 12 开本 56.5 印张

版 次 / 2005 年 11 月第 1 版 2005 年 11 月第 1 次印刷

印 数 / 0001-3000 册

定 价 / 98.00 元 (册)

## **Praise for Head First Design Patterns**

"I received the book yesterday and started to read it on the way home... and I couldn't stop. I took it to the gym and I expect people saw me smiling a lot while I was exercising and reading. This is tres 'cool'. It is fun but they cover a lot of ground and they are right to the point. I'm really impressed."

— **Erich Gamma, IBM Distinguished Engineer,  
and co-author of Design Patterns**

"Head First Design Patterns' manages to mix fun, belly-laughs, insight, technical depth and great practical advice in one entertaining and thought provoking read. Whether you are new to design patterns, or have been using them for years, you are sure to get something from visiting Objectville."

— **Richard Helm, coauthor of "Design Patterns" with rest of the  
Gang of Four - Erich Gamma, Ralph Johnson and John Vlissides**

"I feel like a thousand pounds of books have just been lifted off of my head."

— **Ward Cunningham, inventor of the Wiki  
and founder of the Hillside Group**

"This book is close to perfect, because of the way it combines expertise and readability. It speaks with authority and it reads beautifully. It's one of the very few software books I've ever read that strikes me as indispensable. (I'd put maybe 10 books in this category, at the outside.)"

— **David Gelernter, Professor of Computer Science,  
Yale University and author of "Mirror Worlds" and "Machine Beauty"**

"A Nose Dive into the realm of patterns, a land where complex things become simple, but where simple things can also become complex. I can think of no better tour guides than the Freemans."

— **Miko Matsumura, Industry Analyst, The Middleware Company  
Former Chief Java Evangelist, Sun Microsystems**

"I laughed, I cried, it moved me."

— **Daniel Steinberg, Editor-in-Chief, java.net**

"My first reaction was to roll on the floor laughing. After I picked myself up, I realized that not only is the book technically accurate, it is the easiest to understand introduction to design patterns that I have seen."

— **Dr. Timothy A. Budd, Associate Professor of Computer Science at  
Oregon State University and author of more than a dozen books,  
including "C++ for Java Programmers"**

"Jerry Rice runs patterns better than any receiver in the NFL, but the Freemans have out run him. Seriously...this is one of the funniest and smartest books on software design I've ever read."

— **Aaron LaBerge, VP Technology, ESPN.com**

## More Praise for *Head First Design Patterns*

“Great code design is, first and foremost, great information design. A code designer is teaching a computer how to do something, and it is no surprise that a great teacher of computers should turn out to be a great teacher of programmers. This book’s admirable clarity, humor and substantial doses of clever make it the sort of book that helps even non-programmers think well about problem-solving.”

— **Cory Doctorow, co-editor of *Boing Boing*  
and author of “*Down and Out in the Magic Kingdom*”  
and “*Someone Comes to Town, Someone Leaves Town*”**

“There’s an old saying in the computer and videogame business – well, it can’t be that old because the discipline is not all that old – and it goes something like this: Design is Life. What’s particularly curious about this phrase is that even today almost no one who works at the craft of creating electronic games can agree on what it means to “design” a game. Is the designer a software engineer? An art director? A storyteller? An architect or a builder? A pitch person or a visionary? Can an individual indeed be in part all of these? And most importantly, who the %\$!#&\* cares?”

It has been said that the “designed by” credit in interactive entertainment is akin to the “directed by” credit in filmmaking, which in fact allows it to share DNA with perhaps the single most controversial, overstated, and too often entirely lacking in humility credit grab ever propagated on commercial art. Good company, eh? Yet if Design is Life, then perhaps it is time we spent some quality cycles thinking about what it is.

Eric and Elisabeth Freeman have intrepidly volunteered to look behind the code curtain for us in “Head First Design Patterns.” I’m not sure either of them cares all that much about the PlayStation or X-Box, nor should they. Yet they do address the notion of design at a significantly honest level such that anyone looking for ego reinforcement of his or her own brilliant auteurship is best advised not to go digging here where truth is stunningly revealed. Sophists and circus barkers need not apply. Next generation literati please come equipped with a pencil.”

— **Ken Goldstein, Executive Vice President & Managing Director,  
Disney Online**

“Just the right tone for the geeked-out, casual-cool guru coder in all of us. The right reference for practical development strategies—gets my brain going without having to slog through a bunch of tired, stale professor-speak.”

— **Travis Kalanick, Founder of Scour and Red Swoosh  
Member of the MIT TR100**

“This book combines good humors, great examples, and in-depth knowledge of Design Patterns in such a way that makes learning fun. Being in the entertainment technology industry, I am intrigued by the Hollywood Principle and the home theater Facade Pattern, to name a few. The understanding of Design Patterns not only helps us create reusable and maintainable quality software, but also helps sharpen our problem-solving skills across all problem domains. This book is a must read for all computer professionals and students.”

— **Newton Lee, Founder and Editor-in-Chief, Association for Computing  
Machinery’s (ACM) Computers in Entertainment ([acmcie.org](http://acmcie.org))**

## **Praise for the *Head First* approach**

“Java technology is everywhere—in mobile phones, cars, cameras, printers, games, PDAs, ATMs, smart cards, gas pumps, sports stadiums, medical devices, Web cams, servers, you name it. If you develop software and haven’t learned Java, it’s definitely time to dive in—Head First.”

— **Scott McNealy, Sun Microsystems Chairman, President and CEO**

“It’s fast, irreverent, fun, and engaging. Be careful—you might actually learn something!”

— **Ken Arnold, former Senior Engineer at Sun Microsystems**  
**Co-author (with James Gosling, creator of Java),**  
**“The Java Programming Language”**

“Head First Java is like Monty Python meets the gang of four... the text is broken up so well by puzzles and stories, quizzes and examples, that you cover ground like no computer book before.”

— **Douglas Rowe, Columbia Java Users Group**

“ ‘Head First Java’ ... gives new meaning to their marketing phrase ‘There’s an O Reilly for that.’ I picked this up because several others I respect had described it in terms like ‘revolutionary’ and a described a radically different approach to the textbook. They were (are) right... In typical O’Reilly fashion, they’ve taken a scientific and well considered approach. The result is funny, irreverent, topical, interactive, and brilliant...Reading this book is like sitting in the speakers lounge at a view conference, learning from – and laughing with – peers... If you want to UNDERSTAND Java, go buy this book.”

— **Andrew Pollack, [www.thenorth.com](http://www.thenorth.com)**

“If you want to *learn* Java, look no further: welcome to the first GUI-based technical book! This perfectly-executed, ground-breaking format delivers benefits other Java texts simply can’t... Prepare yourself for a truly remarkable ride through Java land.”

— **Neil R. Bauman, Captain & CEO, Geek Cruises ([www.GeekCruises.com](http://www.GeekCruises.com))**

What a fantastic way to learn!!! I CAN NOT PUT THIS BOOK DOWN!!! My 3 year old woke up at 1:40 a.m. this morning, and I put him back to bed with book in hand and a flashlight so I could continue to read for about another hour.

— **Ross Goldberg**

“This stuff is so fricking good it makes me wanna WEEP! I’m stunned.”

— **Floyd Jones, Senior Technical Writer/Poolboy, BEA**

**Other related books from O'Reilly**

Head First Java  
Head First EJB  
Head First Servlets & JSP  
Learning Java  
Java in a Nutshell  
Java Enterprise in a Nutshell  
Java Examples in a Nutshell  
Java Cookbook  
J2EE Design Patterns

**Be watching for more books in the Head First series!**

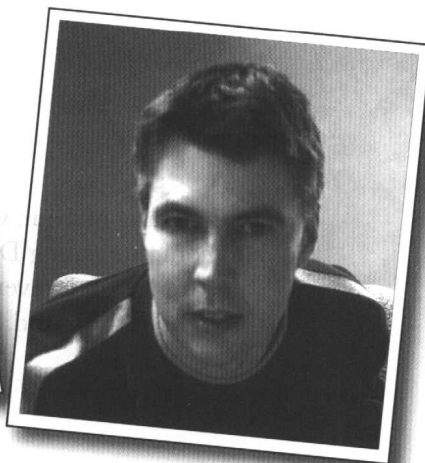
To the Gang of Four, whose insight and expertise in capturing and communicating Design Patterns has changed the face of software design forever, and bettered the lives of developers throughout the world.

But seriously, *when* are we going to see a second edition? After all, it's been only *ten years*!



## Authors/Developers of Head First Design Patterns

Elisabeth Freeman



Eric Freeman

**Elisabeth** is an author, software developer and digital artist. She's been involved with the Internet since the early days, having co-founded The Ada Project (TAP), an award winning web site for women in computing now adopted by the ACM. More recently Elisabeth lead research and development efforts in digital media at the Walt Disney Company where she co-invented Motion, a content system that delivers terabytes of video every day to Disney, ESPN and Movies.com users.

Elisabeth is a computer scientist at heart and holds graduate degrees in Computer Science from Yale University and Indiana University. She's worked in a variety of areas including visual languages, RSS syndication and Internet systems. She's also been an active advocate for women in computing, developing programs that encourage women to enter the field. These days you'll find her sipping some Java or Cocoa on her Mac, although she dreams of a day when the whole world is using Scheme.

Elisabeth has loved hiking and the outdoors since her days growing up in Scotland. When she's outdoors her camera is never far. She's also an avid cyclist, vegetarian and animal lover.

You can send her email at [beth@wickedlysmart.com](mailto:beth@wickedlysmart.com)

**Eric** is a computer scientist with a passion for media and software architectures. He just wrapped up four years at a dream job – directing Internet broadband and wireless efforts at Disney – and is now back to writing, creating cool software and hacking Java and Macs.

Eric spent a lot of the '90s working on alternatives to the desktop metaphor with David Gelernter (and they're both *still* asking the question “why do I have to give a file a name?”). Based on this work, Eric landed a Ph.D. at Yale University in '97. He also co-founded Mirror Worlds Technologies (now acquired) to create a commercial version of his thesis work, Lifestreams.

In a previous life, Eric built software for networks and supercomputers. You might know him from such books as *JavaSpaces Principles Patterns and Practice*. Eric has fond memories of implementing tuple-space systems on Thinking Machine CM-5s and creating some of the first Internet information systems for NASA in the late 80s.

Eric is currently living in the high desert near Santa Fe. When he's not writing text or code you'll find him spending more time tweaking than watching his home theater and trying to restore a circa 1980s Dragon's Lair video game. He also wouldn't mind moonlighting as an electronica DJ.

Write to him at [eric@wickedlysmart.com](mailto:eric@wickedlysmart.com) or visit his blog at <http://www.ericfreeman.com>

## Creators of the Head First series (and co-conspirators on this book)

Kathy Sierra



**Kathy** has been interested in learning theory since her days as a game designer (she wrote games for Virgin, MGM, and Amblin'). She developed much of the Head First format while teaching New Media Authoring for UCLA Extension's Entertainment Studies program. More recently, she's been a master trainer for Sun Microsystems, teaching Sun's Java instructors how to teach the latest Java technologies, and developing several of Sun's certification exams. Together with Bert Bates, she has been actively using the Head First concepts to teach thousands of developers. Kathy is the founder of javaranch.com, which won a 2003 and 2004 Software Development magazine Jolt Cola Productivity Award. You might catch her teaching Java on the Java Jam Geek Cruise ([geekcruises.com](http://geekcruises.com)).

She recently moved from California to Colorado, where she's had to learn new words like, "ice scraper" and "fleece", but the lightning there is fantastic.

Likes: running, skiing, skateboarding, playing with her Icelandic horse, and weird science. Dislikes: entropy.

You can find her on javaranch, or occasionally blogging on java.net. Write to her at [kathy@wickedlysmart.com](mailto:kathy@wickedlysmart.com).

Bert Bates



**Bert** is a long-time software developer and architect, but a decade-long stint in artificial intelligence drove his interest in learning theory and technology-based training. He's been helping clients becoming better programmers ever since. Recently, he's been heading up the development team for several of Sun's Java Certification exams.

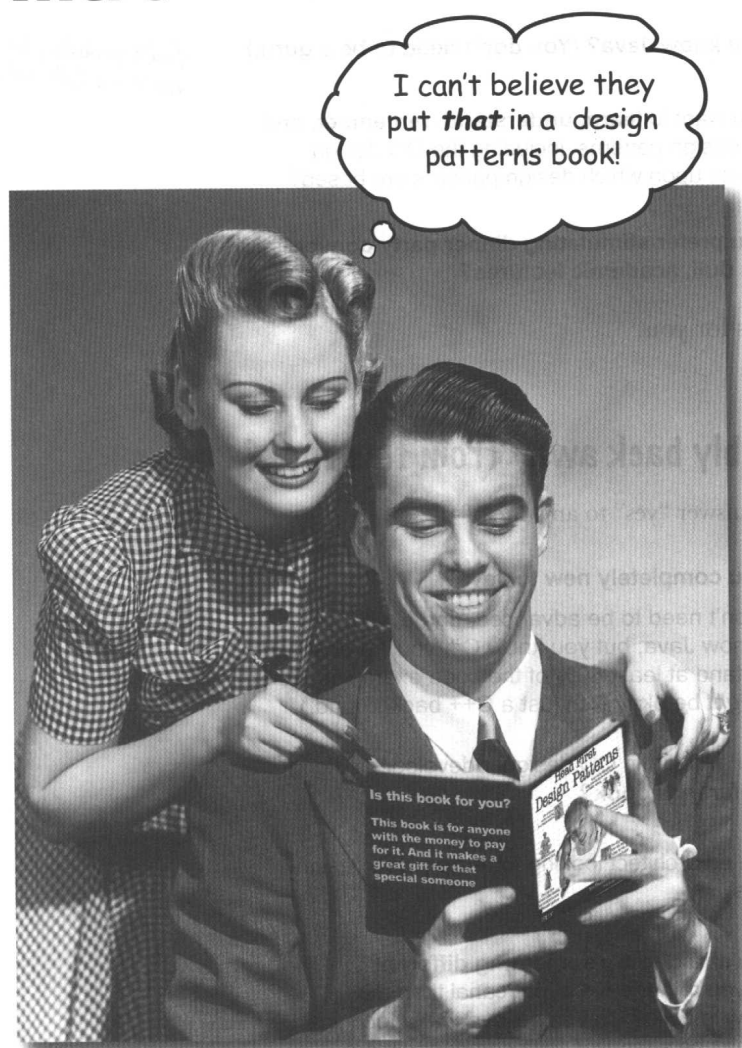
He spent the first decade of his software career travelling the world to help broadcast clients like Radio New Zealand, the Weather Channel, and the Arts & Entertainment Network (A & E). One of his all-time favorite projects was building a full rail system simulation for Union Pacific Railroad.

Bert is a long-time, hopelessly addicted *go* player, and has been working on a *go* program for way too long. He's a fair guitar player and is now trying his hand at banjo.

Look for him on javaranch, on the IGS go server, or you can write to him at [terrapi@wickedlysmart.com](mailto:terrapi@wickedlysmart.com).

how to use this book

## Intro



In this section, we answer the burning question:  
"So, why *DID* they put that in a design patterns book?"

## Who is this book for?

If you can answer “yes” to all of these:

- ① **Do you know Java? (You don't need to be a guru.)**
- ② **Do you want to *learn, understand, remember, and apply* design patterns, including the OO design principles upon which design patterns are based?**
- ③ **Do you prefer stimulating dinner party conversation to dry, dull, academic lectures?**

*You'll probably be okay if you know C# instead.*

this book is for you.

## Who should probably back away from this book?

If you can answer “yes” to any one of these:

- ① **Are you completely new to Java?**  
(You don't need to be advanced, and even if you *don't* know Java, but you know C#, you'll probably understand at least 80% of the code examples. You also *might* be okay with just a C++ background.)
- ② **Are you a kick-butt OO designer/developer looking for a *reference* book?**
- ③ **Are you an architect looking for *enterprise* design patterns?**
- ④ **Are you *afraid to try something different*?**  
Would you rather have a root canal than mix stripes with plaid? Do you believe that a technical book can't be serious if Java components are anthropomorphized?

this book is not for you.



*[note from marketing: this book is for anyone with a credit card.]*

## We know what you're thinking.

"How can this be a serious programming book?"

"What's with all the graphics?"

"Can I actually learn it this way?"

## And we know what your *brain* is thinking.

Your brain craves novelty. It's always searching, scanning, *waiting* for something unusual. It was built that way, and it helps you stay alive.

Today, you're less likely to be a tiger snack. But your brain's still looking. You just never know.

So what does your brain do with all the routine, ordinary, normal things you encounter? Everything it *can* to stop them from interfering with the brain's *real* job—recording things that *matter*. It doesn't bother saving the boring things; they never make it past the "this is obviously not important" filter.

How does your brain *know* what's important? Suppose you're out for a day hike and a tiger jumps in front of you, what happens inside your head and body?

Neurons fire. Emotions crank up. *Chemicals surge.*

And that's how your brain knows...

### **This must be important! Don't forget it!**

But imagine you're at home, or in a library. It's a safe, warm, tiger-free zone. You're studying. Getting ready for an exam. Or trying to learn some tough technical topic your boss thinks will take a week, ten days at the most.

Just one problem. Your brain's trying to do you a big favor. It's trying to make sure that this *obviously* non-important content doesn't clutter up scarce resources. Resources that are better spent storing the really *big* things. Like tigers. Like the danger of fire. Like how you should never again snowboard in shorts.

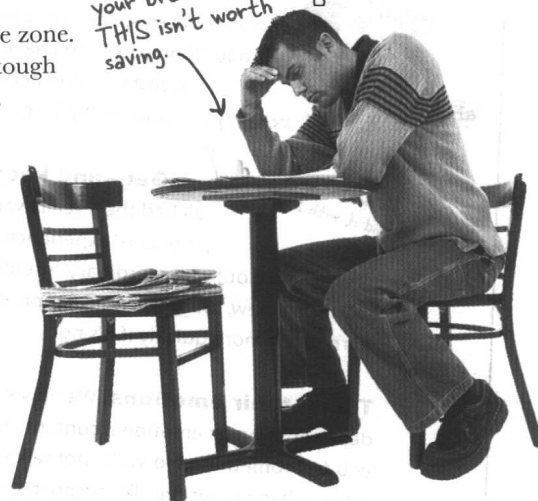
And there's no simple way to tell your brain, "Hey brain, thank you very much, but no matter how dull this book is, and how little I'm registering on the emotional Richter scale right now, I really *do* want you to keep this stuff around."

your brain thinks  
THIS is important.



Great. Only  
637 more dull, dry,  
boring pages.

your brain thinks  
THIS isn't worth  
saving.

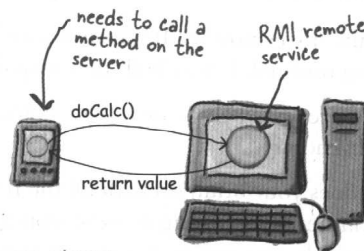


## We think of a “Head First” reader as a learner.

So what does it take to *learn* something? First, you have to *get* it, then make sure you don’t *forget* it. It’s not about pushing facts into your head. Based on the latest research in cognitive science, neurobiology, and educational psychology, *learning* takes a lot more than text on a page. We know what turns your brain on.

### Some of the Head First learning principles:

**Make it visual.** Images are far more memorable than words alone, and make learning much more effective (up to 89% improvement in recall and transfer studies). It also makes things more understandable. **Put the words within or near the graphics** they relate to, rather than on the bottom or on another page, and learners will be up to twice as likely to solve problems related to the content.



**Use a conversational and personalized style.** In recent studies, students performed up to 40% better on post-learning tests if the content spoke directly to the reader, using a first-person, conversational style rather than taking a formal tone. Tell stories instead of lecturing. Use casual language. Don’t take yourself too seriously. Which would you pay more attention to: a stimulating dinner party companion, or a lecture?

It really sucks to be an abstract method. You don’t have a body.



abstract void roam();

No method body!  
End it with a semicolon.

**Get the learner to think more deeply.** In other words, unless you actively flex your neurons, nothing much happens in your head. A reader has to be motivated, engaged, curious, and inspired to solve problems, draw conclusions, and generate new knowledge. And for that, you need challenges, exercises, and thought-provoking questions, and activities that involve both sides of the brain, and multiple senses.

Does it make sense to say Tub IS-A Bathroom? Bathroom IS-A Tub? Or is it a HAS-A relationship?



**Get—and keep—the reader’s attention.** We’ve all had the “I really want to learn this but I can’t stay awake past page one” experience. Your brain pays attention to things that are out of the ordinary, interesting, strange, eye-catching, unexpected. Learning a new, tough, technical topic doesn’t have to be boring. Your brain will learn much more quickly if it’s not.

**Touch their emotions.** We now know that your ability to remember something is largely dependent on its emotional content. You remember what you *care* about. You remember when you *feel* something. No, we’re not talking heart-wrenching stories about a boy and his dog. We’re talking emotions like surprise, curiosity, fun, “what the...?”, and the feeling of “I Rule!” that comes when you solve a puzzle, learn something everybody else thinks is hard, or realize you know something that “I’m more technical than thou” Bob from engineering *doesn’t*.



# Metacognition: thinking about thinking

If you really want to learn, and you want to learn more quickly and more deeply, pay attention to how you pay attention. Think about how you think. Learn how you learn.

Most of us did not take courses on metacognition or learning theory when we were growing up. We were *expected* to learn, but rarely *taught* to learn.

But we assume that if you're holding this book, you really want to learn design patterns. And you probably don't want to spend a lot of time. And you want to *remember* what you read, and be able to apply it. And for that, you've got to *understand* it. To get the most from this book, or *any* book or learning experience, take responsibility for your brain. Your brain on *this* content.

The trick is to get your brain to see the new material you're learning as Really Important. Crucial to your well-being. As important as a tiger. Otherwise, you're in for a constant battle, with your brain doing its best to keep the new content from sticking.

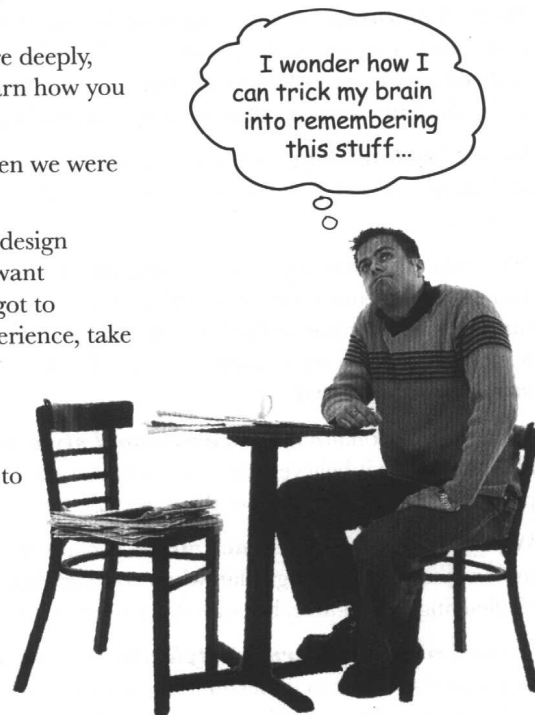
## So how **DO** you get your brain to think Design Patterns are as important as a tiger?

There's the slow, tedious way, or the faster, more effective way. The slow way is about sheer repetition. You obviously know that you *are* able to learn and remember even the dullest of topics, if you keep pounding on the same thing. With enough repetition, your brain says, "This doesn't *feel* important to him, but he keeps looking at the same thing *over and over and over*, so I suppose it must be."

The faster way is to do **anything that increases brain activity**, especially different *types* of brain activity. The things on the previous page are a big part of the solution, and they're all things that have been proven to help your brain work in your favor. For example, studies show that putting words *within* the pictures they describe (as opposed to somewhere else in the page, like a caption or in the body text) causes your brain to try to makes sense of how the words and picture relate, and this causes more neurons to fire. More neurons firing = more chances for your brain to *get* that this is something worth paying attention to, and possibly recording.

A conversational style helps because people tend to pay more attention when they perceive that they're in a conversation, since they're expected to follow along and hold up their end. The amazing thing is, your brain doesn't necessarily *care* that the "conversation" is between you and a book! On the other hand, if the writing style is formal and dry, your brain perceives it the same way you experience being lectured to while sitting in a roomful of passive attendees. No need to stay awake.

But pictures and conversational style are just the beginning.





## Here's what WE did:

We used **pictures**, because your brain is tuned for visuals, not text. As far as your brain's concerned, a picture really *is* worth 1024 words. And when text and pictures work together, we embedded the text *in* the pictures because your brain works more effectively when the text is *within* the thing the text refers to, as opposed to in a caption or buried in the text somewhere.

We used **redundancy**, saying the same thing in *different* ways and with different media types, and *multiple senses*, to increase the chance that the content gets coded into more than one area of your brain.

We used concepts and pictures in **unexpected** ways because your brain is tuned for novelty, and we used pictures and ideas with at least *some* **emotional content**, because your brain is tuned to pay attention to the biochemistry of emotions. That which causes you to *feel* something is more likely to be remembered, even if that feeling is nothing more than a little **humor**, **surprise**, or **interest**.

We used a personalized, **conversational style**, because your brain is tuned to pay more attention when it believes you're in a conversation than if it thinks you're passively listening to a presentation. Your brain does this even when you're *reading*.

We included more than 40 **activities**, because your brain is tuned to learn and remember more when you *do* things than when you *read* about things. And we made the exercises challenging-yet-do-able, because that's what most *people* prefer.

We used **multiple learning styles**, because *you* might prefer step-by-step procedures, while someone else wants to understand the big picture first, while someone else just wants to see a code example. But regardless of your own learning preference, *everyone* benefits from seeing the same content represented in multiple ways.

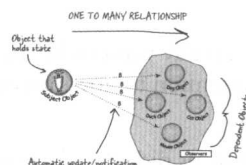
We include content for **both sides of your brain**, because the more of your brain you engage, the more likely you are to learn and remember, and the longer you can stay focused. Since working one side of the brain often means giving the other side a chance to rest, you can be more productive at learning for a longer period of time.

And we included **stories** and exercises that present **more than one point of view**, because your brain is tuned to learn more deeply when it's forced to make evaluations and judgements.

We included **challenges**, with exercises, and by asking **questions** that don't always have a straight answer, because your brain is tuned to learn and remember when it has to *work* at something. Think about it—you can't get your *body* in shape just by *watching* people at the gym. But we did our best to make sure that when you're working hard, it's on the *right* things. That **you're not spending one extra dendrite** processing a hard-to-understand example, or parsing difficult, jargon-laden, or overly terse text.

We used **people**. In stories, examples, pictures, etc., because, well, because *you're* a person. And your brain pays more attention to *people* than it does to *things*.

We used an **80/20** approach. We assume that if you're going for a PhD in software design, this won't be your only book. So we don't talk about *everything*. Just the stuff you'll actually *need*.



The Patterns Guru



BULLET POINTS

Puzzles







## Here's what YOU can do to bend your brain into submission

So, we did our part. The rest is up to you. These tips are a starting point; listen to your brain and figure out what works for you and what doesn't. Try new things.

cut this out and stick it  
on your refrigerator.

### 1 Slow down. The more you understand, the less you have to memorize.

Don't just *read*. Stop and think. When the book asks you a question, don't just skip to the answer. Imagine that someone really is asking the question. The more deeply you force your brain to think, the better chance you have of learning and remembering.

### 2 Do the exercises. Write your own notes.

We put them in, but if we did them for you, that would be like having someone else do your workouts for you. And don't just *look* at the exercises. **Use a pencil.** There's plenty of evidence that physical activity *while* learning can increase the learning.

### 3 Read the "There are No Dumb Questions"

That means all of them. They're not optional side-bars—***they're part of the core content!*** Don't skip them.

### 4 Make this the last thing you read before bed. Or at least the last *challenging* thing.

Part of the learning (especially the transfer to long-term memory) happens *after* you put the book down. Your brain needs time on its own, to do more processing. If you put in something new during that processing-time, some of what you just learned will be lost.

### 5 Drink water. Lots of it.

Your brain works best in a nice bath of fluid. Dehydration (which can happen before you ever feel thirsty) decreases cognitive function.

### 6 Talk about it. Out loud.

Speaking activates a different part of the brain. If you're trying to understand something, or increase your chance of remembering it later, say it out loud. Better still, try to explain it out loud to someone else. You'll learn more quickly, and you might uncover ideas you hadn't known were there when you were reading about it.

### 7 Listen to your brain.

Pay attention to whether your brain is getting overloaded. If you find yourself starting to skim the surface or forget what you just read, it's time for a break. Once you go past a certain point, you won't learn faster by trying to shove more in, and you might even hurt the process.

### 8 Feel something!

Your brain needs to know that this *matters*. Get involved with the stories. Make up your own captions for the photos. Groaning over a bad joke is *still* better than feeling nothing at all.

### 9 Design something!

Apply this to something new you're designing, or refactor an older project. Just do *something* to get some experience beyond the exercises and activities in this book. All you need is a pencil and a problem to solve... a problem that might benefit from one or more design patterns.