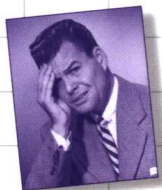


重视大脑的学习指南

Head First 设计模式

(中文版)

避免一些
尴尬的
耦合错误

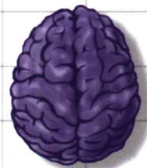


学习为何朋友们
对工厂模式
的认知
可能有错



发掘模式大师
的秘密

把事关紧要的模式
直接装入脑海里



探究星巴兹咖啡
如何以装饰者模式
让自己的股价翻倍



瞧瞧Jim
为何拒绝继承后
改善了爱情生活



O'REILLY®

中国电力出版社

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

O'Reilly Taiwan公司译 UMLChina 改编

TP311.5/195

2007

Head First设计模式

(中文版)

如果有一本设计模式书，读起来比看牙医更有趣，内容比国税局表单更容易理解那该多好啊！这可能只是个梦想吧……



Eric Freeman,
Elisabeth Freeman,
Kathy Sierra &
Bert Bates 著
O'Reilly Taiwan公司译
UMLChina 改编

O'REILLY®

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

O'Reilly Media, Inc. 授权中国电力出版社出版

中国电力出版社

图书在版编目 (CIP) 数据

Head First设计模式 (中文版) / (美) 弗里曼 (Freeman, E.) 等著; O'Reilly Taiwan公司译, UMLChina改编—北京: 中国电力出版社, 2007

书名原文: Head First Design Pattern

ISBN 978-7-5083-5393-7

I. H... II.①弗... ②O... ③U... III. 软件设计 IV. TP311.5

中国版本图书馆CIP数据核字 (2007) 第031335号

北京版权局著作权合同登记

图字: 10-2005-5994号

©2004 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Electric Power Press, 2007. Authorized translation of the 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。

简体中文版由中国电力出版社出版, 2007。英文原版的翻译得到 O'Reilly Media, Inc.的授权。此简体中文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/	Head First设计模式 (中文版)
书 号/	ISBN 978-7-5083-5393-7
责任编辑/	胡顺增, 张旻
封面设计/	Ellie Volckhausen, 张健
出版发行/	中国电力出版社 (www.infopower.com.cn)
地 址/	北京三里河路6号 (邮政编码100044)
印 刷/	北京盛通印刷股份有限公司
开 本/	787毫米×980毫米 16开本 42.5印张 928千字
版 次/	2007年9月第1版 2007年9月第1次印刷
印 数/	0001—5000册
定 价/	98.00元 (册)

敬告读者

本书封面贴有防伪标签, 加热后中心图案消失。

本书如有印装质量问题, 我社发行部负责退换。

版权所有 翻印必究

O'Reilly Media, Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构 O'Reilly Media, Inc.授权中国电力出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media, Inc.是世界上在 Unix、X、Internet 和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的《The Whole Internet User's Guide & Catalog》（被纽约公共图书馆评为二十世纪最重要的 50 本书之一）到 GNN（最早的 Internet 门户和商业网站），再到 WebSite（第一个桌面 PC 的 Web 服务器软件），O'Reilly Media, Inc.一直处于 Internet 发展的最前沿。

许多书店的反馈表明，O'Reilly Media, Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media, Inc.具有深厚的计算机专业背景，这使得 O'Reilly Media, Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media, Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media, Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media, Inc.依靠他们及时地推出图书。因为 O'Reilly Media, Inc.紧密地与计算机业界联系着，所以 O'Reilly Media, Inc.知道市场上真正需要什么图书。

译者序

设计模式 (Design Pattern) 很重要, 不需要我多说。你瞧, 程序员几乎人手一本四人组 (Erich Gamma、Richard Helm、Ralph Johnson、John Vlissides) 所著的《设计模式》。打个比喻: 信耶稣的人都要读圣经, 而信OO的人都要读四人组的《设计模式》, 这就是OO的圣经。更有趣的是, 有人还不只买这本书的原版书、连它的光盘版、和中译本也一并买了收藏, 可见这是一本多么受到重视的书。我打探过这本书的销售量, 它畅销的程度令人咋舌。

许多人反映, 四人组的《设计模式》不容易阅读。对于不容易阅读的书, 会有已经悟道的人写出白话版或注释版, 以飨后进。所以圣经和佛经都有注释版, 用更白的方式阐述其中的道理, 而我认为《Head First 设计模式》也是因应这样的需求而产生, 它可以被视为是白话版、搞笑版、漫画版的《设计模式》。《Head First 设计模式》比起《设计模式》好读得多了, 内容也相当有趣。相信我, 要写出这样的一本书绝对比写一本正儿八经的书难上许多, 可见作者煞费苦心。作者的用心换来空前的成功。《Head First 设计模式》得到相当正面的读者响应, 连《设计模式》原创者Erich Gamma也慨然为《Head First 设计模式》写一段推荐文来“作保证”。《Head First 设计模式》还得到2005年的Jolt Award大奖, 风光至极。

本书大纲

本书共有14章, 每章都介绍了几个设计模式, 完整地涵盖了四人组版本全部23个设计模式。前言先介绍这本书的用法; 第1章到第11章陆续介绍的设计模式为Strategy、Observer、Decorator、Abstract Factory、Factory Method、Singleton、Command、Adapter、Facade、Template Method、Iterator、Composite、State、Proxy。最后三章比较特别。第12章介绍如何将两个以上的设计模式结合起来成为新的设计模式 (例如著名的MVC模式), 作者称其为复合设计模式 (这是作者自创的名称, 并非四人组的标准名词), 第13章介绍如何进一步学习设计模式, 如何发觉新的设计模式等主题, 至于第14章则很快地浏览尚未介绍的设计模式, 包括Bridge、Builder、Chain of Responsibility、Flyweight、Interpreter、Mediator、Memento、Prototype、Visitor。

第1章还介绍了四个OO基本概念 (抽象、封装、继承、多态), 而第1章到第9章也陆续介绍了九个OO原则 (Principle)。千万不要轻视这些OO原则, 因为每个设计模式背后都包含了几个OO原则的概念。很多时候, 在设计时有两难的情况, 这时候我们必须回归到OO原则, 以方便判断取舍。可以这么说: OO原则是我们的目标, 而设计模式是我们的做法。

本书特色

强大的写作阵容。本书作者Eric Freeman和Elisabeth Freeman是作家、讲师和技术顾问。Eric拥有耶鲁大学的计算机科学博士学位，Elisabeth拥有耶鲁大学的计算机科学硕士学位。Kathy Sierra (javarach.com的创始人)和Bert Bates是畅销的Head First系列书籍的创立者，也是Sun公司Java开发人员认证考试的开发者。

本书的产品设计应用神经生物学、认知科学，以及学习理论，这使得这本书能够将这些知识深深地印在你的脑海里，不容易被遗忘。本书的编写方式采用引导式教学，不直接告诉你该怎么做，而是利用故事当作引子，带领读者思考并想办法解决问题。解决问题的过程中又会产生一些新的问题，再继续思考、继续解决问题，这样可以加深体会。作者以大量的生活化故事当背景，例如第1章是鸭子，第2章是气象站，第3章是咖啡店，书中搭配大量的插图（几乎每一页都有图），所以阅读起来生动有趣，不会感觉到昏昏欲睡。作者还利用歪歪斜斜的手写字体，增加“现场感”。精心设计许多爆笑的对白，让学习过程不会太枯燥。还有模式告白节目，将设计模式拟人化成节目来宾，畅谈其内在的一切。

本书大量采用UML的Class Diagram (Static Structure Diagram)。书中的例子程序虽然都是用Java编写，但是本书所介绍的内容对于任何OO语言的用户都适用，包括C++和C#。每一章都有数目不等的测验题。每章最后有一页要点整理，这也是精华所在，我都是利用这一页做复习。

我认为，这本书的作者全都是“变态”！唔，我是说，好的那种“变态”。毕竟要把这么枯燥的主题写得这么有趣而学习效果又好，不是“变态”的作者还真是做不到呢！

《Head First设计模式》的作者/开发者

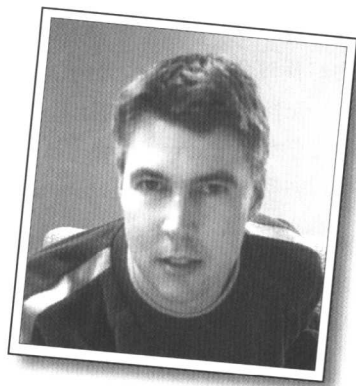
Elisabeth Freeman



Elisabeth是作者、软件开发人员及数字艺术家。她很早就开始进行Internet相关的研究，也是Ada Project的共同发起人（Ada Project是一个针对在计算机界工作的女性而设计的网站，曾获得大奖，现在已经并入ACM）。最近她带领迪士尼的数字媒体研发力量与他人共同发明了一个名为Motion的内容系统，此系统每天传送巨量的数字内容给迪士尼、ESPN及Movies.com的用户。

Elisabeth本质上是一个计算机科学家，拥有耶鲁大学和印第安那大学的计算机科学硕士学位。她的工作领域很广，包括视觉语言、RSS内容整合与Internet系统。她也很积极提倡女性从事计算机工作。今天，你可以发现她在她的Mac上使用Java或Cocoa，但是其实，她最希望的是全世界都使用Scheme。从小在苏格兰长大，Elisabeth喜欢在大自然踏青及户外活动。一旦她在户外，相机总是不离手。她热爱骑单车，是个素食主义者，也很喜欢动物。

她的电子邮件信箱是beth@wickedlysmart.com，你可以发电子邮件给她。



Eric Freeman

Eric是一个计算机科学家，热衷于软件架构和媒体。

他刚刚花了四年的时间在一个梦寐以求的工作上：在迪士尼指导Internet宽带与无线应用。现在，他回到写作的岗位上，用Java和Mac创造很酷的软件。

在90年代，Eric和David Gelernter一起花了大量的时间，寻找Desktop metaphor的替代品。（他们“仍然”在问：我干嘛不得不给计算机文件取个名字）。也因为这样的研究，Eric在1997年获得耶鲁大学的博士学位。他也与他人一同创立了Mirror Worlds Technologies公司（已经被收购），将他的论文内容商业化，创建了一套软件Lifestreams。

以前，Eric为网络和超级计算机写软件，你可以通过《JavaSpaces Principles Patterns and Practice》这本书得知他的名号。他曾在Thinking Machine CM-5上实现了元组空间系统（tuple-space system），也在80年代末期为NASA创建了第一个Internet信息系统，他为此深感自豪。

Eric目前住在圣达菲附近的沙漠中，当他不用写书或代码时，他总是花更多时间摆弄他的家庭影院，而不是观看影片，他利用空档时间试着修复80年代的经典视频游戏Dragon Lair。他也不介意在晚上兼差当个电音DJ。

给他的E-mail可以写到eric@wickedlysmart.com，你也可以去参观他的Blog，网址在<http://www.ericfreeman.com>。

Head First系列的创立者（以及本书共同策划者）

Kathy Sierra



Kathy自从开始设计游戏以来（她为Virgin、MGM、Amblin等都编写过游戏），一直对学习理论很感兴趣。Head First系列的大多数格式都出自她的手，具体来说，都是她在为UCLA Extension（加利福尼亚大学洛杉矶分校）的“Entertainment Studies”研究项目教授“New Media Authoring”（新媒体创作）课程时完成的。最近，她成为Sun公司的一名高级培训人员，负责教Sun的Java讲师如何讲授最新的Java技术，并参与开发了多个Sun的认证考试，其中就包括SCBCD考试。与Bert Bates一道，她积极地使用Head First概念来教成千上万的开发人员。她还是世界上最大的Java群体网站javaranch.com的创始人之一，这家网站赢得了2003年和2004年《软件开发》杂志生产力大奖。有时你还会看到她在Java Jam Geek Cruise（geekcruises.com）给学生上Java认证课程。

她最近从加州搬到了科罗拉多，在这里，她得学习一些新的词汇，包括“刨冰机”、“羊绒大衣”（译注），但是在这里的字典里找不到闪电两个字。

喜欢的事：跑步、滑雪、滑板、和她养的冰岛马玩、以及怪力乱神的玩意儿。不喜欢：Entropy（混乱）。

你可以在javaranch.com找到她，偶而她也会出现在java.net的blog中。写给她的信可以寄到kathy@wickedlysmart.com。

译注：加州会打雷，科罗拉多州会下雪。



Bert Bates

Bert很早就是一位软件开发者和建构师，不过由于在人工智能领域有近十年的经历，使得他对学习理论和基于技术的培训发生了兴趣。从那以后，他一直在教客户学习编程。最近，他成为Sun的Java认证考试开发小组的一员。

在他软件生涯的最初十年，他全世界游历，向Radio New Zealand、Weather Channel和Arts&Entertainment Network(A&E)这样一些客户提供帮助。他最得意的项目是为Union Pacific Railroad构建了一个全轨系统仿真应用。

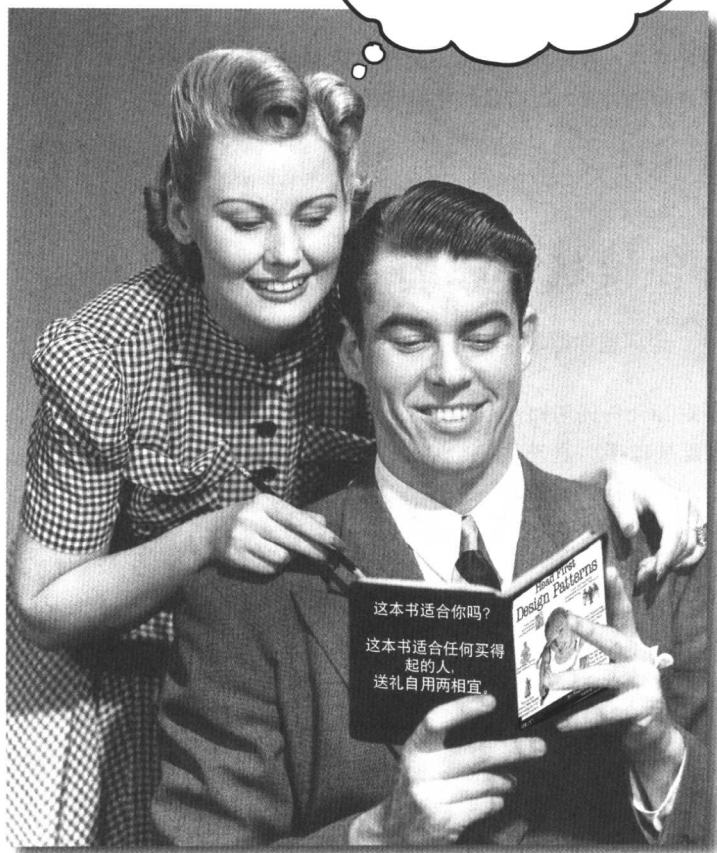
长久以来，Bert一直是无可救药的围棋玩家，玩围棋的时间已经长得超乎想象。他的吉他弹得不错，现在更意图染指Banjo（五弦琴或称斑鸠琴）。

你可以在Javaranch.com找到他，或者在IGS go Server上找到他。你也可以通过terrapi@wickedlysmart.com给他写信。

如何使用这本书

引子

真是无法相信，这样一些东西也能放在一本设计模式书里！



有一个问题真是听得我们耳朵都磨出茧了，这就是：“你们为什么要将这样一些东西放在一本设计模式书里呢？”这一节正是回答这个问题。

谁适合读这本书？

如果对下面的所有问题你都能肯定地回答“是”：

- ① 你懂Java吗？（不过不要求精通。）
- ② 你想学习、了解、记得并应用设计模式，以及其所基于的OO设计原则吗？
- ③ 你是不是更喜欢一种轻松的氛围，就像在餐桌上交谈一样，而不愿意被动地听技术报告似的枯燥乏味的说教？

那么，本书正是你需要的。

如果你会C#可能也可以。

谁暂时还不适合读这本书？

如果对下面任何一个问题你能回答“是”：

- ① 你是不是对Java一无所知？
（你不需要是高手，甚至你只会C#但不会Java也没关系，因为两者的相似度是80%。如果你只有C++背景，其实也应该没关系。）
- ② 你是不是一个很棒的OO设计者/开发人员，正在找一本参考书？
- ③ 你是不是一个架构师，想找企业设计模式？
- ④ 你是不是对新鲜事物都畏手畏脚？你是不是宁愿接受牙根管治疗，也不愿意接受苏格兰花格裙？你是不是觉得，如果把Java组件都拟人化了，这样的一本书肯定不是一本正儿八经的技术书？

那么，太遗憾了，本书不适合你。

[营销备注：本书适合所有有信用卡的人。]



我们知道你在想什么

“这算一本正儿八经的编程书吗？”

“这一堆图是干什么的？”

“我真的能这样学吗？”

我们也知道你的大脑在想什么。

你的大脑总是渴求一些新奇的东西，它一直在搜寻、审视、期待着不寻常的事情发生。大脑的构造就是如此。正是这一点才让我们不至于固步自封，能跟着时代前进。

如今，一般是不太可能被老虎吃掉的。然而，你的大脑还是一直在注意着周围是否有潜伏的老虎。只不过你自己没有意识到而已。但是我们每天都会遇到许多按部就班的事情，这些事情很普通，对于这样一些例行的事情或者平常的东西，你的大脑又是怎么处理的呢？它的做法很简单，就是不让这些平常的东西妨碍大脑真正的工作。那么什么是大脑真正的工作呢？这就是记住那些确实重要的事情。它不会费心地去记乏味的东西；就好像大脑里有一个筛子，这个筛子会筛掉“显然不重要”的东西，如果遇到的事情枯燥乏味，这些东西就无法通过这个筛子。

那么你的大脑怎么知道到底哪些东西重要呢？打个比方，假如你某一天外出旅行，突然一只大老虎跳到你面前，此时此刻，你的大脑里会发生什么呢？

看到这只大老虎，你的神经元会“点火”，情绪爆发，释放出一些化学物质。

好了，这样你的大脑就会知道……

这肯定很重要！可不能忘记了！

不过，假如你正待在家里或者坐在图书馆里，这里很安全，很温暖，肯定没有老虎。你正在刻苦学习，准备应付考试。也可能想学一些比较难的技术，你的老板认为掌握这种技术需要一周时间，最多不超过十天。这就存在一个问题。你的大脑很想给你帮忙。它会努力地把这些显然不太重要的内容赶走，保证这些东西不去侵占本不算充足的脑力资源。这些资源最好还是用来记住确实重要的事情，比如大老虎，再比如火灾险情。如果你曾经只身着短裤被大雪围困，这件事肯定不会忘却，你的大脑会记住绝不要让这种情况再发生第二次。

我们没有一种简单的方法来告诉大脑：“嘿，大脑，真是谢谢你了，不过不管这本书多没意思，也不管我对它是多么的无动于衷，但我确实希望你能帮助我把这些东西记下来。”

你的大脑想着：这真的很重要。



唉，又是637页没意思的文字，又枯燥又乏味。

你的大脑认为，这些根本不值得去记。



我们认为“Head First”的读者就是要学习的人

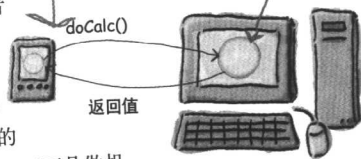
那么，怎么来学习呢？首先，必须了解，然后要保证自己确实不会忘记。这可不是填鸭式的硬塞。根据认知科学、神经生物学和教育心理学的最新研究，学习的途径相当丰富，绝非只是通过书本上的文字。我们很清楚怎么让你的大脑兴奋起来。

下面是一些Head First学习原则：

看得到。与单纯的文字相比，图片更能让人记得住，通过图片，学习效率会更高（对于记忆和传递型的学习，甚至能有多达89%的效率提升）。而且图片更能让人看懂。以往总是把图片放在一页的最下面，甚至放在另外的一页上，与此不同，如果把文字放在与之相关的图片内部，或者在图片的周围写上相关文字，学习者的能力就能得到多至两倍的提高，从而能更好地解决有关的问题。

需要调用服务
器上的一个方法。

RMI远程服务



采用一种针对个人的交谈式风格。最新的研究表明，如果学习过程中采用一种第一人称的交谈方式直接向读者讲述有关内容，而不是用一种干巴巴的语调介绍，学生在学习之后的考试中成绩会提高40%。正确的做法是讲故事，而不是做报告。要用通俗的语言。另外不要太严肃。如果你面对着这样两个人，一个是你在餐会上结识的很有意思的朋友，而另一个学究气十足，喋喋不休地对你指教，在这两个人中，你会更注意哪一个呢？

抽象方法真是简单。这些方法是没有身体的。



`abstract void roam();`

没有方法体（方法内的定义）！用一个分号结束。

让学习的人想得更深。换句话说，除非你很积极地让神经元活动起来，否则你的头脑里什么也不会发生。必须引起读者的好奇，促进、要求并鼓励读者去解决问题、得出结论、产生新的知识。为此，需要提出挑战，留下练习题和拓宽思路的问题，并要求读者完成一些实践活动，让左右脑都动起来，而且要利用多种思维。

说浴缸是一个（IS-A）浴室合适吗？能不能说浴室是一个浴缸？或者，是不是应该说这是一种“有一个”（HASA）关系？



引起读者的注意，而且要让他一直保持注意。我们可能都有这样的体验，“我真的想把这个学会，不过看过一页后就变得昏昏欲睡了”。你的大脑注意的是那些不一般、有意思、有些奇怪、抢眼的、意料之外的东西。学习一项有难度的新技术并不一定枯燥。如果学习过程不乏味，你的大脑很快就能学会。

影响读者的情绪。现在我们知道了，记忆能力很大程度上取决于所记的内容对我们的情绪有怎样的影响。如果是你关心的东西，就肯定记得住。如果你让你感受到了什么，这些东西就会留在你的脑海中。不过，我们所说的可不是什么关于男孩与狗的伤心故事。这里所说的情绪是惊讶、好奇、觉得有趣。想知道“什么……”，还有就是一种自豪感，如果你解决了一个难题，学会了所有人都觉得很难的东西，或者发现你了解的一些知识竟是那些自以为无所不能的傲慢家伙所不知道的，此时就会有一种自豪感油然而生。



元认知：有关思考的思考

如果你是真的想学，而且想学得更快、更深入，就应该注意你怎样才能集中注意力。考虑自己是怎样思考的，并了解自己的学习方法。

我们中间大多数人长这么大可能都没有上过有关元认知或学习理论的课程。我们想学习，但是很少有人教我们怎么来学习。

不过，这里可以做一个假设，如果你手上有这本书，你想学设计模式，而且可能不想花太多时间。另外，因为你要参加考试，所以需要记住你读到的所有内容。为此必须理解这些内容。想要最大程度地掌握这本书或其他任何一本书中介绍的知识，就要让你的大脑负起责任来，要求它记住这些内容。

怎么做到呢？技巧就在于要让你的大脑认为你在学习的新东西确实很重要，对你的生活有很大影响。就像老虎出现在面前一样。如若不然，你将陷入旷日持久的拉锯战中，虽然你很想记住所学的新内容，但是你的大脑却会竭尽全力地把它们拒之门外。

那么，究竟怎样才能让你的大脑把设计模式看作是一只饥饿的老虎呢？

这两条路：一条比较慢，很乏味；另一条路不仅更快，还更有效。慢方法就是大量地重复。你肯定知道，如果反反复复地看到同一个东西，即使再没有意思，你也能学会并记住它。如果做了足够的重复，你的大脑就会说“尽管看上去这对他说好像不重要，不过，既然他这样一而再、再而三地看同一个东西，那么我就假定这是很重要的。”

更快的方法是尽一切可能让大脑活动起来，特别是开动大脑来完成不同类型的活动。如何做到这一点呢？上一页列出的学习原则正是一些主要的可取做法，而且经证实，它们确实有助于让你的大脑全力以赴。例如，研究表明，把文字放在所描述图片的中间（而不是放在这一页的别处，比如作为标题，或者放在正文中），这样会让你的大脑更多地考虑这些文字与图片之间有什么关系，而这就会让更多的神经元点火。让更多的神经元点火=你的大脑更有可能认为这些内容值得注意，而且很可能需要记下来。

交谈式风格也很有帮助，当人们意识到自己正在与“别人”交谈，往往会更加关注，这是因为他们总想跟上谈话的思路，并能做出适当的发言。让人惊奇的是，大脑并不关心“交谈”的对方究竟是谁，即使你只是与一本书“交谈”，它也不会不在乎！另一方面，如果写作风格很正式，干巴巴的，你的大脑就会觉得像坐在一群人当中被动地听人做报告一样，很没意思，所以不必在意对方说的是什么，甚至可以打瞌睡。

不过，图片和交谈风格还只是开始而已，能做的还有很多。

我想知道
怎样才能骗过我的
大脑，让它记住这
些东西……



我们是这么做的：

我们用了很多图，因为你的大脑更能接受看得见的东西，而不是纯文字。对你的大脑而言，一幅图顶得上1024个字。如果既有图片又有文字，我们会把文字放在图片当中，因为文字处在所描述的图片中间时，大脑的工作效率更高，倘佯把这些描述文字作为标题，或者“湮没”在别处的大段文字中，那就达不到这种效果了。

我们采用了重复手法，会用不同的方式，采用不同类型的媒体、运用多种思维手段来介绍同一个东西，目的是让有关内容更有可能储存在你的大脑中，而且能够在多个区中都有容身之地。

我们会用你想不到的方式运用概念和图片，因为你的大脑喜欢新鲜玩艺；在提供图和思想时，至少会含着一些情绪因素，因为如果能产生情绪反应，你的大脑就会投入更大的注意。而这会让你感觉到这些东西更有可能要被记住，其实这种感觉可能只是有点幽默，让人奇怪或者比较感兴趣而已。

我们采用了一种针对个人的交谈式风格，因为当你的大脑认为你在参与一个交谈，而不是被动地听一场演示汇报时，它就会更加关注。即使你实际上在读一本书，也就是说在与书“交谈”，而不是真正与人交谈，但这对你的大脑来说并没有什么分别。

在这本书里，我们加入了40多个实践活动，因为与单纯的阅读相比，如果能实际做点什么，你的大脑会更乐于学习，更愿意去记。练习都是我们精心设计的，有一定的难度，但是确实能做出来，因为这是大多数人所希望的。

我们采用了多种学习模式，因为尽管你可能想循序渐进地学习，但是其他人可能希望先对整体有一个全面认识，另外可能还有人只是想看看一个代码示例。不过，不管你想怎么学，要是同样的内容能以多种方式来表述，这对每一个人都会有好处。

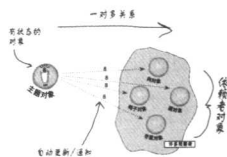
这里的内容不只是单单涉及左脑，也不只是让右脑有所动作，我们会让你的左右脑都动起来，因为你的大脑参与得越多，你就越有可能学会并记住，而且能更长时间地保持注意力。如果只有一大脑在工作，通常意味着另一半有机会休息，这样你就能更有效率地学习更长时间。

我们会讲故事，留练习，从多种不同的角度来看同一个问题，因为如果要求大脑做一些评价和断，它就能更深入地学习。

你会看到我们给出的一些练习，还要回答一些问题，这些问题往往不是直截了当就能做出回答的，通过克服这些挑战，你就能学得更好，因为让大脑真正做点什么的话，它就能学会并记住。想想吧，如果只是在健身房里看着别人流汗，这对于保持你自己的体形肯定不会有什么帮助，正所谓临渊羡鱼，不如退而结网。不过另一方面，我们会竭尽所能不让你钻牛角尖，把劲用错了地方，而是能把功夫用在点子上。也就是说，你不会为搞定一个难懂的例子而耽搁，也不会花太多时间去弄明白一段晦涩难懂而且通篇行话的文字，我们的描述也不会太过简洁而让人无从下手。

我们用了拟人手法。在故事中，在示例中，还有在图中，你都会看到人的出现。这是因为你本身是一个人，不错，这就是原因。如果和人打交道，相对于东西而言，你的大脑会表示出更多的注意。

我们充分利用了80/20方法，我们认为，如果你真的要攻读软件设计博士的话，这本书肯定不是你唯一的设计模式书，所以我们不打算面面俱到。这里只提供了你真正需要的东西。



模式大师

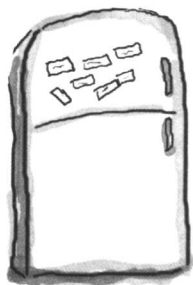


要点



谜题





可以用下面的方法让你的大脑就范

沿着虚线剪下，
贴在水箱上。

好了，我们该做的已经做了，剩下的就要看你自己的了。这些提示只是个开头：听一听你的大脑是怎么说的，弄清楚对你来说哪些做法可行，哪些做法不能奏效。还可以做些新的尝试。

① 慢一点，你理解的越多，需要记的就越少。

不要光是看看而已。停下来，好好想一想。书中提出问题的時候，你不要直接去翻答案。可以假想成真的有人在问你问题。你让大脑想得越深，就越有可能学会并记住。

② 勤做练习，自己记笔记。

我们给你留了练习，但是如果这些练习的解答也由我们一手包办，那和有人替你参加考试有什么区别？不要只是坐在那里看着练习发呆。拿出笔来，写一写、画一画。大量研究都证实，学习过程中如果能实际动手，将改善你的学习效果。

③ 阅读“*There are no Dumb Questions*”部分。

顾名思义，这些问题可不是可有可无的旁注，它们绝对是核心内容的一部分！千万不要把它们跳过去不看。

④ 上床睡觉之前不要再看别的书了，或者至少不再看其他有难度的东西。

学习中有一部分是在你合上书之后完成的（特别是，要把学到的知识长久地记住，这往往无法在看书的过程中做到）。你的大脑也需要有自己的时间来再做一些处理。如果在这段处理时间内你又往大脑里灌输了新的知识，那么你刚学的一些东西就会被丢掉。

⑤ 要喝水，而且要多喝点水。

如果能提供充足的液体，你的大脑才能有最佳表现。如果缺水（可能你觉到口渴之前，就已经缺水了），学习能力就会下降。

⑥ 大声说出来。

说话可以刺激大脑的另一部分。如果你想看懂什么，或者想更牢地记住它，就要大声说出来。更好的办法是，大声地解释给别人听。这样你会学得更快，而且可能会有有一些新的认识，而这是以前光看不说的时候未曾发现的。

⑦ 听听你的大脑怎么说。

注意一下你的大脑是不是负荷太重了，如果发现自己开始浮光掠影地翻看，或者刚看的东西就忘记了，这说明你该休息一会儿了。达到某个临界点时，如果还一味地向大脑里塞，这对加快学习速度根本没有帮助，甚至还可能影响正常的学习。

⑧ 要有点感觉！

你的大脑需要知道这是很重要的东西。要真正融入到书中的故事里。为书里照片加上你自己的说明。你可能觉得一个笑话很憋脚，不太让人满意，但这总比根本无动于衷要好。

⑨ 设计一些东西！

将学来的知识应用到新项目中，甚至重构旧项目。反正就是尽量应用知识，获取实践经验。你所需要的是一枝铅笔和一个难题，试着应用数个设计模式解决这个难题。

Readme

这是一本体验式学习的书，不是一本参考书。对于学习过程有所阻挠的东西，我们都予以排除。读完第一次之后，你需要从头再读一次，因为本书对读者的背景知识做了一些假设。

我们使用简单的“类”UML图（注意，可不是UML类图，而是指与UML图很相似）。

书中用到了UML，但是我们没有详细介绍UML，而UML也不是本书必备的预备知识。如果你以前没见过UML，也别担心。我们会沿路告诉你一些UML的基本用法。换句话说，你根本不需要同时担心UML和设计模式。我们的图示法是“类”UML图——虽然我们试着用真正的UML，但是基于自私的写作必要，我们终究还是做了一些小改变。

我们没有包含所有的设计模式。

设计模式实在是太多了，GoF的基础模式、Sun的J2EE模式、JSP模式、架构模式、游戏设计模式……我们希望这本书的重量能比读者的体重更轻，所以自然不可能涵盖所有的设计模式。我们从GoF模式中，取出更重要的一部分模式，作为本书的焦点，并确保你能够真正地、深入地、彻底地了解如何使用这些模式，以及何时使用这些模式。对于GoF的其他模式，我们也会在附录中概略地介绍。我们相信，读过本书之后，你可以很快地从其他资源中学到本书没有介绍的模式，并且游刃有余。

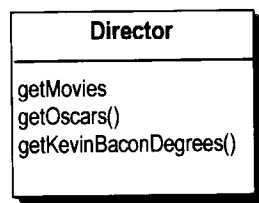
书里的实践活动不是可有可无的。

这里的练习和实践活动并非可有可无的装饰和摆设；它们也是这本书核心内容的一部分。其中有些练习和活动有助于记忆，有些则能够帮助你理解，还有一些对于如何应用你所学的知识很有帮助。所以，请不要略过这些练习。填字游戏是你唯一可以不理会的部分，但是它们可以帮助大脑回想本章的内容。

当我们提到“组合”（composition）一词，我们指的是OO一般概念中的composition，而不是UML严格定义的composition。

当我们说“一个对象和另一个对象组合在一起”，我们的意思是“有一个”（HAS-A）的关系。在一般的OO概念及GoF的书中，都是采用这样的用法。最近UML对于composition有严谨的定义，如果你是UML专家，你还是可以读这本书，只是要注意到此名词定义上的差异。

我们对UML有所修改，使用了一种更简单的“类”UML。



我们有意安排了许多重复，这些重复非常重要。

Head First系列图书有一个与众不同的地方，这就是，我们希望你确实实地掌握这些知识。另外，我们希望在学完这本书之后你能记住学过了什么。尽管重复很有必要，不过，多数参考书都不认为重复和回顾是一个重要的环节，但是在这本书里，你会看到一些概念会一而再、再而三地出现很多次。

代码示例尽可能短小精悍。

有读者告诉我们，如果查了200行代码才能找到要理解的那两行代码，这是很让人郁闷的。这本书里大多数示例往往都开门见山，作为上下文的代码会尽可能地少，这样你就能一目了然地看到哪些东西是需要你学习的。别指望这些代码很健壮，要知道这里的代码甚至是不完整的——毕竟我们的代码是辅助学习之用，所以不见得一定功能完整。

在某些例子中，我们并未将所有需要的package都import进来，但如果你是Java程序员，你应该知道ArrayList类是属于java.util package。如果package不属于J2SE API，我们会特别说明。我们已经将所有的代码都放在网络上，可供下载。网址在：<http://wickedlysmart.com/headfirstdesignpatterns/code.html>。

为了方便学习与测试程序，我们在书中并没有将我们的类放在package中（换句话说，所有的类都是在Java默认的package中）。我们不建议你真实世界中也这么做。如果你到我们的网站下载代码，会发现这些类都放在适当的package中。

“Brain Power”习题没有答案。

对于某些人来说，“Brain Power”习题没有对的答案；对于另一些人来说，动动脑习题所带来的学习经验在于决定是否你的答案是对的，以及何时你的答案是对的。在某些动动脑习题中，我们会提供暗示，为你指引正确的方向。