

# Design Patterns



[Design Patterns 下载链接1](#)

著者:Erich Gamma

出版者:Addison Wesley

出版时间:1994-10-31

装帧:Hardcover

isbn:9780201633610

\* Capturing a wealth of experience about the design of object-oriented software, four top-notch designers present a catalog of simple and succinct solutions to commonly occurring design problems. Previously undocumented, these 23 patterns allow designers to create more flexible, elegant, and ultimately reusable designs without

having to rediscover the design solutions themselves. \* The authors begin by describing what patterns are and how they can help you design object-oriented software. They then go on to systematically name, explain, evaluate, and catalog recurring designs in object-oriented systems. With Design Patterns as your guide, you will learn how these important patterns fit into the software development process, and how you can leverage them to solve your own design problems most efficiently.

作者介绍:

四位作者均是国际公认的面向对象软件领域的专家。

Erich Gamma博士是瑞士苏黎士国际面向对象技术软件中心的技术主管。

Richard Helm博士是澳大利亚悉尼IBM顾问集团公司面向对象技术公司的成员。

Ralph Johnson博士是Urbana-Champaign伊利诺大学计算机科学系成员。

John Vlissides博士是位于纽约Hawthorne的IBN托马斯J.沃森研究中心的研究人员。

目录:

[Design Patterns 下载链接1](#)

标签

设计模式

计算机

软件工程

Programming

design

软件开发

编程

经典

## 评论

虽然只是挑了几个章节来读，但已经确定这是读过的关于设计模式的书籍中相当好的一本。不像外国的头大系列，也不像国内大话系列，那些都太浮夸了。这本就是那么的简明・实用！

我們當年都叫它《輪子大全》，全方位地告訴你怎麼造輪子……用做一個文字編輯器為例，系統講述了為什麼要造輪子，怎麼造輪子。全書寫作語言非常學術，旁證博引，外加很多不知道的計算機遠古時期的知識。力推。

非常好, 读了两遍以上。学习面向对象，这本是必读书。另外就是敏捷软件开发。

An amazingly handy handbook for conceptualizing and developing design patterns.

如果只接触过脚本语言如我这本书没多大实际意义（实际意义＝能现学现用，对目前工作有直接启发），但不能不说这本书影响太深远了，以至于JS里的模式（如Addy Osmani那本）都几乎完全照搬其定义，只是觉得甚为勉强。有启发，但很可能是过度解读扭曲了的启发

设计模式的经典～

The gang of four elevates the art of tools to a magnificent philosophical level.

需要经常consult的书籍

-----  
Currently studying the Decorator pattern (p. 179). And Decorator vs. Strategy pattern.

-----  
简明扼要

-----  
经典中经典。有些pattern还是太简单了，但作为程序的building block...还是有用的。

-----  
写得挺清楚的，但是感觉没什么收获。大部分的pattern不是自然而然就想出来了么

-----  
Classic!

-----  
in a word: design for CHANGES, 封装变化点

-----  
项目需要考虑设计模式，很多设计模式能在实际代码中看到，但要读懂书中的分析，需要一定的项目经验和代码阅读量

-----  
比Head First那本高到不知道哪里去了

-----  
這是一個 problem-solution 的列表，如果你要解決的問題和書中提到的類似，不妨參考下它提出的解決方案。

-----  
有用的还是常见模式例子背后的抽象

-----

A must read for all software engineers. It did not make much sense when I read it at school, but made tons of sense now

软件开发模式其实都是自然产生的，不过了解一下设计语言也是不错的。

[Design Patterns\\_下载链接1\\_](#)

## 书评

我感觉看这本书，有几个层次，每个层次对应着软件开发方面的境界。  
第一个层次，看不懂。所谓模式，就是在长期写代码的过程中总结出来的一种经验。这在刚毕业或者工作一两年的软件开发的工程师中，很多就反应这本书很深奥，难看懂。一方面，书中的例子是C++写的，而且时间比较...

误解1：设计模式只有23种 GoF在前言中就说到 “We don’ t consider this collection of design patterns complete and static; it’ s more a recording of our current thoughts on design.” 这本设计模式领域的开山鼻祖，第一扩展了人们的面向对象设计思路，第二则启示人们用模...

第一次看，看了40页，实在枯燥，看不下去了。  
第二次看，是一年后，期间做了个项目，项目十分糟糕，偶尔又翻了这本书，感觉实在不错。这次总算看了大概。第三次看，又是一年后，这次当参考书来了，呵呵呵  
我的建议，一定要有不成功的开发、维护经历，这本书才能够让你深入理...

1.  
现在的计算机书已经厚到令人发指。比如这学期的《操作系统》教材，厚度堪比辞海。  
2.  
学术论文也一样。一个人写自己在巨人肩膀上看到的风景，一定要从自己爬巨人时看到的腿毛开始描写。  
3. 这本书254页，可是取其精华远多于那些学术垃圾。  
4.  
因为考试原因，两天看完，酣...

作者坦言，书中描述的23种模式并不是他们创造出来的，他们只是进行了总结和归纳。

这本书并不适于初学编程的人，甚至是没有相关知识背景的熟练程序员。因为其对概念的描述太过简洁和学术化，其C++的例子也不是很好懂--我自己就愣是没有看懂，后来还是通过另一本以Java为例子的...

-----  
如果你觉得这本书不知所云，那说明你其实不懂编程  
如果你觉得这本书对你裨益甚多，那说明你开始了解编程  
如果你觉得这本书不过如此，那你不是高手就是完全的外行

-----  
在支持函数式的动态语言里，绝大多数design pattern都变的简单直接，以至于你甚至感觉不到它们的存在。  
在面向对象的限制之下，敞开的大门不走，偏要爬窗户。比如strategy pattern的本意是通过composition而非inheritance，使能够在运行时(runtime) 动态绑定某对象的成员...

-----  
这就是书中的阳春白雪。当然，高人是一看就懂。  
前10遍，只懂几个模式；到20遍，似乎都懂了，但是放下书就不能自己构造一个出来；30遍后，至今没再看。书也看烂了，但是敝帚自珍。  
读者最好熟悉c++，以及一定的利用c++做开发的经历。否则，最好看别的类似的书。  
如果希望用o...

-----  
因为可以买到这本书的地方和版本这么多。所以这个质量纯粹是指的外在质量（内在质量不用我介绍了吧？其实我觉得设计模式读一两本书就够了，最重要的还是应用吧）。不知道为什么，电力出版社出版的一套开本和印刷都让人读起来挺舒服的深蓝色封面的特辑（原版风暴系列）里没有它(...

-----  
经典就是经典，GoF在本书中提出的设计模式，至今仍被人津津乐道。很多人自称高手，自以为是的那些所谓高手，基本的一条就是懂得若干设计模式。很多人说这本书很枯燥，虽然GoF的思想很好，但是书写的挺差劲，很容易让人昏昏欲睡。确实是的，如果不能潜下心来研读，只会看到很多不...

-----  
由李英军同志翻译的中文版我是无论如何读不懂。  
本来这本书GOF就是以学术专著的形式写成，所以晦涩啊，艰深啊，再加上中文翻译的

不准确性，就造就了该书中文版被芸芸众生顶礼膜拜为天书的壮观景象。不过还好，机械工业出版社自己都觉得问心有愧，遂直接出版『影印版』，于是...

-----  
这本书很适合，有一定编程经验的程序员来看，不太适合刚刚入门的程序员。设计模式有好多种，本书阐述了27种（我记得是）我们还可以自己总结一些其它模式，这样就更有意义了。

-----  
书绝对是好书，但是译者的翻译给理解本书的原本含义造成了巨大的障碍，对翻译的理解的障碍甚至大于对于设计模式本身理解的障碍。要读就读原版，否则可能越看越不懂。看不懂得时候，不一定是自己的智商低，多想想别人的智商。没有10万行代码勿读，少儿不宜。

-----  
但是发现现在所谓模式的滥用比用好的比率大得多。反复思考，到底OO和模式将给偶们带来什么？不能为了模式而模式。

-----  
软件行业从业者必读的经典，从我就业以来就一直想把这本书看一遍，但是之前一直在搞Linux开发，玩的是C语言。没有什么模式可言。直到近期我做点QT的开发，才有这样的机会。直接拿起这本书看觉得有点费劲，所以先看了《Head First设计模式》，并把书中的所有模式都用C++实现了...

-----  
TITLE： 设计模式——可复用面向对象软件的基础\\ ENGLISH TITLE: Design Patterns, Elements of Reusable Object-Oriented Software\\ AUTHOR： 软件界四人帮 (Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides)  
本书共讲了23个常用的设计模式。创建型模式 1. ...

-----  
花了一个月时间，把英文版大致过了一遍。由于有一些失败的编程经历，大致还是能够理解作者的用意的。个人喜欢先看motivation，structure和sample code，再回过头看讨论。有点遗憾的是书中对各个模式的横向对比做得不够。有些地方容易混淆，有些场合的名词太过空洞。或许只有...

-----  
这本书的翻译完全就是不怎么行，推荐一本英文版的《设计模式解释》，深入浅出，归纳出设计模式的本质就是“抽象变化”，其实万变不离其中

-----  
再一次翻阅——是翻阅 这本书很大的优点就在 当你第一次认真读完  
之后允许你用翻阅的方式迅速拎起渐渐忘去的内容 a nice refined book  
虽然初读时很觉枯燥——现在也是 书中的例子很多我都没看下去 但是 精炼  
都说《深入浅出设计模式》非常优秀，我相信。 几次动念头去...

-----  
[Design Patterns\\_下载链接1](#)