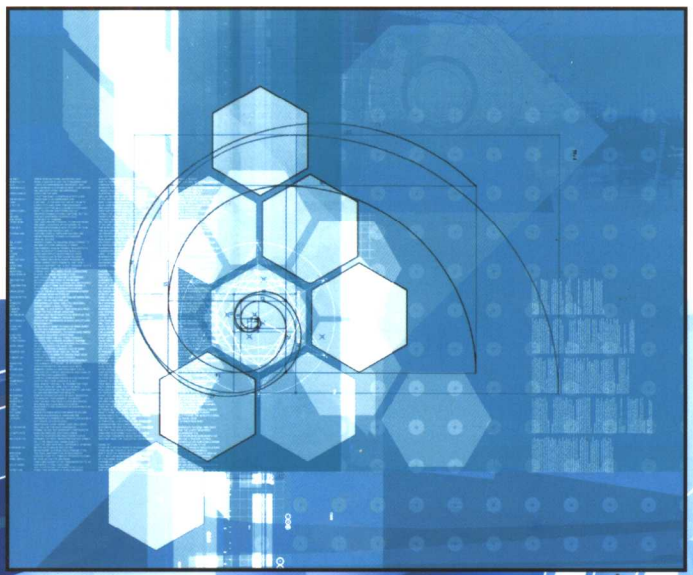
PEARSON
Addison Wesley

设计模式精解

Design Patterns Explained



(美) Alan Shalloway & James R.Trott 著
熊节 译

Pearson Education
清华大学出版社

设计模式精解

[美] Alan Shalloway & James R. Trott 著
熊节 译

清华大学出版社
北京

图书在版编目(CIP)数据

设计模式精解/(美)沙洛韦(Shalloway, A.),(美)特罗特(Trott, J. R.)著;熊节译. —北京:清华大学出版社, 2004.12
书名原文: Design Patterns Explained
ISBN 7-302-09841-7

Ⅰ. 设... Ⅱ. ①沙...②特...③熊... Ⅲ. 面向对象语言—程序设计 Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2004)第 111911 号

出 版 者: 清华大学出版社 **地 址**: 北京清华大学学研大厦
http://www.tup.com.cn **邮 编**: 100084
社总机: 010-62770175 **客户服务**: 010-62776969
责任编辑: 林庆嘉
封面设计: 立日新
印 刷 者: 北京四季青印刷厂
装 订 者: 三河市李旗庄少明装订厂
发 行 者: 新华书店总店北京发行所
开 本: 185×230 **印张**: 16.5 **字数**: 361 千字
版 次: 2004 年 12 月第 1 版 2005 年 2 月第 2 次印刷
书 号: ISBN 7-302-09841-7/TP·6786
印 数: 3001~5000
定 价: 35.00 元

译　序

Alan Shalloway 的《设计模式精解》是一本很好的新书，其中有很多很好的应用模式的例子，尤其适合模式的初学者。

——John Vlissides

这本书(《设计模式精解》)最大的优点之一就是：它用类比的方式将概念解释得非常清楚，而很少使用程序代码。我正在做一套关于 OOP 和软件开发的录音教材，这本书讲述概念的方式对我非常有启发。

——Bruce Eckel

模式

从去年 9 月开始翻译《设计模式精解》到现在，已有半年的时间。在这半年里，我很欣喜(同时也很讶异)地看到：在中国的软件开发者里面，有那么多的人爱好设计模式，有那么多的人关注着“模式”这个话题。模式在中国受重视的程度，远远超出了我当初的想象。

但是必须看到，随着“模式”这个词的日益流行，“人们对模式的混淆、惊惶和以讹传讹已经不是一点半点”。就像任何流行的东西一样，人们对模式也存在着各种各样的误解。有人说“模式就是某种编程技巧”，有人说“看了《设计模式》中的几个模式就足够了”，有人说“模式需要有某种工具或者方法学的支持才会有效”，有人说“模式可以保证软件的复用性”，还有一种听得最多的：“模式是针对面向对象设计和实现的”。喜欢模式的读者，你可曾意识到，这些都是对“模式”一词的误解呢?

还是模式

究竟什么是模式? 按照 Christopher Alexander 的定义，模式就是“在某一个场景下的问题解决方案”。但是，在 John Vlissides 看来，连这种观点都是片面的，他认为，除了场景、问题和解决方案之外，一个模式还必须有三个要点：

1. 可重复性。解决方案应该对应于外部的场景。
2. 可传授性。一个解决方案应该可以移植到问题的不同情况中(绝大多数模式的可传

授性都建立在“约束”和“效果”的基础上)。

3. 用来表示这个模式的名称。

而我们经常挂在嘴边的“模式”，常常就是指《设计模式》中的“设计模式”，或者再准确些讲，是“面向对象的设计模式”。顾名思义，它是在面向对象方法基础上发展起来的，它是用于设计阶段的，它是模式的子集，却不是模式的全部。**模式是针对特定场景下的特定问题的可重复、可表达的解决方案。**它不限于面向对象，不限于设计阶段，甚至不限于软件开发领域。这跟我们这本书没有太大关系，但是记住这一点是有必要的。

模式精解

《设计模式》早在 1995 年就面世了，中译本的出版也是在两三年以前。泰山北斗在前，还需要有这样一本设计模式的书籍吗？我说：需要。为什么会有那么多的误解？为什么会有那么多人抱怨“《设计模式》看不懂”？我想，因为这部经典太过精练又太重理论，再加上陌生的语言、失当的例子、过时的场景，使读者很难对其中的模式形成有感性的认识，更遑论理性的理解了。

请来想想这几个问题：首先，你真正了解面向对象吗？其次，你准确地知道什么是模式吗？最后，你知道应该如何使用设计模式吗？实际上，面对这三个问题，很少有人能够毫不犹豫地连答三个“Yes”。现在，你手上的这本书，就是为了“精解”这些疑惑的。用作者自己的话来说：

也许你使用一种面向对象或基于对象的语言已经有好几年了。但是，你是否已经知道，对象真正的威力不是继承而是“行为封装”？也许你对设计模式很感兴趣，并且发现关于设计模式的专著往往有点过于深奥。如果是这样，这本书便是为你准备的。

读过这本书之后，你将获得对十个最基本的设计模式的全面理解。你将明白：设计模式不是单独存在的，而是需要与其他设计模式协同工作以帮助你创建更健壮的应用程序。你将获得足够的基础以阅读设计模式的专著，并且——如果你愿意——可能发现新的模式。

这就是这本书存在的理由，也就是你读它的理由。

致谢

首先要感谢潘爱民老师。是他的一番教诲让我对模式产生了浓厚的兴趣，并最终决定翻译本书。

感谢 UMLChina 的站长潘家宇，他帮助我处理了本书前期的许多事务。他的自信一直感染着我，让我找到自我的价值。

感谢清华大学出版社的汤斌浩，他对我无休止的问题从来没有厌烦过。

感谢所有热心的读者，他们忍受了我拙劣的翻译，他们对本书的热切期待是我前进的动力。

特别感谢我亲爱的女友马姗姗。当我处于困境中的时候，当我心情沮丧的时候，是她一直支持、鼓励我，照顾我的生活。没有她，就不可能有本书的完成。

最后，感谢所有为本书的翻译、制作、出版、发行做出贡献的人们。

祝读者在学习模式的路上走好。

前　言

设计模式和面向对象的程序设计曾经承诺：让软件设计开发者的生活更加轻松。每一天，技术传媒乃至大众传媒都传播着它们的术语。但是，要真正学习它们、精通它们、懂得它们的用途，还是很困难的。

也许你使用一种面向对象或基于对象的语言已经有好几年了。但是，你是否已经知道，对象真正的威力不是继承而是“行为封装”？也许你对设计模式很感兴趣，并且发现关于设计模式的专著往往有点过于深奥。如果是这样，这本书便是为你准备的。

在多年向软件开发者——面向对象领域的老手和新手——传授设计模式的基础上，有了本书的诞生。我们相信——并且我们的经验也证明——如果你懂得了这些概念之下的原则和动机、明白了这些设计模式行为的理由，你的学习过程将会让人难以置信地缩短。并且从我们对设计模式的讨论之中，你将可以明白真正的面向对象思想；只有明白了这些，你才能真正成为一个专家。

读过本书之后，你将获得对十个最基本的设计模式的全面理解。你将明白：设计模式不是单独存在的，而是需要与其他设计模式协同工作以帮助你创建更健壮的应用程序。你将获得足够的基础以阅读设计模式的专著，并且——如果你愿意——可能发现新的模式。

最重要的是，你将变得更加训练有素，将能够创建更加灵活、更加完整、更容易维护的软件。

从面向对象到模式再到真正的面向对象

本书的很多地方都复述了我自己学习设计模式的经验。在学习设计模式之前，我认为自己理所当然是面向对象分析和设计的专家。我曾经为各种行业的客户做过一些还算给人深刻印象的设计和实现。我会使用C++并且已经开始学习Java。我编写的代码中对象格式优美、封装紧密。我可以在继承体系中设计优秀的数据抽象。我想我已经懂得面向对象了。

现在回头看看，我发现那时其实我还根本不知道面向对象设计的全部能力，尽管我一直按照专家建议的方式去做。直到开始学习设计模式，我的面向对象设计能力才得到了扩展和增强。学习设计模式使我成为了一个更好的设计者，甚至是在我还没有直接使用那些模式的时候。

我从1996年开始学习设计模式。当时我正在美国西北部一家大型航天公司担任C++/面向对象设计顾问。有几个人劝说我领导一个设计模式学习组。正是在那里我遇到了本书

的另一位作者 Jim Trott。在那个学习组中发生了几件有趣的事情。首先，我开始对设计模式着迷。我可以把自己的设计和其他更有经验的人的设计相比较，我爱上了这种感觉。另一方面，我发现自己并没有完全做到“针对接口进行设计”，也没有随时注意“一个对象是否可以在不知道另外对象的类型的情况下使用另外对象”。同时我还注意到，那些面向对象的初学者——通常认为他们现在开始学习设计模式是太早了——从这个学习组得到的收益与那些面向对象的专家不相上下。设计模式向学习者展现出优秀的面向对象设计实例并阐述基本的面向对象设计原则，而这些使学习者的设计更快地成熟起来。在整个学习进程结束之后，我确信：设计模式，继面向对象设计的发明之后，将是软件设计领域中最好的东西。

但是，看看那个时候我自己的工作，我发现我根本还没有在自己写的代码中结合任何一个设计模式。

当时我只是认为自己还没有学到足够的设计模式，还需要学习更多。那时候，我只知道六个设计模式。一个偶然的机会，我得到了顿悟。当时我在一个项目中担任面向对象设计顾问，并且要为这个项目做一个高层设计。这个项目的领导人极其聪明，但在面向对象设计领域，他还是一个新手。

这个问题本身并不困难，但需要非常注意确保代码容易维护。和往常一样，在看过问题两分钟之后，我便有了一个设计——采用了我常用的数据抽象的途径。很不幸，显然这不会是一个好的设计。单纯地采用数据抽象途径已经让我尝到过失败的滋味。我必须找到一些更好的设计思路。

两个小时过去了，在使用了我所知道的所有设计技术之后，情况仍然没有好转。我的设计基本上都还是和从前一样。而最让我感到受挫的是，我知道一定有一个更好的设计，但我就是找不到它。更具讽刺意味的是，我甚至还知道四个设计模式就“生活”在我的问题中，但我看不出应该如何使用它们。在这里，我，一个被认为是面向对象设计专家的人，被一个简单的问题困住了！

我实在觉得很难受，于是我停了下来，走出办公室以清醒头脑，并告诉自己：至少 10 分钟里我不再想这个问题。呵呵，30 秒之后，我又开始想它了！但我获得了一种领悟并完全改变了我对设计模式的看法：设计模式无法作为独立的条款使用；我应该把设计模式放在一起使用。

模式是应该被结合在一起来共同解决一个问题的。

以前我曾经听到过这句话，但那时我并没有真正理解它。因为软件开发中的模式往往被介绍为“设计模式”，所以我总是在“模式最主要的贡献是在设计阶段”的假设下努力。我的想法是：在设计世界里，模式就好像是类之间优美的联系。然后，我阅读了 Christopher Alexander 那本令人惊讶的书——*The Timeless Way of Building*。我学到了：模式存在于所有阶段——分析、设计乃至实现之中。Alexander 在书中讨论了如何使用模式来帮助理解(乃至描述)问题领域，而不是仅仅在理解了问题领域后使用模式来创建一个设计。

我的错误是：我尝试先创建问题领域中的类，然后将这些类缝合起来形成最终的系统——Alexander 把这样的过程称为“一个坏主意”。我从来没有问过自己：我是否拥有正确的类？仅仅因为这些类看起来如此正确、如此明显。我拥有的，是在开始分析时立刻进入了脑海里的类，是我们的老师告诉我们应该在系统的描述中寻找的“名词”。但是我的错误就是仅仅尝试把它们简单地放在一起。

当我回过头，开始使用设计模式和 Alexander 的方式来指导自己创建类时，仅仅几分钟之后，一个更优秀得多的解决方案在我的脑海中显露出来。这是一个很好的设计，于是我们把它应用在产品之中。我很兴奋——为我设计了一个好的解决方案，也为设计模式的威力而兴奋。从此，我开始在自己的开发工作和教学中结合设计模式。

我开始发现，那些刚开始学习面向对象设计的程序员也可以学习设计模式。并且他们可以在这个学习过程中为自己的面向对象设计能力打好基础。这对于我以及我的学生都是如此。

想象一下我的惊讶！我读过的设计模式书籍以及与我交谈过的设计模式专家都曾经告诉我：在开始学习设计模式之前，你真的需要认真进行面向对象设计的基础训练。然而，我所见到的，同时学习面向对象设计和设计模式的那些学生，他们掌握面向对象设计的进度比那些只学习面向对象设计的学生更快。甚至他们掌握设计模式的进度看上去几乎和那些有经验的面向对象实践者一样快。

我开始把设计模式用做我的教学基础。我开始把我的课程叫做“*面向模式设计：从分析到实现的设计模式*”。

我希望我的学生能理解这些模式，并且我发现使用一个探索的过程是帮助他们理解的最好办法。举个例子，我发现如果要向学生们讲解 Bridge 模式，我最好能向他们展示一个实际问题，然后让他们尝试为这个问题设计一个解决方案。我会给他们一些指导性的原则和策略——我发现大多数设计模式都指出了这些。经过这个探索过程，学生们最后找到了解决方案——被称为 Bridge 模式——并牢牢记住了它。

无论如何，我还发现这些指导性的原则和策略可以用来“推导”出好几个这样的设计模式。“推导出一个设计模式”，我说这句话的意思是：如果我看到一个问题并且知道可以用一个设计模式来解决这个问题，我就可以通过这些指导性的原则和策略来得到该模式所表达的解决方案。我向我的学生们明确指出，我们不会真的通过这种方法得到设计模式。我只是阐明一种可能的思考过程。模式的发现者通过这样的过程得到了最初的解决方案，并最终把解决方案归类成设计模式。

几句题外话

在我现在看来，这些指导原则及策略都非常清楚了。当然，它们在“四人团”的设计模式书中都有描述。但是，由于我自己对面向对象范式的理解有限，我花了很多时间来理解这些原则和策略。直到我在自己的思想中结合了四人团及Alexander的研

究成果、Jim Coplien 在共同点和变化点上的研究成果、Martin Fowler 在方法论和分析模式上的研究成果之后，这些原则对我才算足够清楚，我才能和他人谈起这些原则。这帮助我决定开始过一种为人解惑的生活，这样我不会过分轻易地假想自己的能力——当仅仅为自己工作时，我很容易产生这样的假想。

能力的提高可以帮助我更好地解释这几个很有威力的原则和策略。并且当我开始解释更多四人团的模式时，它们更加有用了。实际上，在设计模式课程中，我用这些原则和策略来解释 12 ~ 14 个模式。

我还发现，我开始在自己的设计中使用这些原则，不管是否使用设计模式。这并没有让我感到惊讶。如果使用这些原则和策略最终让我的设计中出现了一个设计模式，这就是说它们给了我得出优秀设计的方法(因为设计模式都是已经得到承认的优秀设计)。如果使用了这些技术，难道我还会因为不知道某个模式——不管它是否出现——的名字而得到不好的设计吗?

这些领悟帮助我更好地进行我的训练过程(以及我现在的写作过程)。我已经把我的教学进行了好几个阶段。我正在向学生们教授面向对象分析和设计的基础。我在课程中教授设计模式，使用它们来阐述优秀的面向对象分析和设计的例子。另外，通过使用设计模式来教授面向对象概念，我让我的学生们更好地理解了面向对象的原则。而且通过学习指导性原则和策略，我的学生们现在可以创建出质量与模式相媲美的设计。

我在这里讲这个故事，因为本书所讲的模式几乎与我的课程所讲的一样。实际上，从第 3 章开始，这本书基本上就是我的两天课程——*面向模式的设计：从分析到实现的模式*——中的第一天的内容。

阅读本书，你可以学到这些模式。但更重要的是，你可以学到：为什么它们可以起作用? 它们怎样在一起工作? 以及它们所依赖的原则和策略。这对你积累自己的经验将很有帮助。当我在本书中展现出一个问题时，如果你能联想到一个你曾经历过的类似的问题，这将对你很有帮助。本书并不讲述新的知识或新的模式，而是给你一个看待面向对象软件开发的新的视角。我希望在你的学习过程中，你自己的经验与设计模式的原则结合之后能给予你巨大的帮助。

Alan Shalloway
2000 年 12 月

从人工智能到模式再到真正的面向对象

进入设计模式的旅程，我的起点与 Alan 的不一样，但我们都得到了同样的结论。

- 基于模式的分析使你成为一个更有力、更高效的分析者，因为它们让你更抽象地处理你的模型，因为它们向你展示许多其他分析者积累的经验。
- 模式帮助人们学习面向对象的原则。模式帮助解释“为什么我们要这样处理对象”。

我的人工智能(AI)的职业生涯开始于创建基于规则的专家系统。这包括倾听专家的叙述，为他们的决策过程建模，以及把这些模型编码成为一个基于知识系统的规则。在创建这个系统的过程中，我开始看到重复的主题：对于一般形式的问题，专家们往往以相似的方式来解决。比如说，为设备做诊断的那些专家往往先简单而快速地看看问题，然后他们开始更系统化地将问题分解成一个个小的组成部分；但在他们的系统化诊断中，他们也往往先做成本较小的测试或能大规模排除问题可能性的测试，然后才进行其他的测试。无论我们是为计算机还是为油田设备进行诊断，这些规律都是适用的。

今天，我把这些重复的主题称为“模式”。很直观地，当我开始设计新的专家系统时，我开始寻找这些重复的主题。因此，对于模式，我的思想是开放而友好的，尽管当时我还不知道它们是什么。

然后，在1994年，我发现欧洲的研究者们已经把这些专家的行为模式编码并放进一个程序包中。他们把这个程序包称为“知识分析及设计支持”，或KADS。Karen Garder博士，一个非常有才华的分析者、建模专家、顾问，一个天才，在美国率先在工作中引入了KADS。她拓展了欧洲同行的工作，把KADS引入了面向对象系统。她在我的眼前展开了一个完整的世界——基于模式的分析和设计构成的软件世界。这里面很大一部分来自于Christopher Alexander的思想。她的书——Cognitive Patterns(剑桥大学出版社，1998)——描述了这些思想。

突然，我有了一个为专家行为建模的结构。用这个结构进行建模，我不会过早地受困于复杂和异常情况。后面的三个项目花了我更少的时间和重复工作，并且使最终用户更加满意，因为：

- 我可以更快地设计模型，因为模式向我预报了将要遇到的情况。它们告诉我：基本的对象是什么，应该对什么特别注意。
- 我可以更有效地与专家沟通，因为我们有了处理细节和异常情况的更结构化的途径。
- 模式让我可以为我的系统开发更好的最终用户培训程序，因为模式已经向我预报了这个系统最重要的特性。

最后这一点的意义非常重大。模式可以帮助最终用户理解系统，因为是模式提供了系统的来龙去脉——为什么我们用一种特定的方法来处理问题？我们可以用模式来描述系统的指导性原则和策略。而且我们可以利用模式来开发最好的范例，用以帮助最终用户理解这个系统。

我对这一切深深着迷。

所以，当我工作的地方有人开办了一个模式学习组时，我热切地加入了。正是在这个学习组中，我遇到了Alan。在他作为面向对象设计者和顾问的工作中，他和我有类似的感

受。我们相遇，结果就有了这本书。

我希望本书中的原则能给你帮助，帮助你走上使自己成为一个更有力、更高效的分析者的旅程。

James R. Trott
2000 年 12 月

附注：关于使用本书的约定

在写这本书的过程中，我们选择了许多风格和约定。一些选择让我们的读者吃惊。因此，有必要在这里留下一个备注：为什么我们要进行这些选择？

方式	理由
第一人称语气	本书是两位作者合作的结果。为了找到解释这些概念的最佳途径，我们对我们的观点进行了争论和精练。Alan 在他的课程中试讲了其中的一些概念，我们又精练了一些。在本书的主体部分我们选择使用单数第一人称，因为这样我们可以按照我们的希望，用更生动更自然的方式讲出我们的故事。
易于浏览的格式	我们试图让这本书更容易浏览，这样你可以在不阅读主体的情况下得到本书的要点，或者快速找到你需要的信息。我们重点使用了表格和加黑点的列表。我们在边界之外加上了概述段落内容的文字。在讨论每个模式时，我们提供了模式关键特性的概述表。我们的希望是让这本书更容易理解。
程序代码段	本书关注分析和设计超过关注实现。我们的意图是在面向对象设计者们的领悟和优秀实践经验的基础上，帮助你思考如何制做出色的设计，正如在设计模式中所表现出来的一样。我们所有这些程序员都面临的一个挑战是：避免过早进入实现阶段，在做之前先想。由于知道这一点，我们特意试图避免过多地讨论实现。我们的代码看起来可能有点过分简单而不完整。特别是，我们从来不在代码中提供错误检查逻辑。这是因为我们试图用这些代码来说明设计概念。
策略和原则	我们这本书是一本介绍性的书籍。它将帮助你在设计模式方面快速入门。你将理解到促成设计模式的原则和策略。在阅读本书之后，你可以转向更加精深的模式书籍或模式方面的参考书。本书的最后一章将为你指出很多可能对你有用的参考书。
展现宽度，浅尝体会	我们试图让你体会多种设计模式，向你展示模式世界的广阔，但不在其中任何一点过于深入(参见上一点)。 我们的想法是：如果你带一个人到美国参观游览两周，你会带他看什么？也许一些场所可以帮助他了解美国的建筑、社会、城市及城市之外的广大地域、高速公路、还有咖啡店。但你无法带他去看所有的这些。为了丰富他对美国的认识，你可能会带他浮光掠影地参观许多景观和城市，让他形成对这个国家的体验。然后，他可以自己为将来的游览制定一个计划。在这本书里，我们也是在带你参观设计模式中的一些主要景观，让你形成对设计模式的体验，这样你可以为自己学习设计模式的旅程制定一个计划。

反馈

设计模式是一门发展中的科学，是发现最佳做法的从业者之间的对话。正是这些从业者发现了面向对象的基本原则。

我们渴望你关于本书的反馈：

- 我们哪里做得好，哪里做得不好？
- 书中有没有需要改正的错误？
- 书中有没有什么地方写得容易产生混淆？

可以浏览原版书的网页，URL 是 http：//www. netobjectives. com/dexplained。在这个网页上，你会找到一个表单，你可以通过它把你的意见和问题发送给我们。你也会找到我们最新的研究成果。

致谢

几乎每篇前言的结尾部分都有一个列表，对帮助推出该书的人致谢。在我们自己写一本书之前，我们从来没有认识到作者对这些给予帮助的人有多么感激。这些成就实在是许多人集体合作的结果。给予我们帮助的人写出来会是一个长长的单子。下面这些人给了我们特别重大的帮助：

- Addison-Wesley 出版社的 Debbie Lafferty，他不知疲倦地鼓励和鞭策我们。
- 我们的同事 Scott Bain，他非常耐心地审阅了我们的书稿，并给了我们很多启发。
- 另外，尤其是 Leigh 和 Jill，我们耐心的妻子，她们忍受了一切，并不断地鼓励我们为自己的梦想——这本书——努力。

Alan 特别要感谢的：

- 我早期的几个学生，他们可能永远都不知道他们造成的影响。有很多次，我犹豫是否应该在我的课程中引进新的思想，我觉得我应该固守那些经过实践证明的东西。但是，他们在我第一次开始我的课程时表现出的那种激情，鼓励了我越来越多地在课程中引进我自己的思想。感谢 Lance Young、Peter Shirley、John Terrell 和 Karen Allen。对于我来说，他们的名字就是一个永远的暗号，永远在我的征途中给我鼓励。
- 感谢 John Vlissides，他向我们提出了很有思想性的意见和非常宝贵的建议。

Jim 特别要感谢的：

- Karen Gardner 博士，人类思维模式方面的顾问和博学的教师。
- Marel Norwood 博士和 Arthur Murphy，我在 KADS 和基于模式分析项目中最初的合作者。
- Brad Van Beek，他给了我在这个学科中发展的空间。
- Alex Sidey，他教给我这个学科的知识和技术写作的诀窍。

目　录

第1篇　面向对象软件设计简介

第2篇　传统面向对象设计的局限性

第3篇 设计模式

第 5 篇 用设计模式处理变化