The Zen of Design Patterns

禅宗曰：“教外别传，不立文字”，禅的境界本不该用文字来描述，言语也道不明白，但为了传道，悟道者仍要借言语来说明。

何为禅？一种境界，一种体验，一种精神领域的最高修为。何为设计模式？对面向对象思想的深刻理解，对软件设计方法和编码经验的完美总结。

本书是创造者的心路历程，是实践者的智慧结晶，是得道者的禅悟。它通过幽默风趣的故事和通俗易懂的讲述方式，引导你悟透设计模式的真谛。

如果你在思考下面这些问题，也许本书就是你想要的！

- 业务分析如此细致，架构设计如此健壮、可靠和稳定，但为何仍然无法适应业务发展的需要，而且生命周期只有短短几年？
- 为何你的团队协作了多年却始终无法沉淀出可复用的组件或构件？依赖和解耦的标准是什么？如何才能做到既不相互“刺伤”，又能相互“温暖”？
- 架构设计时，如何才能实现高可扩展性和易维护性？如何避免维护成本大于开发成本的悲哀现状？
- 交易型的系统如何大规模地借用设计模式的思想，以实现高性能、高可靠性的建设目标？
- 架构设计时，如果遇到这样的情况：“有一个请求者和多个处理者，同时要求二者之间解耦，以便处理者可以动态地扩展”，这该如何处理？
- 如果遇到过这样的场景：“多个对象依赖一个对象，该对象状态改变时所有的依赖者都要相应地获得通知，并且要求对象间松散耦合”，这该如何处理？
- 万物皆对象，不可能把每一个对象都分解到原子级别，如何适度地细化对象的粒度？怎样界定对象的粒度大小？
- 同为创建类模式，工厂方法模式和建造者模式都可以创建对象，它们之间有何区别？适用的场景又有何不同？
- 状态模式和策略模式的通用类图如此相似，在实际的应用场景中如何区分它们？
- 如何使命令模式和责任链模式完美搭配并建立一个高可扩展性的系统架构，以解决客户端和处理者都参数化的场景？
- 观察者模式和责任链模式真的没有可比性吗？它们的主要区别何在？实际应用中如何使用？
- 组合模式只能用来表示部分和整体的关系吗？其扩展出的规格模式是如何实现的？透明的组合模式和安全的组合模式有何区别？

客服热线：(010) 88378991, 88361066
购书热线：(010) 68326294, 88379649, 68995259
投稿热线：(010) 88379604
读者信箱：hzjsj@hzbook.com

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

上架指导：计算机/程序设计
ISBN 978-7-111-29544-0

9 787111 295440

定价：69.00元

华章原创精品

设计模式之禅

The Zen of Design Patterns

秦小波 著

机械工业出版社
China Machine Press

如果说“四人帮”的《设计模式》是设计模式领域的“圣经”，那么之后出版的各种关于设计模式的书都可称之为“圣经”的“注释版”或“圣经的故事”。本书是得道者对“圣经”的“禅悟”，它既不像“圣经”那样因为惜字如金、字字珠玑而深奥、晦涩和难懂，又比“圣经”的“注释版”更深刻和全面、更通俗和生动、更接近开发者遇到的实践场景、更具指导性。本书兼收并蓄、博采众长，是设计模式领域里的里程碑之作。

全书共分为四部分，第一部分从原理的角度阐述了面向对象程序设计的6大原则；第二部生动地讲解和剖析了23种常见的设计模式，并进行了扩展，通俗易懂，趣味性极强而又紧扣模式的核心；第三部分对各种相关联的设计模式进行了深入分析和比较，旨在阐明各种设计模式比较理想的应用场景和它们之间的区别；第四部分探讨了设计模式的混编，讲解了如何在实际开发中将各种设计模式混合起来使用，以发挥设计模式的最大效用。最后，本书还附有一份设计模式彩图，可以裁剪，便于参考。

图书在版编目（CIP）数据

设计模式之禅/秦小波著. —北京：机械工业出版社，2010.1

ISBN 978-7-111-29544-0

Ⅰ.设… Ⅱ.秦… 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（2010）第007491号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）
责任编辑：李俊竹　周茂辉　陈佳媛　迟振春　刘立卿
北京京师印务有限公司印刷
2010年3月第1版第1次印刷
186mm×240mm · 35.75印张（含0.5印张彩插）
标准书号：ISBN 978-7-111-29544-0
定价：69.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991；88361066
购书热线：（010）68326294；88379649；68995259
投稿热线：（010）88379604
读者信箱：hzjsj@hzbook.com

Praise 赞 誉

“禅”是一种境界，是得道者的智慧结晶。本书在写作方式上别出心裁，不是将设计模式的概念强行灌输给读者，而是以浅显的故事作为衬垫，深入浅出地展示了设计模式中蕴涵的哲理，进而引发读者的思考，提升他们的实际开发水平。

51CTO读书频道（http://book.51cto.com/）

聪明的人，把房子盖在磐石上；无知的人，把房子盖在沙土上。对于开发者而言，设计模式就是那坚固的磐石。本书是设计模式领域难得一见的佳作，它用一种创新的方式对面向对象的设计原则和设计模式的要义进行了全新的阐释。对于所有Java开发者而言，无论你是初窥门径，还是深谙其道，本书都值得你阅读并收藏。

——Java开发者社区

本书不仅从开发者的角度对设计模式进行了独到而具有创意性的讲解，而且还从架构师的角度深刻地分析了设计模式在软件架构中的重要性。设计模式是架构师必备的技能之一，它的思想和原则能指导架构师设计出更优秀的软件架构。强烈推荐所有架构师阅读本书。

——架构师社区

多年前学设计模式，犯困无数次，打盹若干回。程序员一直被幽默着，现在终于可以反幽默一回了。这是一本厚积薄发的书，作者以其丰富的实践经验和通俗的讲解方式，把难懂的设计模式“化为”橡皮泥，让程序员想怎么玩就怎么玩，以致读完全书，仍觉意犹未尽。

——江伍开，知名外企资深架构师，
51CTO博客之星（http://wakan.blog.51cto.com/）

很多初学者都抱怨设计模式的抽象和深奥，对于他们来说，本书的出版不啻是一种福音！作者利用诙谐的语言和生动的叙述方式，结合当前国内读者熟知和易于理解的故事和开发场景，对设计模式进行了独特而全面的阐述，大大降低了设计模式的学习曲线，实在不可多得！

——计文柯，资深软件开发专家和项目经理，
《Spring技术内幕——深入解析Spring架构和设计原理》作者

本书以设计模式为主题，是作者多年软件开发经验的总结。它介绍了一些重要的设计原则，并对各种经典的设计模式进行了详细的分析、比较和综合。全书通俗易懂、实例丰富，对想要学习设计模式的程序员有很大的启发和帮助。

——郑晖，资深软件开发专家和CTO，《冒号课堂》作者

无论是在建筑领域，还是在面向对象软件开发领域，如何强调设计模式的重要性都不过分。如果你觉得设计模式难学，推荐你认真阅读这本书，它用大量生动、有趣的故事将设计模式的深奥、晦涩化解于无形；如果你对设计模式一知半解或只能“纸上谈兵”，建议你反复阅读这本书，它对面向对象的原则、设计模式的内涵和外延，以及设计模式的应用场景做了全面而深刻的阐述。

——王福强，资深软件开发专家和架构师，《Spring揭秘》作者

不管是新手还是大侠，本书绝不容错过，因为不能熟练地运用设计模式，就不能算是一名合格的程序员。它通俗易懂，作者精心挑选和设计的故事和案例引人入胜，是初学者的不二选择；它系统而全面，但又不乏深度，处处渗透着作者对设计模式的“悟”，是合格程序员必备的修炼秘籍。

——徐彬，资深软件开发专家和架构师，《GWT揭秘》作者

作者在本书中表现出来的想象力、创造力以及对设计模式和软件架构的深刻理解，让我十分震撼。我喜欢这种通过想象来讲解设计模式和剖析软件结构的方式，它一定会让你也受益匪浅。

——倪健，资深架构师和项目经理，《软件开发之禅》作者

Preface 前　言

终于可以写前言了，这说明本书已经基本完成，可以长嘘一口气了。

为什么写这本书

今年5月份，我在JavaEye上发了一个帖子，其中提到自己已经工作9年了，总觉得这9年不应该就这么荒废了，应该给自己这9年的工作写一个总结，总结的初稿就是这本书。

在谈为什么写这本书之前，先抖抖自己这9年的职业生涯吧。大学时我是学习机械的，当时计算机刚刚热起来，自己也喜欢玩一些新奇的东西，记得最清楚的是用VB写了一个自由落体的小程序，模拟小球从桌面掉到地板上，然后计算反弹趋势，很有成就感。于是2000年毕业时，我削尖了脑袋进入了IT行业，成为了一名真正的IT男，干着起得比鸡早、睡得比鸡晚的程序员工作，IT男的辛酸有谁知晓！

坦白地说，我的性格比较沉闷，属于典型的程序员型闷骚，比较适合做技术研究。在这9年里，项目管理做过，系统分析做过，小兵当过，团队领导人也当过，但至今还是一个做技术的。要总结这9年技术生涯，总得写点什么吧，最好是还能对其他人有点儿用的。那写什么好呢？Spring、Struts等工具框架类的书太多太多，很难再写出花样来，经过一番思考，最后选择了一个每一位技术人员都需要掌握的、但普及程度还不是非常高的、又稍微有点难度的主题——设计模式（Design Pattern，DP）。

中国人有不破不立的思维，远的如秦始皇焚书坑儒、项羽火烧阿房宫，近的如破“四旧”。正是由于有了这样的思想，于是乎能改的就改，不能改的就推翻重写，没有一个持续开发蓝图。为什么要破才能立呢？为什么不能持续地发展？你说这是谁的错呢？是你架构师的错，你不能持续地拥抱变化，这是一个系统最失败的地方。那怎么才能实现拥抱变化的理想呢？设计模式！

设计模式是什么？它是一套理论，由软件界的先辈们（The Gang of Four：包括Erich Gamma、Richard Helm、Ralph Johnson、John Vlissides）总结出的一套可以反复使用的经验，它可以提高代码的可重用性，增强系统的可维护性，以及解决一系列的复杂问题。做软件的人都知道需求是最难把握的，我们可以分析现有的需求，预测可能发生的变更，但是我们不能控

制需求的变更。问题来了，既然需求的变更是不可控的，那如何拥抱变化呢？幸运的是，设计模式给了我们指导，专家们首先提出了6大设计原则，但这6大设计原则仅仅是一系列“口号”，真正付诸实施还需要有详尽的指导方法，于是23种设计模式出现了。

设计模式已经诞生15年了，在这15年里出版了很多关于它的经典著作，相信大家都能如数家珍。尽管有这么多书，工作5年了还不知道什么是策略模式、状态模式、责任链模式的程序员大有人在。不信？你找个机会去“虚心”地请教一下你的同事，看看他对设计模式有多少了解。不要告诉我要翻书才明白！设计模式不是工具，它是软件开发的哲学，它能指导你如何去设计一个优秀的架构、编写一段健壮的代码、解决一个复杂的需求。

因为它是软件行业的经验总结，因此它具有更广泛的适应性，不管你使用什么编程语言，不管你遇到什么业务类型，设计模式都可以自由地“侵入”。

因为它不是工具，所以它没有一个可以具体测量的标尺，完全以你自己的理解为准，你认为自己多了解它，你就有可能产生多少的优秀代码和设计。

因为它是指导思想，你可以在此基础上自由发挥，甚至是自己设计出一套设计模式。

世界上最难的事有两件：一是让人心甘情愿地把钱掏出来给你，二是把自己的思想灌输到别人的脑子里。设计模式就属于第二种，它不是一种具体的技术，不像Struts、Spring、Hibernate等框架。一个工具用久了可以熟能生巧，就像砌墙的工人一样，长年累月地砌墙，他也知道如何把墙砌整齐，如何多快好省地干活，这是一个人的本能。我们把Struts用得很溜，把Spring用得很顺手，这非常好，但这只是一个合格的程序员应该具备的基本能力！于是我们被冠以代码工人（Code Worker）——软件行业的体力劳动者。

如果你通晓了这23种设计模式就不同了，你可以站在一个更高的层次去赏析程序代码、软件设计、架构，完成从代码工人到架构师的蜕变。注意，我说的是“通晓”，别告诉我你把23种设计模式的含义、适应性、优缺点都搞清楚了就是通晓。错了！没有工作经验的积累是不可能真正理解设计模式的，这就像大家小时候一直不明白为什么爸爸妈妈要工作而不能每天陪自己玩一样。

据说有的大学已经开了设计模式这门课，如果仅仅是几堂课，让学生对设计模式有一个初步的了解，我觉得并无不妥，但如果是专门的一门课程，我建议取消它！因为对一个尚无项目开发经验的学生来说，理解设计模式不是一般困难，而是非常非常困难！之前没有任何的实战经验，之后也没有可以立即付诸实践的场景，这样能理解设计模式吗？

在编写本书之前，23种设计模式我都用过，而且还算比较熟练，但是当真正要写到书中时，感觉心里没谱儿了。这个定义是这样的吗？是需要用抽象类还是应该用接口？为什么在这里不能抽取抽象呢？为什么在实际项目中这个模式要如此蜕化？这类小问题有时候很纠结，需要花费大量的精力和时间去分析和确认。所以，在写作的过程中我有过很多忧虑，担心书中会有太多瑕疵，这种忧虑现在仍然存在。遇到挫折的时候也气馁过，但是我坚信一句话：“开弓没有回头箭，回头即是空”，既然已经开始，就一定要圆满完成。

本书的特色

简单、通俗、易懂，但又不肤浅，这是本书的最大特色。自己看过的技术书还算比较多，很痛恨那种大块头的巨著，搁家里当枕头都觉得太硬。如果要是再晦涩难懂点，那根本没法看，看起来实在是太累。设计模式原本就是理论性的知识，讲解的难度比较大，但我相信这本书能够把你对设计模式的恐惧一扫而光。不信？挑几页先看看！

我的理念是：**像看小说一样阅读本书。**我尽量用浅显通俗的语言讲解，尽量让你有继续看下去的欲望，尽量努力让你有兴趣进入设计模式的世界，兴趣是第一老师嘛！虽然我尽量让这本书浅显、通俗、易懂，但并不代表我的讲解就很肤浅。每个设计模式讲解完毕之后，我都附加了两个非常精华的部分：设计模式扩展和最佳实践，这是俺压箱底的技能了，为了博君一看，没招了，抖出来吧！**尤为值得一提的是，本书还有设计模式PK和混编设计模式两部分内容教你如何自如地去运用这些设计模式，这是当前所有设计模式类的图书都不具备的，连最权威的那本书也不例外。**

我很讨厌技术文章中夹杂着的那些晦涩难懂的文字，特别是一堆又一堆的名词堆砌，让人看着就反胃。但是为了学习技术，为了生存，还是必须看下去。国内的技术文档，基本上都是板着一副冷面孔讲技术，为什么要把技术弄得这么生硬呢？技术也有它幽默、柔情的一面，只是被我们的“孔夫子们”掩盖了，能用萝卜、白菜这种寻常人都熟悉的知识来讲解原子弹理论的人，那是牛人，我佩服这样的人。记住，用一堆名词把你忽悠晕的人很可能什么都不懂！

本书想告诉你的是，技术也可以很有乐趣，也可以让你不用皱着眉头思考，等待你的只是静静地看，慢慢地思考，本书的内容会润物细无声地融入你的思维中。

本书面向的读者

热爱技术并且讨厌枯燥乏味技术文章的读者都可以看本书；

你是**程序员**，没问题，本书能够让你写出更加高效、优雅的代码；

你是**架构师**，那更好，设计模式可让你设计出健壮、稳定、高效的系统，并且自动地预防未来业务变化可能对系统带来的影响；

你是**项目经理**，也OK，设计模式可以让你的工期大大缩短，让你的项目团队成员快速地理解你的意图，最终的成果就是优质的项目：高可靠性、高稳定性、高效率和低维护成本。

如何阅读本书

首先声明，本书中所有的例子都是用Java语言来实现的，但是你可以随手翻翻看，基本上能保证每三条语句一个注释，可以说是在用咱们的母语讲解设计模式。即使你不懂Java语言，

也没有关系，只要知道在Java中双斜杠（//）代表注释就足够了，况且Java如此强大和盛行，多了解一点没有坏处。类图看不懂？没关系，不影响你理解设计模式，多看看就懂了！

如果你还没有编程经验，我建议你把它当作小说来看，懂行的看门道，不懂行的看热闹，这里的热闹足够多，够你看一壶的了。你现在能看懂多少是多少，不懂没有关系，你要知道，经验不是像长青春痘一样，说长就长出来了，它是需要时间积累的，需要你用心去感受，然后才能明白为什么要如此设计。

如果你已经对编程有感觉了（至少2年开发经验），我相信你都能看懂，但能“懂”到什么程度，就很难说了，看你的水平了。但是，我可以保证，这里的设计模式都是你能看懂的，没有你看不懂的！我建议你通读这本书，然后挑门你最得意的编程语言，动手写吧！给自己制定一个计划，每天编写一段代码，不需要太多，200行足够，时不时地把设计模式融入到你的代码中。甭管是什么代码，比如你想编写一个识别美女图片的程序，好呀，抓紧时间去写吧，写好了就不用到处看美女了，程序一跑就把网上的美女图片都抓过来了，牛呀（记住，程序写好了要分享给我）。看吧，坚持下去，一年以后你再跟你的同侪比较一下，那差距肯定不是一般的大。

如果你是资深工程师、架构师、技术顾问等高等级的技术人员，那我告诉你，你找对这本书了。系统架构没有思路？没有问题，看看扩展部分，它会开阔你的思路。系统的维护成本居高不下？看看本书，设计模式也许能帮你省点银子。开发资源无法保证？设计模式能让你用有限的资源（软硬件资源和人力资源）设计出一个优秀的系统。项目质量参差不齐，缺陷一大堆？多用设计模式，它会给你意想不到的效果。给人讲课没有素材？没问题，本书中的素材足以让你赢得阵阵掌声！

编程是一门艺术活，我有一个同事，能把类图画成一个小乌龟的形状，天才呀！作为一位技术人员，最基本的品质就是诚实，“知之为知之，不知为不知，是知也”，自己不懂没有关系，去学，学无止境，但是千万不要贪多，这抓一点，那挖一点，好像什么都懂，其实什么都不懂。中国一直推崇复合型人才，我不是很赞成，因为这对年轻人来说是一个误导。先精一项技术，然后再发散学习，先点后面才是正道。

记得《武林外传》中有这样一段对话：

刑捕头：手中无刀，心中有刀。

老白：错了，最高境界是手中无刀，心中也无刀。

体验一下吧，我们的设计模式就是一把刀，极致的境界就是心中无设计模式，代码亦无设计模式——设计模式随处可见，俯拾皆是，已经融入软件设计的灵魂中，这才是高手中的高手，简称高高手。

哦，最最重要的忘记说了，请把附录中的“23种设计模式附图”撕下来，贴在你的办公桌前，时不时地看看，也让老板看看，咱是多么地用心！

关于书名

乍一看，书名和内容貌似不相符呀，其实不然！

在我们的常规思维中，“禅”应该是很高深的东西，只可意会，不可言传。没错，禅宗也是如此说。禅是得道者的“悟”，是不能用言语来表达的，但是得道者为了能让更多的人“悟”，就必须用最容易让人理解的文字把自己的体会表达出来。本书的“禅”是作者对设计模式的“悟”，本书的“形”就是你现在看到的这些极其简单、通俗、易懂的文字。

至此，大家应该不会再对书名有疑虑了吧，嘿嘿。

致谢

本书的写作耗时7个月，可以说是榨干了海绵里所有的水——基本上能用的时间都用上了。在公交车上打腹稿，干过！在马桶上查资料，干过！在睡梦中思考案例，也有过！就差没有走火入魔了！

首先，感谢杨福川编辑，没有他的慧眼，这本书不可能出版。其次，感谢妻子和儿子，每天下班回到家，一按门铃，儿子就在里面叫：“我来开门，我来开门。”儿子三岁，太调皮了，他不睡觉我基本上是不能开写的，我一旦开始写东西，他就跑过来问：“爸爸，你在干什么呀”，紧接着下一句就是“爸爸，你陪我玩”，基本都是拿我当玩具，别的小朋友都是把父亲当马骑，他却不，他把我当摩托车骑，还要加油门，发动……小家伙脚太重了，再骑摩托，非被他踩死不可！

再次，还要感谢《Spring技术内幕——深入解析Spring架构与设计原理》的作者计文柯先生、《冒号课堂：编程范式与OOP思想》的作者郑晖先生、《Spring揭秘》的作者王福强先生、《GWT揭秘》的作者徐彬先生，他们专业、细致、耐心的审核使得本书更加完美，特别是郑晖先生，虽言语不多，但言必中的，受益匪浅。非常幸运能获得他们四位的指导！

最后要感谢我的朋友王骢，周末只要小家伙在家，我只有找地方写书的份儿，王骢非常爽快地把钥匙给我，让我有一个安静的地方写书。一个人沉浸在自己喜欢的世界里也是一件非常幸福的事。

当然，还要感谢JavaEye上所有顶帖的网友，没有你们的支持我就失去了写作的动力，就像希腊神话中的巨人安泰失去了大地的力量一样，是你们的回帖让我觉得不孤单，让我知道我不是一个人在战斗！

最后，再次对本书中可能出现的错误表示歉意，真诚地接受大家轰炸！如果你在阅读本书时发现错误或有问题想讨论，请发邮件给我。

我的邮箱：cbf4life@126.com。

目　录 Contents

第四部分 完美世界——设计模式混编

Part1 第一部分

大旗不挥，谁敢冲锋——6大设计原则全新解读