设计模式之禅（第2版）

作者：秦小波

华章原创精品

设计模式之禅（第2版）

秦小波　著

ISBN：978-7-111-43787-1

本书纸版由机械工业出版社于2014年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

目　录

前言

为什么写这本书

2009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，先抖抖自己前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、睡得比狗晚的程序员工作，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。

设计模式已经诞近20年了，其间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。

第2版与第1版的区别

本书是第2版，在写作中吸取了读者对上一版的许多意见和建议，修订了一些代码的变量、类、方法名称，以更加符合自然语言；删除了部分有争议的内容（如单例模式的垃圾回收问题）；修改了一些常用的名词，确保与编程人员的习惯相匹配。希望通过这些改进，给读者提供一个更完美的设计模式盛宴，弥补上一版中的诸多不足。

第2版第38章中新增了4种新的设计模式：对象池模式、雇工模式、黑板模式、空指针模式。这些模式是我们在实际工作中经常遇到，或者在开源代码时常看到的，但是我们却没有升级到“模式”这一理性高度。特别是像空指针模式，我们在编码中经常会遇到空值判断问题，但我们没有去想一想是否可以有更好的方式解决。第2版对空指针模式进行了讲解，虽然简单，但相信对你提升编码质量有很大的帮助。

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，我建议你把它当做小说来看，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（至少两年开发经验），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。