<<设计模式之禅>>

图书基本信息

书名：<<设计模式之禅>>

13位ISBN编号：9787111295440

10位ISBN编号：7111295447

出版时间：2010年3月

出版时间：机械工业出版社

作者：秦小波

页数：545

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。

<<设计模式之禅>>

内容概要

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，也许是设计模式领域里的下一个里程碑之作。

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。

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，悟道者仍要藉言语来说明。

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！

1. 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？

2.
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”？

3. 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？

4. 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？

5.
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？

6.

如果遇到过这样场景：“ 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？

7. 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颗粒度？
怎样界定对象的粒度大小？

8. 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？

9. 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？

10. 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？

11. 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？

12. 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？

作者简介

秦小波，资深软件开发工程师、项目经理、系统分析师和架构师（获Sun架构师认证），从事IT行业10余年，经验极其丰富，现就任于交通银行软件研发中心。
精通设计模式，对设计模式有深刻认识和独到见解，创造性地提出了自己在大量实践中总结出来的新的设计模式。
擅长于SSH、iBati

<<设计模式之禅>>

书籍目录

章节摘录

插图：第一部分大旗不挥，谁敢冲锋——热身篇第1章单一职责原则单一职责原则的英文名称是SingleResponsibilityPrinciple，简称是SRP。
这个设计原则备受争议，只要你想和别人争执、怄气或者是吵架，这个原则是屡试不爽的。
如果你是老大，看到一个接口或类是这样或那样设计的，你就问一句：“你设计的类符合SRP原则吗?”保准对方立马“萎缩”掉，而且还一脸崇拜地看着你，心想：“老大确实英明”。
这个原则存在争议之处在哪里呢?就是对职责的定义，什么是类的职责，以及怎么划分类的职责。
我们先举个例子来说明什么是单一职责原则。
只要做过项目，肯定要接触到用户、机构、角色管理这些模块，基本上使用的都是RBAC模型（Role-BasedAccessControl，基于角色的访问控制，通过分配和取消角色来完成用户权限的授予和取消，使动作主体（用户）与资源的行为（权限）分离），确实是一个很好的解决办法。
我们这里要讲的是用户管理、修改用户的信息、增加机构（一个人属于多个机构）、增加角色等，用户有这么多的信息和行为要维护，我们就把这些写到一个接口中，都是用户管理类嘛，我们先来看它的类图，如图1.1所示。

媒体关注与评论

“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地阐述。
——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

<<设计模式之禅>>

编辑推荐

《设计模式之禅》：继GOF《设计模式》之后的又一里程碑之作！
6大原则，23+1种设计模式设计模式PK、设计模式混编、设计模式最佳实践设计模式领域的又一里程碑之作同样是导演，为什么詹姆斯·卡梅隆、史蒂芬·斯皮尔伯格能够制作出如此让人惊心动魄的旷世巨著呢？
同样是建筑师，为什么贝聿铭、圣地亚哥·卡拉特拉瓦能够创造出如此美丽、和谐、雄伟的建筑呢？
同样是程序员或架构师，我们的作品又应该达到怎样的境界？
诚然，技术和创造力我们都不缺，缺少的是为软件注入灵魂的方式和方法，设计模式正是这一系列方式和方法的集大成者。
巧妙地应用设计模式可以让我们的代码变得更健壮、更易于理解和维护，从而显著提高系统的性能、可靠性、稳定性、可维护性和可扩展性，是成长为优秀程序员和架构师的必备技能。
“他山之石，可以攻玉”，《设计模式之禅》以亲切、自然的风格阐述了设计模式的核心思想，潜移默化地提升我们面向对象的架构和编程能力，带我们进入“物我合一、见性成佛”的最高设计境界。

版权说明