

# 《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

出版社：机械工业出版社

作者：秦小波

页数：545

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：[www.tushu111.com](http://www.tushu111.com)

## 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。

# 《设计模式之禅》

## 作者简介

秦小波，资深软件开发工程师、系统分析师和架构师（获Sun架构师认证），从事软件开发工作10余年，实践经验极其丰富。资深Java技术专家，精通Java语言、Spring、Struts 2、Hibernate、iBatis、jBPM等Java技术，在企业级Java应用领域积累了大量工程经验，对ESB、BPEL等整合技术也有较深入的认识。精通设计模式，对设计模式有深刻的认识和独到见解，而且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。他撰写的《设计模式之禅》一书凭借优质的内容和良好的可读性广获读者好评，被誉为“设计模式领域的里程碑之作”。此外，他还是一位优秀的DBA，获IBM DB2 DBA资格认证，对海量数据处理有深入的研究。

## 书籍目录

前言

第一部分 大旗不挥，谁敢

冲锋—热身篇

第1章 单一职责原则

1.1 我是“牛”类，我可以担任多职吗

1.2 绝杀技，打破你的传统思维

1.3 我单纯，所以我快乐

1.4 最佳实践

第2章 里氏替换原则

2.1 爱恨纠葛的父子关系

2.2 纠纷不断，规则压制

2.3 最佳实践

第3章 依赖倒置原则

3.1 依赖倒置原则的定义

3.2 言而无信，你太需要契约

3.3 依赖的三种写法

3.4 最佳实践

第4章 接口隔离原则

4.1 接口隔离原则的定义

4.2 美女何其多，观点各不同

4.3 保证接口的纯洁性

4.4 最佳实践

第5章 迪米特法则

5.1 迪米特法则的定义

5.2 我的知识你知道得越少越好

5.3 最佳实践

第6章 开闭原则

6.1 开闭原则的定义

6.2 开闭原则的庐山真面目

6.3 为什么要采用开闭原则

6.4 如何使用开闭原则

6.5 最佳实践

第二部分 我惹了谁—

真刀实枪篇

第7章 单例模式

7.1 我是皇帝我独苗

7.2 单例模式的定义

7.3 单例模式的应用

7.4 单例模式的扩展

7.5 最佳实践

第8章 工厂方法模式

8.1 女娲造人的故事

8.2 工厂方法模式的定义

8.3 工厂方法模式的应用

8.3.1 工厂方法模式的优点

8.3.2 工厂方法模式的使用场景

8.4 工厂方法模式的扩展

## 8.5 最佳实践

## 第9章 抽象工厂模式

### 9.1 女娲的失误

### 9.2 抽象工厂模式的定义

### 9.3 抽象工厂模式的应用

#### 9.3.1 抽象工厂模式的优点

#### 9.3.2 抽象工厂模式的缺点

#### 9.3.3 抽象工厂模式的使用场景

#### 9.3.4 抽象工厂模式的注意事项

### 9.4 最佳实践

## 第10章 模板方法模式

### 10.1 辉煌工程—制造悍马

### 10.2 模板方法模式的定义

### 10.3 模板方法模式的应用

### 10.4 模板方法模式的扩展

### 10.5 最佳实践

## 第11章 建造者模式

### 11.1 变化是永恒的

### 11.2 建造者模式的定义

### 11.3 建造者模式的应用

### 11.4 建造者模式的扩展

### 11.5 最佳实践

## 第12章 代理模式

### 12.1 我是游戏至尊

### 12.2 代理模式的定义

### 12.3 代理模式的应用

#### 12.3.1 代理模式的优点

#### 12.3.2 代理模式的应用

### 12.4 代理模式的扩展

#### 12.4.1 普通代理

#### 12.4.2 强制代理

#### 12.4.3 代理是有个性的

#### 12.4.4 虚拟代理

#### 12.4.5 动态代理

### 12.5 最佳实践

## 第13章 原型模式

### 13.1 个性化电子账单

### 13.2 原型模式的定义

### 13.3 原型模式的应用

#### 13.3.1 原型模式的优点

#### 13.3.2 原型模式的使用场景

### 13.4 原型模式的注意事项

#### 13.4.1 构造函数不会被执行

#### 13.4.2 浅拷贝和深拷贝

#### 13.4.3 clone与final两个冤家

### 13.5 最佳实践

## 第14章 中介者模式

### 14.1 进销存管理是这个样子的吗？

### 14.2 中介者模式的定义

- 14.3 中介者模式的应用
- 14.4 中介者模式的实际应用
- 14.5 最佳实践
- 第15章 命令模式
  - 15.1 项目经理也难当
  - 15.2 命令模式的定义
  - 15.3 命令模式的应用
    - 15.3.1 命令模式的优点
    - 15.3.2 命令模式的缺点
    - 15.3.3 命令模式的使用场景
  - 15.4 命令模式的扩展
    - 15.4.1 未讲完的故事
    - 15.4.2 反悔问题
  - 15.5 最佳实践
- 第16章 责任链模式
  - 16.1 古代妇女的枷锁——“三从四德”
  - 16.2 责任链模式的定义
  - 16.3 责任链模式的应用
    - 16.3.1 责任链模式的优点
    - 16.3.2 责任链模式的缺点
    - 16.3.3 责任链模式的注意事项
  - 16.4 最佳实践
- 第17章 装饰模式
  - 17.1 罪恶的成绩单
  - 17.2 装饰模式的定义
  - 17.3 装饰模式应用
    - 17.3.1 装饰模式的优点
    - 17.3.2 装饰模式的缺点
    - 17.3.3 装饰模式的应用
  - 17.4 最佳实践
- 第18章 策略模式
  - 18.1 刘备江东娶妻，赵云他容易吗
  - 18.2 策略模式的定义
  - 18.3 策略模式的应用
    - 18.3.1 策略模式的优点
    - 18.3.2 策略模式的缺点
    - 18.3.3 策略模式的应用
    - 18.3.4 策略模式的注意事项
  - 18.4 策略模式的扩展
  - 18.5 最佳实践
- 第19章 适配器模式
  - 19.1 业务发展——上帝才能控制
  - 19.2 适配器模式的定义
  - 19.3 适配器模式的应用
    - 19.3.1 适配器模式的优点
    - 19.3.2 适配器模式的应用
    - 19.3.3 适配器模式的注意事项
  - 19.4 适配器模式的扩展
  - 19.5 最佳实践

## 第20章 迭代器模式

### 20.1 整理项目信息—苦差事

### 20.2 迭代器模式的定义

### 20.3 迭代器模式的应用

### 20.4 最佳实践

## 第21章 组合模式

### 21.1 公司的人事架构是这样的吗

### 21.2 组合模式的定义

### 21.3 组合模式的应用

#### 21.3.1 组合模式的优点

#### 21.3.2 组合模式的缺点

#### 21.3.3 组合模式的应用

#### 21.3.4 组合模式的注意事项

### 21.4 组合模式的扩展

#### 21.4.1 真实的组合模式

#### 21.4.2 透明的组合模式

#### 21.4.3 组合模式的遍历

### 21.5 最佳实践

## 第22章 观察者模式

### 22.1 韩非子身边的卧底是谁派来的

### 22.2 观察者模式的定义

### 22.3 观察者模式的应用

#### 22.3.1 观察者模式的优点

#### 22.3.2 观察者模式的缺点

#### 22.3.3 观察者模式的应用

#### 22.3.4 观察者模式的注意事项

### 22.4 观察者模式的扩展

#### 22.4.1 java世界中的观察者模式

#### 22.4.2 项目中真实观察者模式

#### 22.4.3 订阅发布模型

### 22.5 最佳实践

## 第23章 门面模式

### 23.1 我要投递信件

### 23.2 门面模式的定义

### 23.3 门面模式的应用

#### 23.3.1 门面模式的优点

#### 23.3.2 门面模式的缺点

#### 23.3.3 门面模式的应用

### 23.4 门面模式的注意事项

#### 23.4.1 一个子系统可以有多个门面

#### 23.4.2 门面不参与子系统内的业务逻辑

### 23.5 最佳实践

## 第24章 备忘录模式

### 24.1 如此追女孩子，你还不乐

### 24.2 备忘录模式的定义

### 24.3 备忘录模式的应用

#### 24.3.1 备忘录模式的应用



- 24.3.2 备忘录模式的注意事项
- 24.4 备忘录模式的扩展
  - 24.4.1 clone方式的备忘录
  - 24.4.2 多状态的备忘录模式
  - 24.4.3 多备份的备忘录
  - 24.4.4 封装得更好一点
- 24.5 最佳实践
- 第25章 访问者模式
  - 25.1 员工的隐私何在？
  - 25.2 访问者模式的定义
  - 25.3 访问者模式的应用
    - 25.3.1 访问者模式的优点
    - 25.3.2 访问者模式的缺点
    - 25.3.3 访问者模式的应用
  - 25.4 访问者模式的扩展
    - 25.4.1 统计功能
    - 25.4.2 多个访问者
    - 25.4.3 双分派
  - 25.5 最佳实践
- 第26章 状态模式
  - 26.1 城市的纵向发展功臣—电梯
  - 26.2 状态模式的定义
  - 26.3 状态模式的应用
    - 26.3.1 状态模式的优点
    - 26.3.2 状态模式的缺点
    - 26.3.3 状态模式的应用
    - 26.3.4 状态模式的注意事项
  - 26.4 最佳实践
- 第27章 解释器模式
  - 27.1 四则运算你会吗
  - 27.2 解释器模式的定义
  - 27.3 解释器模式的应用
    - 27.3.1 解释器模式的优点
    - 27.3.2 解释器模式的缺点
    - 27.3.3 解释器模式使用的场景
    - 27.3.4 解释器模式的注意事项
  - 27.4 最佳实践
- 第28章 享元模式
  - 28.1 内存溢出，司空见惯
  - 28.2 享元模式的定义
  - 28.3 享元模式的应用
    - 28.3.1 享元模式优点和缺点
    - 28.3.2 享元模式的应用
  - 28.4 享元模式的扩展
    - 28.4.1 线程安全的问题
    - 28.4.2 性能平衡
  - 28.5 最佳实践
- 第29章 桥梁模式
  - 29.1 我有一个梦想……

- 29.2 桥梁模式的定义
- 29.3 桥梁模式的应用
  - 29.3.1 桥梁模式的优点
  - 29.3.2 桥梁模式的应用
  - 29.3.3 桥梁模式的注意事项
- 29.4 最佳实践
- 第三部分 谁的地盘谁做主—模式pk篇
- 第30章 创建类模式大pk
  - 30.1 工厂方法模式vs建造者模式
    - 30.1.1 按工厂方法建造超人
    - 30.1.2 按建造者模式建造超人
    - 30.1.3 最佳实践
  - 30.2 抽象工厂模式vs建造者模式
    - 30.2.1 按抽象工厂模式生产车辆
    - 30.2.2 按建造者模式生产车辆
    - 30.2.3 最佳实践
- 第31章 结构类模式大pk
  - 31.1 代理模式vs装饰模式
    - 31.1.1 代理模式
    - 31.1.2 装饰模式
    - 31.1.3 最佳实践
  - 31.2 装饰模式vs适配器模式
    - 31.2.1 按装饰模式描述丑小鸭
    - 31.2.2 按适配器模式实现丑小鸭
    - 31.2.3 最佳实践
- 第32章 行为类模式大pk
  - 32.1 命令模式vs策略模式
    - 32.1.1 策略模式实现压缩算法
    - 32.1.2 命令模式实现压缩算法
    - 32.1.3 小结
  - 32.2 策略模式vs状态模式
    - 32.2.1 策略模式实现人生
    - 32.2.2 状态模式实现人生
    - 32.2.3 小结
  - 32.3 观察者模式vs责任链模式
    - 32.3.1 责任链模式实现dns解析过程
    - 32.3.2 触发链模式实现dns解析过程
    - 32.3.3 小结
- 第33章 跨战区pk
  - 33.1 策略模式vs桥梁模式
    - 33.1.1 策略模式实现邮件发送
    - 33.1.2 桥梁模式实现邮件发送
    - 33.1.3 最佳实践
  - 33.2 门面模式vs中介者模式
    - 33.2.1 中介者模式实现工资计算
    - 33.2.2 门面模式实现工资计算
    - 33.2.3 最佳实践
  - 33.3 包装模式群pk

- 33.3.1 代理模式
- 33.3.2 装饰模式
- 33.3.3 适配器模式
- 33.3.4 桥梁模式
- 33.3.5 最佳实践
- 第四部分 完美世界—混编模式
- 第34章 命令模式+责任链模式
- 34.1 搬移unix的命令
- 34.2 混编小结
- 第35章 工厂方法模式+策略模式
- 35.1 迷你版的交易系统
- 35.2 混编小结
- 第36章 观察者模式+中介者模式
- 36.1 事件触发器的开发
- 36.2 混编小结
- 第37章 规格模式
- 37.1 规格模式的实现
- 37.2 最佳实践
- 第38章 mvc框架
- 38.1 mvc框架的实现
- 38.1.1 mvc的系统架构
- 38.1.2 模型管理器
- 38.1.3 值栈
- 38.1.4 视图管理器
- 38.1.5 工具类
- 38.2 最佳实践
- 附录：23个设计模式

## 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、愠气或者是吵架，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。

## 《设计模式之禅》

### 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。

## 精彩短评

- 1、比小说还要有趣的技术书，可是读完了只记得像小说的部分，不知道是幸运还是不幸。。。。
- 2、不推荐读，作者解读的虽然通俗但是产生了很多模式上曲解。入门的话推荐head first设计模式，熟练后还是读gof吧。
- 3、本书对23种设计模式和OOP六大原则都有很好的解释和例子，通俗易懂，很适合学习设计模式的一本书。
- 4、还行 网上好像有这样的例子
- 5、书中讲了很多设计模式，在工作中有很多值得借鉴的地方。尤其作者是有丰富的工作经验，书写的很好，希望有更多这类的电子书。
- 6、  
酒吧以高薪聘“男女公关”为名 骗了广州市快腾贸易有限公司近百人  
“不许动，统统举起手来！”3月8日下午2时30分，在深圳南山区粤海街道大冲村的飞马酒吧内  
，“内应”侦查  
员配合数十便衣警察，将酒吧内50多名涉案人员全部控制起来。至此，一个以高薪招聘“男女公关”为名的18人诈骗团伙全部落网。

为“两万元月薪”

交出500元血汗钱

今年20岁的小林来自江西，在深圳找工作期间看到写有“内部直招，非中介，免押金”字样的广告，一下子就被其“男女公关月薪2万元”“保安一职不算小费也有2700元工资”的优厚条件吸引住了，于是到约定地点飞马酒吧面试。

酒吧守门人直至小林报出联系人贺小姐手机后面4位号码“9662”的暗号后，才允许其进入酒吧包间接受“面试”。“面试”时，接待人黄某告知小林保安一职已招满，而经理助理也早有人选了。正当失望的小林准备离去时，黄某改口说：“现在紧缺的是高级服务员和公关。”

当得知“公关”就是陪富婆聊聊天喝喝酒，一个月能轻松赚三四万元的职业时，小林毫不犹豫地按要求将身上仅有的500元交了出来，在酒吧里一心等消息。

“入围者”还要再交

4000元“继续包装”

此时，酒吧里突然冲进来数十名便衣警察，把小林吓了一跳。随着一句“不许动，统统举起手来”的大喝，酒吧里两名“望风”人员、一名“咨客”，酒吧大厅内30余人及“包房”内“面试”的10多人迅速被控制，民警还搜出账本、传单、赃款等证据一批。

警方在围捕过程中，酒吧里又先后涌进了15名应聘者。经过辨认、清点、勘查，18名涉嫌诈骗的疑犯无一漏网。

经查，酒吧内存在一个特大的招工诈骗团伙。这个大团伙又分为两个相互独立的团伙，其中以王×勇、刘×明等人为首的团伙，以7000元的月租租得场地。他们将其中一半场地租给以方×萍、王某等人为首的另一个团伙共同“营业”。

男性应聘者交纳300元到1500元不等的“包装费”后，会被带到某些夜总会“绕一圈”，然后告知“富婆”没看上，想要继续做“公关”，还需交4000元的“继续包装费”。而少数容貌姣好的女应聘

## 《设计模式之禅》

者则成为“坐台”小姐，大部分女子所交的入职费都落入了诈骗团伙成员的腰包。

据警方初步统计，该团伙总“营业额”在10万元以上，受骗者多达上百人。

- 7、读了一部分，举的例子很生动有趣，同时也适合作为一本工具书查阅
- 8、设计模式之禅pdf比书上少了相当多的东西，lz可以看一下原书
- 9、讲的浅显易懂，结构清晰
- 10、学习设计模式还是以+理论+动手+反思+重构的路线比较能掌握。此书有很多形象的例子，但是我觉得部分例子和讲解，还是没有打到心坎去。
- 11、这类书籍，找一本好的看就好了，它写的只是有趣。
- 12、书本超赞，针对java语言
- 13、一般般 适合入门
- 14、我是个刚刚入行半年的小鸟，只读完了前六章，因为答应了要在收到书的2周内写出书评，所以断章取义的写了些文字....ok, 切入正题:

本书前6章比较详细的介绍了6大设计原则，相对其他设计模式的书籍而言我觉得这种方式比较能让我这种菜鸟入门；作者在每章首先抛出定义，然后是比较有趣实例和一些容易理解的代码，接着阐述为什么要用，优缺点和一些经验建议；但是总感觉有点意犹未尽，换言之我觉得如果能再添加一些点睛之笔那句更好啦，对于这点我觉得冒号课堂的文章写的就不错（不过T-T太深了，有很多地方我功力不够...），就个人而言我觉得在阐述优缺点和经验的地方如果能更深入一些或者说更细致一些就好了：)

个人觉得如果想掌握一种新的事物，要了解他的定义和后才可能知道这种事物怎么用，而经过对这种新事物应用并总结优缺点后，才可能真正开始了解这种事物的“价值观”，才可能无招胜有招。其实之所以提及这个，我觉得设计模式只是一种经验总结，换句话说，即使你没看过，也可能自己不知不觉的用过了，问题关键就是怎么去读懂设计模式的核心思想，因为最近很忙所以就选择仔细的读前6章，我认为前六章才是我的重中之重！如果你问我作者是否将这种精炼的思想提取并融入全书中，我暂时还没法回答，当然也可能是我太菜，不识庐山真面目，等等！难倒这就是作者题目所提及的禅吗？恩，等读后面我再好好的体会吧.....

嘿嘿，我仔细的看前言了，作为一本将自己多年工作经验总结，并通过写成书的方式来共享自己知识，单从这一点，哥们！额，不！前辈，菜鸟挺感动的，希望以后国内能看到更多类似这类书籍，而不是那些胡乱粘贴的垃圾书籍。呼，可以去睡觉了...

15、在我的印象里，技术类书籍一向是相当枯燥的，至少我之前看的一直是这样。眼睛死盯着码起来的文字一个个地啃下去，遇到难理解的地方，自己看不明白，往往还得回头再精读一遍，就是神仙也没了兴趣。学习本应是一个快乐的过程，相信那些技术类书籍的作者也不愿意看到读者把自己的著作看做一个个纸疙瘩，那就真杯具了。《设计模式之禅》（以下简称《设禅》），很厚的一本书，跟我之前看的一本相当枯燥的《Web信息架构》略厚。当时一拿到手，心想：完了，这要看到什么时候！我这人看书本身比较慢，非得要把书里面的每一个字都要读透（这是个不好的习惯）。但是拿起《设禅》真正读起来的时候，彻底打消了我之前的疑虑，书中通过一个个实例给我们讲了一个个编程的故事，平和的语言，滑稽的比喻，使得整个阅读过程都相当流畅，一点没有之前《Web信息架构》那般拖泥带水的感觉.....（惭愧，难为了那本很专业的书。）完全可以理解华章图书对这本书的重视，书的作者的确是花了脑筋的。作者的想象力和创造力也在书中有很好的体现，经常会在代码示例中加入一些很幽默的注释，例如“运行的时候开电梯门？你疯了！电梯不会给你开的”，“这是绝对合理的，只运行不停止还有谁敢坐这个电梯？！估计只有上帝了”等等。书中也经常把一个个编程对象比作成现实中的例子，犹如看小说一般行云流水。其实，每个软件在设计者的心中，都是一本...小说，一个故事.....能把深奥枯燥的理论知识变得风趣幽默通俗易懂，没有丰富的经验是不可能办到的，也体现了此书的难能可贵。阅读更多 &rsquo;

16、看过《Head First Design Pattern》一书之后，再对比下本书....还是前者更好些。无论是对知识点的



## 《设计模式之禅》

理解，还是讲解的方式都更好些。

不过，这本书名字更NB些。

17、 书是在再次读完 Head First Design Patterns 后读的，易于做横向比较，估计接下来会把《大话设计模式》也一并扫读了。

我是看完后随手把书评发到微博上，整理到这里，就不再添字了。

扫完「设计模式之禅」，读的是PDF版本，缺了几节。整体质量一般，最值得看就是对SOLID解说这一章。作者说读过 Head First Design Patterns，但不喜欢书中的西方幽默，但我觉得 Head First Design Patterns 这书更好。

Head First Design Patterns 较之「设计模式之禅」的优点：两者都在章节起始导入问题，但前者更深入地通过文字引导读者思路，并且伴随着教导读者些许测试驱动和重构的方法。此外，Head First Design Patterns 会比较类似模式之间的异同优劣。

Head First Design Patterns 没有说尽23个设计模式，它有针对性地选择、编排设计模式讲述的先后，并在设计模式的讲述中插入对设计原则的讲解。相反，「设计模式之禅」只是简单介绍各个模式，代码篇幅太多，甚至只给出代码就让读者自个领悟去。

引用「设计模式之禅」中的一句，看官自个评质量："策略模式的好处就是：体现了高内聚低耦合的特性呀，缺点嘛，这个那个，我回去再查查"、"这个就不多说了，自己领悟去"（第7页）。注：我看的是PDF版，与实体书可能有些许不同。但大体内容和风格应该是一致的。

总结：有闲情可以翻一翻，不值得入手。Head First Design Patterns 更适合用于了解设计模式。

这篇简评很中肯：<http://book.douban.com/review/3219343/>

18、这本书绝对是有深厚编程功底的人才能写出来的。作者相当的务实，他不是简单摆弄理论，而是用自己的实际经验，用相当切合实际的例子对理论进行深入且深刻的阐述。这些例子不复杂，但确实能够比较好的说明问题。相信作者在例子的选择上是很下了一番功夫的。最值得欣赏的是作者是真的做到了活学活用，而不是读死书。作者在对理论加以灵活运用的时候真正考虑了让人纠结的问题：度的问题。任何理论的运用都有一个度，不及或过之都是欠缺。而对度的把握则是难中之难，因为它没有可遵循的固定规律，只能依赖个人的经验和智慧综合考虑，这是任何机器都做不了的事情。相信大家能够从作者对度的把握上收益良多。阅读建议：1.如果有一定coding功底，建议看到作者举出的例子时，先自己加以重构/重新设计，然后再和作者改良后的代码进行比较。这样会收益多多。甚至可以在看到作者用汉字描述的例子时，就可以开始尝试用程序实现，进而作两次比较，一次代码抽象原型比较，一次是改善后的代码的比较。这样更锻炼人。2.尽信书则无书，适合别人的东西不一定适合你。所以用怀疑的态度去读书，尽可能尝试去推翻作者。这样你才能将作者写出来的东西变成你自己的东西，也就是做到活学活用。至于不足之处，呵呵，相对于这本书的优点而言，基本上都可以忽略，而且关于通俗和幽默本身也是一个度的问题，其间的斟酌也是相当费时，所以我也就不再鸡蛋里挑骨头了。（转读者书评，原文地址见《设计模式之禅》豆瓣页面）

19、作者喜欢讲故事，整体内容的模式都差不多。

20、 这书我是先在网上找到电子版看的。虽然不全，但是足以认定本书乃“十世修来的”好书。我认为好书需要三个条件：一作者有真才实学；二作者善于讲解和表达；三作者态度认真，力求打造经典。而这本书就具备全部条件。于是我买了实体书，还包了书皮儿...

至于书的内容嘛，讲设计模式呗。这东西的重要性需要强调吗？如果你觉得需要那你别犹豫了赶快找本书来恶补吧，这本就不错。

推荐大家都来看看，绝对不亏的。

## 《设计模式之禅》

21、 简单、通俗、易懂，但又不肤浅，这是本书的最大特色。尤为值得一提的是，本书还有设计模式PK和混编设计模式两大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“四人帮”的那本书也不例外。

（来自卓越）

22、 哟，平心而论，算是国人写的一本不错的书！

23、

对与设计模式的认识源于2年前的一本书《设计模式》，GoF（“四人帮”，指Gamma, Helm, Johnson & Vlissides, Addison-Wesley四人）的《设计模式》第一次将设计模式提升到理论高度，并将之规范化，提出了23种基本设计模式，自此，在可复用面向对象软件的发展过程中，新的大量的设计模式不断出现。

但是对于一个初学者来说，这本书毕竟是比较专业化的视角来描述，难免有些苦涩难懂，自从前段时间在51cto 读书频道，接触到“设计模式之禅”这本书后，使我的想法完全发生了改变，作者以亲切自然的风格阐述了设计模式的核心思想，潜移默化地提升可我们面向对象的架构和编程能力，带我们进入了“物我合一，见性成佛”的最高设计境界。全书以简单通俗易懂但又不肤浅的描写方法，让我们在静静的思考中，慢慢的会意中，把设计模式的思维无声地融入自己的思维中。

就像作者在前言中所说：“会看的看门道，不会看的看个热闹”本书对无论有无编程经验的人来说都是一本书中的尚品。对于经验欠缺的编程人员，完全可以把它当成一本小书看。

书中通过对通过对六大设计原则的全新解读、23种设计模式的完美演绎、设计模式的pk、设计模式混编等，向我们展示了作者的想象力，创造力和对设计模式的深刻理解，通过具体的开发场景，以讲故事的方式向我们阐释了设计模式的灵魂思想。

通过这两天的阅读，建议大家反复研读，必将使自己的设计思想得以升华，使我们再次学习设计模式之时，领悟到“禅”的境界，禅的思想。

总之，这本书无论对初学者也好，资深编程人员也好，都是一本难得的好书，在细细玩味之后，在潜移默化之中，我们对设计模式的理解必将富有禅意！

就和大家分享这么多，希望大家在读这本书时，能共同交流，共同提高，是我们无论对技术的学习还是对生活的领悟都有一个更高的境界。（我的邮箱daiyuxi110@sina.com，希望和大家分享！）

24、 食之无味，弃之可惜啊

25、 本书引用许多实例，清楚描述了每个模式的应场景。并且还对不同模式间应用在同一问题进行比较，体现了作者对各模式深刻透彻的理解。

26、 原来借阅过阎宏的《Java与模式》，当初看时，刚入门，对模式设计的兴趣不大，而且对传统文化了解浅薄，感觉读不懂，就放下了。

所以，拿到这本书，看到名字的时候，就是怕自己再次看不懂。但在认真的看了几天之后，就明显感觉这本书讲述的比较朴实，比较结合现实。

工作了这么多年，或多或少的用到了些模式，但对它们之间区别和运用往往不够重视，所以开发中往往犹如芒刺在背，本书开头用了六章来讲解最基本原则。

本书定位读者应该是对模式有所了解和项目联合开发经验的，不过没有这些经验，也可以作为指导性书籍来看。

我所在公司不大，项目基本都是单兵作战，往往是一个人同时扛多个项目，而每个项目的参与者顶多不过3-5人（包括页面设计和客服等），开发上往往只有一两个人，所以，接口用的不多，就是有也放着不用。但看了这本书以后，感觉受益良多，发现很多东西如果设计的好，可以避免重复工作的浪费。

设计模式，不是说看过明白了就可以把书扔掉，此书也介绍了一下常见的组合模式，看起来，就不费劲。推荐给大家此书，时常看看。

另外，特别感谢哲思社区和华章公司，给我这个机会做此书的评论员，书评有点晚了，非常抱歉！

27、 比较简单易懂，个人感觉水平比大话设计模式高

28、 不过 评价不错。给个好评

# 《设计模式之禅》

29、用来理解设计模式很好。

30、刚知道自己有机会拿到这本书时，真是喜出望外，倍感兴奋。因为以前下了n次的决心想学设计模式，四人帮的《设计模式》也从图书馆借了换，换了又借。一次都没看完过，并且都是翻了几页就放下了，感觉设计模式太可怕了。但他却又是程序员不得不学的东西，于是为之深感烦恼。有时候时常安慰自己说“现在还没到学习设计模式的时候呢，等做过几个项目再去看自然也就懂了。”说是这样说的，那只不过是安慰自己的话罢了，仍想早些了解他，并能很好的运用它。这时候我极想能找到一本设计模式入门的书籍，先对设计模式有一个初步的了解，让后再去看那所谓的经典，但一直没找到合适的。就在这愁人之际看到了这本书，一下子就被这本书的介绍吸引住了：

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。

这介绍正合我意，心想我终于找到了它。我终于可以参悟设计模式中的奥秘了。之后便是每天都在期待着这本书的来临，可是之后一个月都没拿到书，那可恨的邮局啊。还好华章的张艳又赶紧快递了一本给我。真是太感谢她了。虽然收到了书，但此时已步入毕业设计的繁忙时期，白天工作，晚上回来做毕业设计，这本书只有等熄灯后每天看上半个小时了。到今天已收到书恰好两周了，不过至今仍没看完，但我已了解本书的写作风格，下面说说我对这本书的感受吧：

本书使用JAVA语言描述的小例子向大家展示了6大设计原则和23中设计模式。之后又做了扩展来了个设计模式大PK，更令人欣喜的是最后一部分讲解设计模式在实际中的运用，若真没有这一章，我还真对本书有点小失望呢。呵呵。

本书的优点：

## 1 纸张质量

这本书的质量没得说，华丽精美的封面，书的纸张也相当的不错，绝对适合收藏。

## 2 读者范围广

这本书的读者范围可谓相当广，不管您有没有编程经验，这本书您都可以去读，就像作者说的，没有编程经验的您可以把它当本小说去读。但说是这么说，我相信有一定的编程基础的同学都能看懂些。

## 3 通俗易懂，风趣幽默

作者之所以敢说没有编程经验的同学可以把本书可以当小说读，完全在于作者那章章节节的小故事，非常风趣幽默，能完全吸引住读者跟着作者的思路走，再加上作者使用相当简单的程序去描述每个模式，非常通俗易懂，我相信只要有点编程经验的人都能看懂一些的。

## 4 讲解细致

相比200多页的“四人帮”经典，作者可谓大方之极，看来“惜墨如金”这四个字在作者身上没起作用哈。^\_^，作者结合实例细致讲解了每个设计模式，之后仍感觉不过瘾，又来了个设计模式大PK，通过PK能读者能更好的理解各个设计模式。PK完之后作者貌似体力仍十足有余啊，又从实际编程的角度给大家讲解设计模式的综合运用，真是一盘绝顶大餐啊。正和我口味，呵呵。

# 《设计模式之禅》

本书的缺点：

俗话说世上没有完美的东西，如果完美了，他肯定不是来自地球。呵呵！而每个人对美丑的欣赏角度也不一样，观点定不会统一，我认为本书的缺点是：

1 看本书需要一定的UML基础，这点真是有点令人不悦，虽然我是学C\C++的，对JAVA的了解为0，但代码我基本上都能看懂，只可惜作者的类图有些实在让人摸不着头脑，有些类图画的是在让我搞不懂有些类的关系到底是什么，目前实在懒得去看UML了，我想作者能对此进行一下解释就更好了，我认为在这个地方，作者设计好小例子之后，可以专门拿出一小节根据类图讲述类之间的关系，并由此引申一下设计模式，我想会更好。如果所有的东西都让读者从代码中看的话，没有编程经验的确实也只能看个热闹了，只能留给有2年以上工作经验的人看了。

2 我感觉作者的例子可以在深入一些，我感觉有些例子稍微有些牵强，能深入些会更好。

总之，这本书是一本学习设计模式的很好书籍，如果您还觉得设计模式枯燥无味，难学之极，那请你来看本书吧，它能在你不知不觉中把你带入设计模式的领地，并不断浸润设计模式之精华。看完本书并做一些练习之后，我相信你不但不会再对设计模式有所恐怖，反而还可以能在自己的项目中运用自如的使用设计模式，使自己的代码设计更加灵活。

31、比较容易理解，尤其是穿插着各种段子

32、Java 代码看醉了、、、

33、个人觉得这本书没有把设计模式的内在讲清楚，与《大话设计模式》相比差了许多。

34、还是挺不错的

35、浅显易懂，不够深入

36、读技术如小说。。

37、电子书

38、这几年来，前后读过3本设计模式的书：

1. 《大话设计模式》

2. 《HeadFirst设计模式》

3. 《设计模式之禅》

前两本书出版的时间比较早，也比较畅销，于是我都买了，总体而言，我能给《大话设计模式》打60分，因为它有的仅仅只是对设计模式的解读，而且是最基础部分的解读，试图用一些很能让读者容易懂的方式去解读，但在我看来，这方面其实做得并不好。缺乏比较核心的内容。我能给

《HeadFirst设计模式》打75分，这本书在设计模式领域还是相当有地位的，因为它开创了讲解和学习设计模式的新方法，这本书的优点就在于通俗易懂，精准到位，不足之处在于没有能很好地把这23种设计模式进行融会贯通和综合比较，也没有太多关于如何才能更好地应用这些设计模式的讲解，属于入门价的，对提高没有太多帮助。

《设计模式之禅》算是后起之秀了，这本书刚出版时在社区里引起了轰动，身边的同事也有几个人买了，我借来看了看，一下子被吸引住了，于是自己也买了1本，到现在为止，我已经看了2遍了，感觉自己设计模式的理解加深了很多。这本书几乎继承了前面两本书的所有优点，而且避免了前面两本书的不足。

强烈向想学习设计模式的朋友推荐这本《设计模式之禅》

39、很言过其实的一本书。

第一：作者说了，是用咱们的母语讲解设计模式的书，可是每次下定义的时候都先用英文下，然后再用母语重复一遍，估计是为了凑字数的。

## 《设计模式之禅》

建议：如果能看懂这本书中的英文，建议直接看HeadFirst Design pattern原版,该书比本书至少要好三个档次。如果看不懂本书的英文，请看HeadFirst Design pattern的中文版，虽然有很多经典被翻译糟蹋了，但HeadFirst Design pattern翻译的还是不错的。

书中错误百出，我昨天稍微翻了翻，就发现几处很明显的错误：

例如第11页，“直接把sanMao.killEnemy(new Rifle())改为sanMao.killEnemy(new MachineGun()),”简直不知所云，原来每开一枪就要换一把枪。

第217页，“你要一个研究生，你派了一个高中生给我，那算什么事”，第一个你应该是我吧，要不你要一个研究生和你派了一个高中生给我有什么关系。

还有一些明显的错别字。

当然，这种情况的主要责任在编辑。建议以后编辑将主要精力放在提高书籍质量，避免低级错误身上。不要光顾吹牛，设计模式的里程碑，按照我的看法只能是四人帮的原著，虽然该书有些晦涩，而且翻译的很糟糕，但里程碑就是里程碑，不是随便挑块石头就可以当里程碑的。

40、很不错的书啦，谢谢

41、挺一般的，没买的就不推荐了。

42、总的结构好像跟《大话设计模式》差不多，设计模式都是yy家忽悠用的，其实那些技巧都是实战中搞出来的，一般没有太大必要提前设计成那个鸟样，但是需要在实践编码过程中体现这些思路。书籍总的来说还不错，给4星，最后几章太冗长，读起来很费劲，第三部分的不同模式pk较为深刻，想了一想原来我们mp前端框架用了好多这样的思路，虽然完全没考虑过设计模式，这个就是经验，哇哈

43、书写的通俗易懂，有作者自己的思考，值得一读~

44、送货速度很快，很方便，比书店买书好。

45、实例丰富。全书545页，很少连着两面或三页只见文字不见代码，实际上基本每页都有代码。这些例子是作者九年的工作总结啊，其价值是不言而喻的！

通俗易懂。先是提出问题，给出一个相对简单的解决方案，然后不断完善，循序渐进，层层深入，比如在讲工厂模式时，先由女娲造人的神话引出一个简单工厂的模式，然后根据变化过渡到多个工厂类的模式，再到抽象工厂，用贴近生活的例子和语言娓娓道出高深的模式，让人感觉自己就像在读白居易的诗一样，让一个不懂设计的人感受到了学习与设计的快乐。

客观、全面。在由浅入深地给出相对完善的模式后，还从优缺两面进行总结，能让人有更深的了解。同时，该书的讲解是很全面的。比如将唯一的单例扩展成固定数量的“单例”，再如原型模式中，讲解了浅拷贝与深拷贝后，还讲解了String的特殊性以及clone与final的冲突等，真是知无不言，言无不尽！

来于现实，最后又回归现实。来书每章开始都从生活中常见事例引出某类经典问题，当我们怀着好奇之心一步步地看完解决方案后，发现自己已经又学习一种模式了，此时再去看那相对“抽象”的模式定义，已经感觉不到抽象了。最后，在每章的结束部分，又给出了该运用该模式的具体场景，将读者的思绪从定义又拉回了现实，给了我们巨大的思考与想象的空间。这也正是我们学习模式的原因啊！

46、在一小段时间前，有点空闲时间，想学习一下设计模式。几年前就一直听“高人”们提到设计模式，当时也着实想认真研习一下，并认为不明白设计模式，始终是比较瘸腿。但找到的读物都令人费解，读起来很吃力，所以每次都中途而废。

在网络上无意中发现了秦小波先生关于设计模式的一个PDF，感觉写的非常好，很对胃口，有眼前一亮的感觉。然后追根溯源就找到了这本《设计模式之禅》。这本书相对于其他讲模式的书，还是非常容易理解的，也比较容易记住。

这本书让我重新拾回学习设计模式的信心，非常感谢作者。

我感觉这本书可以这样读并从而学习设计模式：

1、像读小说一样耐心读完，读的过程中要集中精力，尽量记住每一种模式的描述

## 《设计模式之禅》

- 2、对照目录，认真回忆每种模式是怎么描述的，用于何种场景
- 3、对于不能记清楚的模式，立刻找到对应章节快速浏览，从而加强记忆
- 4、有几个模式会容易记混，所以要认真比较和加强记忆
- 5、书本只是对模式进行了容易理解描述，关键还在于理解每个模式的思想，然后使用于自己的设计中
- 6、针对每个模式，都重新举个例子，并用代码实现，从而更加强化记忆和理解
- 7、书中大部分的例子，都是先写不完全正确的例子，然后循序渐进，给出正确的例子。应该增加一个特殊的索引，直接指向正确的例子，便于读过的人可以快速定位。不过这本书很不错的一个做法是在最后提供了一个很大篇幅的关于各个模式的彩页，我觉得个人可以在这个彩页上直接标记出每个模式对应的页号，这样以后就方便快速索引了。
- 8、对于手头有源码并且设计复杂的系统或者软件，可以对照学习，对号入座，从而也可以加强理解。久而久之，以后再“看高人”的代码就再也不迷惑了，可以一目十行的快速阅读代码了。特别是一些架构级别的代码。

47、前一段时间终于领到了我期待已久的《设计模式之禅》一书，但是由于工作的原因，一直没有时间静下心来细细品味作者那些来自自己工作实践中的禅语。我把这本书放在我的床前，每当我临睡前，我都会翻翻此书。在此之前我也看过一些设计模式方面的书，但是看看就想睡觉。有些设计模式的书看的是云里雾里的，看完了都不知道人家说的是什么。

《设计模式之禅》这本书更多的是从实践和应用的角度来讲设计模式，作者根据自己的项目实践总结出一些使用设计模式的方法和应注意事项。再加上作者通俗易懂的语言描述，读起这本书来就像读武侠小说一样过瘾。翻看了一下本书的目录，我觉得本书不仅仅是对23种设计模式的解读，更多的是对这23种设计模式进行比较，比较各设计模式的优缺点，以便在实际应用中进行合理的使用。

《设计模式之禅》确实是一本值得读的好书，我会好好的读这本书的。

48、在我看来，这是一本讲解设计模式不错的书。例子都很通俗易懂。非常不错。只是设计模式有些多，要经过运行真的不容易。看这本书我真正明白软件是现实世界的抽象，请多用软件思维来实现现实世界

49、相当不错，作者讲的很好

50、看的是PDF版本直接看了关键词设计禅开始读，才发现原来是写软件开发架构和模式滴，第一次接触到的我还是被吸引，第一，作为IT编程小白竟然可以看的下去可见通俗易懂有趣有趣是第一动力切记切记！第二，逻辑清晰架构原则其实平时生活工作也能贯通，足够了。

51、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 and 低维护成本。

（来自卓越）

52、刚拿到书，看的不多，感觉该书与其他书的一个最大的区别就在于语言的通俗易懂。书中的例子也大多是平时开发工作中遇到的，感觉很亲切，作者将设计模式的禅潜移默化的融入在简单的语言和生动的实例中，不是说教，不会让你感觉枯燥，我觉得这起码是好书的一个标准，禅，其实就这么简单！（转互动网读者评论）

53、很喜欢的一本书，比任何看到过的感觉都好，个人感觉，比本书的第二版更好！！

54、设计模式，无论看多少次都很难入脑。因为现在快速开发框架太多，很少有机会能用到这些；但这不代表它们可以忽略

55、三分，不能再多了。感觉跟原本比没太多信息量，而且无禅意。

56、在我的印象里，技术类书籍一向是相当枯燥的，至少我之前看的一直是这样。眼睛死盯着码起来的文字一个个地啃下去，遇到难理解的地方，自己看不明白，往往还得回头再精读一遍，就是神

## 《设计模式之禅》

仙也没了兴趣。学习本应是一个快乐的过程，相信那些技术类书籍的作者也不愿意看到读者把自己的著作看做一个个纸疙瘩，那就真杯具了。

《设计模式之禅》（以下简称《设禅》），很厚的一本书，跟我之前看的一本相当枯燥的《Web信息架构》略厚。当时一拿到手，心想：完了，这要看到什么时候！我这人看书本身比较慢，非得要把书里面的每一个字都要读透（这是个不好的习惯）。但是拿起《设禅》真正读起来的时候，彻底打消了我之前的疑虑，书中通过一个个实例给我们讲了一个个编程的故事，平和的语言，滑稽的比喻，使得整个阅读过程都相当流畅，一点没有之前《Web信息架构》那般拖泥带水的感觉……（惭愧，难为了那本很专业的书。）

完全可以理解华章图书对这本书的重视，书的作者的确是花了脑筋的。作者的想象力和创造力也在书中有很好的体现，经常会在代码示例中加入一些很幽默的注释，例如“运行的时候开电梯门？你疯了！电梯不会给你开的”，“这是绝对合理的，只运行不停止还有谁敢坐这个电梯？！估计只有上帝了”等等。书中也经常把一个个编程对象比作成现实中的例子，犹如看小说一般行云流水。其实，每个软件在设计者的心中，都是一本小说，一个故事……

能把深奥枯燥的理论知识变得风趣幽默通俗易懂，没有丰富的经验是不可能办到的，也体现了此书的难能可贵。

57、 “设计模式领域又一里程碑之作”，这个评价夸大其词了。买了这本书，还是有点后悔的。如果需要入门设计模式，还是看那本head first（例子java写的）更好一点。

然后四人帮的那本经典，也是必读的。

这本书名字起的很大气，但是内容并没有达到这个高度。

58、简单、通俗，但又不肤浅，是本书的最大特色。尤为值得一提的是，本书还有设计模式PK和混编设计模式两大部分内容讲述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“四人帮”的那本书也不例外。

59、 本来应该早些写出来的，不过最近家里杂事太多，一直抽不出时间了，所以耽搁了，还请大家见谅。

活动里面说提倡原创，吓得我都不敢看其他已经贴出来的书评，生怕自己的思路受到影响。

好了，言归正传。

最深印象：

这本书绝对是有深厚编程功底的人才能写出来的。作者相当的务实，他不是简单摆弄理论，而是用自己的实际经验，用相当切合实际的例子对理论进行深入且深刻的阐述。这些例子不复杂，但确实能够比较好的说明问题。相信作者在例子的选择上是很下了一番功夫的。最值得欣赏的是作者是真正的做到了活学活用，而不是读死书。作者在对理论加以灵活运用的时候真正考虑了让人纠结的问题：度的问题。任何理论的运用都有一个度，不及或过之都是欠缺。而对度的把握则是难中之难，因为它没有可遵循的固定规律，只能依赖个人的经验和智慧综合考虑，这是任何机器都做不了的事情。相信大家能够从作者对度的把握上收益良多。

阅读建议：

1.如果有一定coding功底，建议看到作者举出的例子时，先自己加以重构/重新设计，然后再和作者改良后的代码进行比较。这样会收益多多。甚至可以在看到作者用汉字描述的例子时，就可以开始尝试用程序实现，进而作两次比较，一次代码抽象原型比较，一次是改善后的代码的比较。这样更锻炼人。

2.尽信书则无书，适合别人的东西不一定适合你。所以用怀疑的态度去读书，尽可能尝试去推翻作者。这样你才能将作者写出来的东西变成你自己的东西，也就是做到活学活用。

至于不足之处，呵呵，相对于这本书的优点而言，基本上都可以忽略，而且关于通俗和幽默本身也是一个度的问题，其间的斟酌也是相当费时，所以我也就不再鸡蛋里挑骨头了。

60、 pass

61、一般，没有想象的那么好，里面的例子看的我也是醉了，生搬硬套

## 《设计模式之禅》

62、可能是作者的疏漏吧。

63、其他都不错有些地方略显啰嗦

64、说实在的，我根本不懂设计模式，本来看到好多人在力荐这本书，心里就痒的难受，但是，“书，不读，如废纸”，心里还是隐隐的告诉自己“再想想，买那么多书了，有几本彻底的看完过！”

恰巧有pdf格式的下，下来看看吧，再次说实在啊，看这本书的前言“如何阅读本书”，一个字“贼拉的震撼”，银生第二次看到发自内心的话语啊，而不是商人般的唯好必说、逢疵必让，第一次是看《21天学会C++》，作者前言说道“没有任何一个人能够在21天内掌握一门结构严谨的语言”，人家把道都给你指明了。第二次，就是本书，为什么本书会让我娇弱的小心眼儿为之震撼呢？我说过，我是一只小鸟，小phper，知道成长的艰辛，绞尽脑汁去参悟作者的意思、看花了眼睛去阅读手册，成之路真是辛苦啊！可能是受尽以往资料生硬语言的束缚，本书叙事方式和以往书籍截然相反，举例说明：大家都有几个要好的朋友吧，和朋友谈话时候什么感觉，和朋友谈心时什么感觉，和朋友扯淡是什么感觉？看这本书就是啥感觉，还是一个字“贼舒服”，换句话，你和杨教授聊天，啥感觉？(你和杨教授关系老铁了，另当别论)；

嘿嘿，上学的时候语文没学好，想到哪说到哪，倒是全部发自内心，谢谢：)

65、转卓越网读者评论：

是程序员都知道设计模式很重要，几乎所有还没有悟透设计模式的程序员都指导设计模式很难搞，学习的成本非常高，过来人都说要一边实践，一边理解，然后一边总结。

关于设计模式的书非常多，有GOF的，也有其他解读版的，我个人在学习设计模式的过程中也接触过几本，比如GOF的《设计模式》，书是很经典，但是太难懂了；《Head First设计模式》也读过了，还不错；《大话设计模式》在书店里也翻了翻，能打个60分吧，远没有大家说的那么好；《设计模式之禅》这本书我读了快一半了，应该说的确弥补了这几本书的不足之处，浅显易读但是又不肤浅是这本书最大的特点，能从书中设计的案例看得出来作者的道行比较深，而且写这本书时很用心，实在是读者的福气。

建议想学设计模式的朋友首选这本书，应该不会让你失望。

66、读这本书的起因源于csdn学生大本营的一次活动《设计模式之禅》试读员招募，身为程序员兼之学生大本营的老师没有道理不踊跃参加了（参加时可没走任何后门），佛祖显灵，真的能有幸成为了试读员。从得知寄出样书，居然用了半个多月才拿到手，长了些。却也将我急切等待读书的浮躁心情洗涤为一种平和的心情，此时读这本《设计模式之禅》未尝不是一件好事，正应了“禅”的心境了。

翻开书 赞誉映入眼帘，这么多还是直奔主题吧，我这人历来读书都是先看目录，读懂目录，书也就有机会读懂了。在参加活动时，在秦少波老师主页中，简单阅读了部分样章（我这人比较懒，看书还可以，让我在电脑上看书向来懒惰），作者将要阐述的知识分成了四大块。从面向对象的6大设计原则入手，逐步演绎了GOF23个设计模式的一招一式，到后来更是演练了一番海陆空联合作战（设计模式混编）。每个设计模式用一个小故事引出该模式的定义，随之展开、深入。结构分块清晰，正是我喜欢并常建议我的学生学习时采用的学习和思考方式，由原则规矩入手- 一招一式- 组合拳。

清楚了这本书的内容的组织结构，闭眼那些知识走马灯似在脑中过了一遍，设计模式就是这样啊，禅之何来。往后看下来，不对，不同。章节中深入浅出的故事让初学者对设计模式通俗易懂；几句话的定义又将知识点透彻清晰的总结了；而随之的应用、经验技巧正是唐三藏要求取的真经啊！！

快速翻过跳到“第三部分设计模式PK”，慢慢的读下来，PK相似、PK不同，不是调侃似的PK，



## 《设计模式之禅》

不是理论似的PK，好好好，设计模式在心中越来越清晰的勾画出来。书读到这里还没完，作者居然在最后的最后增加了几个“开发项目”。好生动的演练。通过前边知识的准备，潜移默化的深入再深入地探讨了几个主要模式。

阳春白雪下里巴人。好佩服作者深厚的编程开发功力和文字处理能力，居然在一部书里层次清晰的实现了雅俗共赏。再次闭目冥想，设计模式如春雪入土，慢慢消溶；土中有雪，雪中有土；不着于痕迹——何时悟禅？

初读，事多，时间紧，未及细读，很多未及文字，不通处见谅，欲知后事，再读，再再读有感时……

67、我在读第二遍

68、书中通过对通过对六大设计原则的全新解读、23种设计模式的完美演绎、设计模式的pk、设计模式混编等，向我们展示了作者的想象力，创造力和对设计模式的深刻理解，通过具体的开发场景，以讲故事的方式向我们阐释了设计模式的灵魂思想。通过这两天的阅读，建议大家反复研读，必将使自己的设计思想得以升华，使我们再次学习设计模式之时，领悟到“禅”的境界，禅的思想。总之，这本书无论对初学者也好，资深编程人员也好，都是一本难得的好书，在细细玩味之后，在潜移默化之中，我们对设计模式的理解必将富有禅意！

69、纸质书第16页下面刚改的代码，在17页仍然输出“父类被执行”而不是“Map转Collection被执行”，难道我理解有误？

70、窃以为设计模式之禅写得很好，很幽默，pdf版的

设计模式也是要一个一个地写代码的，光看没啥作用

71、兄台，我平时不用，这篇评论是转的卓越网一位读者的评论。

72、衍生书目 中规中矩

73、还不错，先读大话，再读这个，挺好

74、深入浅出将设计模式！推荐

75、代码太多了，特别是BO代码，完全没必要，唯一的好处就是让人有看书一目十行的感觉。。

另外，作者废话挺多，有时候对基础婆婆妈妈，比如讲组合模式的时候还介绍什么是树叶，什么是根，有时候却对关键的内容一笔带过，比如在第一次讲到ssh的时候，连个简称说明都没有。

不过，书中的例子都还是不错的，最后两章关于不同模式的对比和混合写的也很好，相信作者还是很费心思的。

76、从程序员到初涉软件架构设计，对于我，“设计模式”仿佛从“Hello World”开始就是一个谜，面对完整的项目，复杂的需求，如何设计模型，这的确是一个问题。完美的设计出自对模式的理解和丰富的项目经验，不得不承认，这需要我们花费很多时间和精力于此，并付诸于实践中去。但是最近笔者阅读了机械工业出版社的《设计模式之禅》一书，对设计模式的“禅理”娓娓道来，并且有很多精彩的例子做辅，语言平淡直白，通俗易懂，就像是在与深谙其道的朋友聊天。纵览全书，本书分为四部分：设计原则，针对六个设计原则进行阐述，其中不乏经验之谈；设计模式演绎，分别描述了23中设计模式，并且每一种都有例程在里边，每种模式应用的场景被形象的描述出来，比如第十六章的责任链模式，形象的用我国古代“三从四德”来举例，描述责任链模式是怎样应用的，等；设计模式比较，分别就创建类模式，结构类模式和行为类模式等进行PK，设计模式混编，列举几例模式嫁接应用，和MVC框架。总之，该书语言平实，实例精准，内容翔实，涵盖面广，是软件设计模式中不可多得的一本书。简单的几行文字不足以概括其精妙，还需细细品味。

77、例子过于简单了，适合入门的读读。有些模式讲解的太牵强，也就前四章对原则的讲解还有点用

78、回顾性的看得一本书温故而知新

## 79、 1. 综合评论

### 【一句话总结】

值得一读。比大话系列严谨，比GOF圣经易懂。69块钱，24小时，划算。

### 【各部分感受】

第一部分，六大原则，及其受用，适用于程序开发也适用于做人做事。书中有大量生动活泼的故事，有些十分贴切，想必作者费了不少脑汁。

第二部分，对GOF的模式以有趣的方式庖丁解牛了一番，有些很独到。初学者上手快，已悉者温故知新。有趣味，有过程，有血肉。

第三部分，PK很有特色，巩固知识，加深印象，消食通便。不打不相识，越打越亲近。条条大道通罗马，风景各不同。

第四部分，合作共赢，综合应用，凝聚开发者的智慧。重点看需求上下文和程序架构，模式名字已不重要。

附录的23种设计模式类图，是杀人越货之必备阿。

### 【阅读建议】

每节的【最佳实践】都应当理智阅读，原则也，实践也。

演示代码只是为了说明问题，在实际项目中采用，要斟酌。

建议阅读顺序，四一二三四一。

阅读目标，忘记模式吧，融入场景，心中无刀。

## 2. 事件考古

【2010年03月12日 五】看分享，《设计模式之禅》试评员招募

【2010年03月13日 六】凑热闹，“申请，看看什么是禅”

【2010年03月19日 五】出结果，直到莹美女分享才知道。

【2010年03月22日 一】发邮件，“哲思-设计模式之禅-试评员-登记”

【2010年03月23日 二】收邮件，恭喜您成为华章公司《设计模式之禅》试评员。

【2010年04月02日 五】第一篇，哲思杨某侠书评出炉。

【2010年04月08日 四】怕误事，邮件确认：样书已于3月25日寄出，平邮。

【2010年04月15日 四】样书到，雯美女亲临。

## 3. 试评计划

设计模式之禅/秦小波/机械工业出版社/ISBN978-7-111-29544-0/545页/69元

试评员约定要在样书到手的两周内出书评，500字以上，发布于网上。

作者学机械的，9年技术，儿子三岁。豆腐学化工的，7年技术，儿子175天。

前辈啊，O( \_ )O哈哈~，这书有点啃头，有计划有步骤的耐心嚼之。

尽信书不如无书，豆腐读书，绝对是批判的继承，批判是我思考，不直接反映书的品质。

和买东西一样，挑刺越多，越易成交，满口好好的，可能路过。

白天上班，晚上亲子。整块的时间不多，遂分而治之，评一次为一里程，路标如下。

【C1.1】即第1.1节。(C)hapter,为章节标记。

【P123】即第123页。(P)age,为页数标记。

【BTW】随便说一下。(by the way)

【小结】小小的总结一下。

# 《设计模式之禅》

## 4. 第一里程

【2010年04月15日 四 18:30】

【书名】禅者，心也。有点玄。要是蝉就好了，知了也。

【封皮】诗经·小雅·鹤鸣“它山之石，可以攻玉。”

【作者】交行阿，信用卡用他家的，网银有待改进。

【赞誉】很多人很高评价，还有阿福，有点看头。

【前言】挺实在。致谢那段有意思。23种模式都用过，是否存在过度设计。

【C1.1】info隐喻是属性，不应实现Biz行为接口，见仁见智吧。

【C1.2】IPhone的例子很好，‘注意’‘学究’，提示背景和视角。

【C1.3】我单纯，所以我快乐。

【C1.4】很现实的问题，多快好省，服从领导。

【C2.1】继承必须拥有父类的所有属性和方法。private的呢，从哪个角度讲。

【C2.2】OO5大原则之里氏替换，09图灵奖女得主。例子很好，尤其CS那个。

【2010年04月15日 四 21:00】

## 5. 第二里程

【2010年04月16日 五 18:08】

【P24】表面类型和实际类型，头一次听说，为何不用大众叫法。

【C3.x】DIP，真的是讲了不少。

【C4.1】实例接口，这种讲法，长见识。

【C4.2】星探找美女，不如改成相亲，更有杀伤力。

【C4.3】IBookSearcher例子不错。

【C5.1】你知道的太多了，所以越单纯越好。

【C5.2】command通假command？4层含义讲的很透彻。

【BTW】不太习惯，下划线开始的变量。

【2010年04月16日 五 21:30】

## 6. 第三里程

【2010年04月17日 六 12:11】

【C6.x】开闭原则，拥抱变化，第一思考的原则。

【小结】

六大原则很是受用，程序员进化之必备。例子多样，很大众化。

建议增设第七原则，就是奥康姆剃刀----“如无必要，勿增实体”。

【P59】say()还是不要static了吧。

【P60】构造函数private只是确保了非本类不能new，而不是仅产生一个实例。

【P60】“类中其他方法，尽量是static”，public static违反LoD，隐患多。

【C7.3】对单例讲的很全面。除了clone外，反序列化也值得注意。

【C7.4】场景假设的好，还普及历史知识，但代码有点不妥。

【P62】countNumOfEmperor，static太糟糕，钰有时会说他是镇，都不用多线程。

【P63】直接Emperor.say()试试看。

【C7.5】单例会被JVM的GC么？何种情况，理论依据或证据呢？豆腐认为有Ref就不GC。

【C8.1】大话女娲造人的故事比较有趣。

【P67】”其中的’？‘表示的是，……“没看到代码里使用。言之何物？

【P67】Class.forName(c.getName()).newInstance();为何不直接c.newInstance();

【P67】使用泛型T，为何要Human强制转换一下呢。

【P77】Map<String,Product> prMap = new HashMap();建议HashMap parameterized

【C8.x】GOF说的很好。此节的例子不太恰当，有点为了工厂而工厂。

【C9.x】略读。还是女娲娘娘这块比较有趣，想必费了不少脑筋。

【2010年04月17日 六 15:20】

## 7. 第四里程

【2010年04月17日 六 18:20】

【C10.1】本小三，纯属虚构。

【C10.2】final防覆盖，这个提示很到位。

【C10.x】模板这块讲的很细腻。记得JIC系统，就问过袜子，如何限定子类行为。

【C11.1】变化是永恒的，Builder+Templet的例子很有代表性，值得研习。

【P108】显式调用Collection的clear()，不仅可避免意外惊喜，还可避免泄漏。

【小结】模板和创建这两节非常好，一个场景贯穿，一气呵成。

【C12.x】用游戏代练类比Proxy，用“审计”点出AOP，本节的看点。

【P146】this.arrayList改成thing.arrayList，笔误。

【C13.4.3】冤家是因为final赋值，浅clone可以，深clone曲线救国也是可以的。

【C13.x】电子账单的场景有来路，带有实际业务的影子，比虚构的系列好很多。

【2010年04月17日 六 21:33】

## 8. 第五里程

【2010年04月18日 日 08:00】

【C14.1】两个“库存情况”。后者应该是“采购情况”。

【P148】“折半采购”少了stock.increase(buyNumber);

【C14.x】贴近生活，容易理解。

【P170】命令模式通用类图，Client关联Receiver还是Invoker？

【C15.5】原来伏笔在此揭开，算我读的很仔细。

【C15.x】有别于GOF，从新的角度讲解了Command模式。

【C16.x】责任链，全体妇女同志站起来……了。

【2010年04月18日 日 09:30】

## 9. 第六里程

【2010年04月18日 日 13:00】

【C17.x】成绩单大话的很热闹，JDK中例子很多。

【C18.x】略读。策略枚举慎用。

【C19.x】略读。RMI慎用。

【C20.x】如书中所说，太普遍了。研究研究Collection框架有好处。

【C21.x】组合模式这么讲有点晕。树这么整不合适。

【C22.x】先看反面例子，再看注意事项，再研究“暴露狂”。

【C23.1】letterInotoEnvelope() 通假 Into。

【C23.x】略读。应该谈谈JDBC。

【2010年04月18日 日 14:40】

## 10. 第七里程

【2010年04月18日 日 17:30】

【P307】宽接口，窄接口，讲的很好。

【C24.x】备忘录，有用的东西。注意不同场景下的策略略和细节。

【P317】图25-5，这个类图怪怪的，method首字大写。

【BTW】类图有些没标返回值，图25-6，25-5，25-4。

【P328】双反派，双分派。

【C25.x】以邻居访问为故事讲解的很好。

【BTW】很多方法论，都有很贴切的例子，在生活中活灵活现。

【P340】代码26-13，还是setLiftState(Context.closeingState);好。

【BTW】closeing,openning这两个ing很smilence(笑而不语)。

【C26.x】电梯的例子很好，明了贴切，值得细品。

【C27.x】堆栈计算器，适合学习。

【C28.1】对厂商的分析工具感兴趣。

【P360】工厂是200多个并发，咋没进行线程安全控制呢。

【P365】讲了。

【P369】10万次，多按了2个零 10000000

【C28.x】故事场景不错。对象在内存的大小可以通过成员变量估算的。

【P378】眼中无黑体部分，心中有黑体部分。印丢了，呵呵。

【C30.x】又是一个很有趣的例子，山寨公司。

【小结】23个模式总算过完了，场景设计的很用心，老少皆宜。

【2010年04月18日 日 20:15】

## 11. 第八里程

【2010年04月19日 — 20:20】

【C30】PK阿，刺激，类比再对比，是深入了解事物的最佳实践。

【C30.1】第一回合，PK的不错，有点内裤外穿的势头。

【C30.2】第二回合，不太地道，应该工厂PK抽象工厂，工厂欺负Builder。

【C30.x】这么P如何：超人工厂PK抽象工厂，抽象工厂PK建造者于汽车。

【C31.1】场景合适，级别相当，恰到好处。

【C31.2】丑小鸭的例子很好很强大，目前为止最赞的一个，是如何想到的呢。

【C32.1】命令模式在这场PK中状态不好，晕乎乎的，结论很清楚。

【C32.2】压缩对策略有力，对命令不利，容易漂移。

【C32.3】DNS这段PK也到位，买一赠一，看PK，送DNS原理。

【2010年04月19日 — 21:40】

## 12. 第九里程

【2010年04月20日 二 09:10】

【C33】各种职业互P，法师对战士。

【C33.1】略读，最佳实践总结了，尤其是抓到耗子就是好猫。

【C33.2】看点是类图和最佳实践，可以扩展下场景应用。

【C33.3】五大高手，阵容强大，从头读到尾，必有收获。

【P474】全书中，代码34-9(好像)是第一个关键字加粗的。

【C34.x】连横合纵，天下一统。要是作为上机考试题如何。

【C35.1】貌似这个例子背景强大，眼睛一亮。过程很Mini。

【C35.x】过程很Mini，点到为止。

【C36.x】行，过，有所获。

【C37.x】规格模式，AND，OR，NOT的结构是看点。

【BTW】SSH曾害我找ssh资料费了不少搜商，简历上也见到最多。

【C38.x】MVC火的不得了，略读。

【小结】一不小心，看完了，回头写总评。

【2010年04月20日 二 11:20】

## 13. 终于拿下

好久没这么有压力，有计划，有步骤的读书了。限时读书是个好套路。

免费总是有魔力的。看着柜里间歇冬眠的书，真是非借不能读也。

回顾一下读书过程，9个里程，累计23小时，感觉3个小时一里程效果很好。

书评将要发出，心如跳兔。不是见仁见智，就是贱人贱智，O( \_ )O。

有人的地方就有江湖，有评的地方就有争辩，评论员不是什么好差事( o )。

## 《设计模式之禅》

最后，支持真原创。大家积极修书立传，授业解惑。

-----  
更详细评论参阅：

<http://www.trydofor.com/a9w3-auhome/trydofor/article/2010/0420133401/body.htm>

80、有趣，还在读

81、讲解言简意赅，比四人帮的书容易懂。但不知怎么，老是看了就忘，需要反复看看

82、书评是不是花钱找人写的啊，很多例子还不如网上博客文章写的清楚，比如builder那章，那讲的是什么跟什么啊，越看越一头雾水。

83、过于啰嗦

84、工具书。

85、例子恰当，更好的一点在于合理通过范例论述多种模式结合的应用场景。属于国人技术图书之佳作。

86、很有煽动性！

你平时工作会用到设计模式吗？

87、用生活化的语言来讲解设计模式是本书的最大特点。优点是通俗易懂，读起来很轻松。缺点：1. 不够严谨。比如错别字，比如不当的例子，比如代码中误用的英文单词，比如几处提到“线程安全”的地方存在错误。2. 针对Kindle版电子书，所有代码中需要标为“黑体”部分没有用黑体标出来。

88、走马观花，看完最大的收获就是要及早给自己想出来的做法起个名字，这样就算别人想到了也算你的了（逃

89、深入浅出

90、语言通俗易懂，例子丰富，拉近了设计模式的距离

91、需要经历过，才能更懂得

92、很不错的一本书 通俗易懂

93、其实现在都是组件式开发，面向对象的很多方法并不适用，但我还是决定看第二遍，算是打算把代码写得优秀一点，发现了不少错误，还有不少讲得很模糊的地方

94、想要不滥用设计模式，需要掌握这些原则：「单一职责」「里氏替换」「依赖倒置」「接口隔离」「迪米特法则」「开闭原则」。

95、看了几条书评都是说Head First Design Patterns比「设计模式之禅」好，本想想买来《设禅》的（目前已经在图书馆把它前面原则设计等前部看完了），搞得我现在都想去看看前者是个怎么好法咯。

96、通俗易懂，适合入门。

97、阅读起来比较轻松

98、很荣幸能成为《设计模式之禅》的试读员，以前自己也看过一些设计模式的书籍，都是在以导弹火箭作为题材阐述，有两点，第一、理解非常难，第二，都是抄的，很是郁闷

抱着尝试的心态去读了这本书，因为我自己读书本身比较慢，所以2周的时间只看到设计模式中的装饰模式那里。非常想说的一句话是，这本书不是抄的，比较容易理解，

这本书的结构我很喜欢，从写代码和设计代码的原则入手，直至设计模式到设计模式的混合搭配还有最后的大量设计模式的应用案例，很感谢作者，这本书对我的帮助很大

99、从程序员到初涉软件架构设计，对于我，“设计模式”仿佛从“Hello World”开始就是一个谜，面对完整的项目，复杂的需求，如何设计模型，这的确是一个问题。完美的设计出自对模式的理解和丰富的项目经验，不得不承认，这需要我们花费很多时间和精力于此，并付诸于实践中去。

但是最近笔者阅读了机械工业出版社的《设计模式之禅》一书，对设计模式的“禅理”娓娓道来，并且有很多精彩的例子做辅，语言平淡直白，通俗易懂，就像是在与深谙其道的朋友聊天。

纵览全书，本书分为四部分：

设计原则，针对六个设计原则进行阐述，其中不乏经验之谈；

设计模式演绎，分别描述了23中设计模式，并且每一种都有例程在里边，每种模式应用的场景被形象的描述出来，比如第十六章的责任链模式，形象的用我国古代“三从四德”来举例，描述责任链

## 《设计模式之禅》

模式是怎样应用的，等；

设计模式比较，分别就创建类模式，结构类模式和行为类模式等进行PK，

设计模式混编，列举几例模式嫁接应用，和MVC框架。

总之，该书语言平实，实例精准，内容翔实，涵盖面广，是软件设计模式中不可多得的一本书。简单的几行文字不足以概括其精妙，还需细细品味。

100、设计模式之禅是本好书。国人也能写出好书地

101、在kindle上买的，初学设计模式，算是了解了一些

- 1、这种形式写出来，很多例子总是给人一种牵强附会，生搬硬套的感觉。尽管编程领域里面的很多概念可能源自于现实生活，但大多数概念又是没法完全对应到现实生活的。没有《敏捷软件开发》讲的实在。很多例子里用了接口，用了抽象类，但是给人感觉，除了徒增复杂度之外，又看不到其它的好处。有些模式，之前的了解跟作者的表述还是有不小的差别，比如Builder模式，桥接模式这些。DIP原则的High level感觉应该指的是class集成层次的base class。有些场合下也称一个class的调用者为High level。书里面有些地方好像混淆了这两种不同的high level的概念，在一些不甚恰当的地方又去套DIP。解释器模式，确实比较难以说清楚。好的地方就是，讲的比较齐全，后面还补充了一些不包含在23中设计模式之内的模式。
- 2、前面6个原则+4个模式看完了，很爽，确实像看小说，不管了，记不住就记不住，至少能让我看进行，不像看哪些所谓专家推荐的，SHIT，就是催眠用的。等看完了再详细写书评。
- 3、这书我是先在网上找到电子版看的。虽然不全，但是足以认定本书乃“十世修来的”好书。我认为好书需要三个条件：一作者有真才实学；二作者善于讲解和表达；三作者态度认真，力求打造经典。而这本书就具备全部条件。于是我买了实体书，还包了书皮儿...至于书的内容嘛，讲设计模式呗。这东西的重要性需要强调吗？如果你觉得需要那你别犹豫了赶快找本书来恶补吧，这本就不错。推荐大家都来看看，绝对不亏的。
- 4、1. 综合评论【一句话总结】值得一读。比大话系列严谨，比GOF圣经易懂。69块钱，24小时，划算。【各部分感受】第一部分，六大原则，及其受用，适用于程序开发也适用于做人做事。书中有大量生动活泼的故事，有些十分贴切，想必作者费了不少脑汁。第二部分，对GOF的模式以有趣的方式庖丁解牛了一番，有些很独到。初学者上手快，已悉者温故知新。有趣味，有过程，有血肉。第三部分，PK很有特色，巩固知识，加深印象，消食通便。不打不相识，越打越亲近。条条大道通罗马，风景各不同。第四部分，合作共赢，综合应用，凝聚开发者的智慧。重点看需求上下文和程序架构，模式名字已不重要。附录的23种设计模式类图，是杀人越货之必备阿。【阅读建议】每节的【最佳实践】都应当理智阅读，原则也，实践也。演示代码只是为了说明问题，在实际项目中采用，要斟酌。建议阅读顺序，四一二三四一。阅读目标，忘记模式吧，融入场景，心中无刀。2. 事件考古【2010年03月12日 五】看分享，《设计模式之禅》试评员招募【2010年03月13日 六】凑热闹，“申请，看看什么是禅”【2010年03月19日 五】出结果，直到莹美女分享才知道。【2010年03月22日 一】发邮件，“哲思-设计模式之禅-试评员-登记”【2010年03月23日 二】收邮件，恭喜您成为华章公司《设计模式之禅》试评员。【2010年04月02日 五】第一篇，哲思杨某侠书评出炉。【2010年04月08日 四】怕误事，邮件确认：样书已于3月25日寄出，平邮。【2010年04月15日 四】样书到，雯美女亲临。3. 试评计划设计模式之禅/秦小波/机械工业出版社/ISBN978-7-111-29544-0/545页/69元试评员约定要在样书到手的两周内出书评，500字以上，发布于网上。作者学机械的，9年技术，儿子三岁。豆腐学化工的，7年技术，儿子175天。前辈啊，O( \_ )O哈哈~，这书有点啃头，有计划有步骤的耐心嚼之。尽信书不如无书，豆腐读书，绝对是批判的继承，批判是我思考，不直接反映书的品质。和买东西一样，挑刺越多，越易成交，满口好好的，可能路过。白天上班，晚上亲子。整块的时间不多，遂分而治之，评一次为一里程，路标如下。【C1.1】即第1.1节。(C)hapter,为章节标记。【P123】即第123页。(P)age,为页数标记。【BTW】随便说一下。(by the way)【小结】小小的总结一下。4. 第一里程【2010年04月15日 四 18:30】【书名】禅者，心也。有点玄。要是蝉就好了，知了也。【封皮】诗经·小雅·鹤鸣“它山之石，可以攻玉。”【作者】交行阿，信用卡用他家的，网银有待改进。【赞誉】很多人很高评价，还有阿福，有点看头。【前言】挺实在。致谢那段有意思。23种模式都用过，是否存在过度设计。【C1.1】info隐喻是属性，不应实现Biz行为接口，见仁见智吧。【C1.2】IPhone的例子很好，‘注意’‘学究’，提示背景和视角。【C1.3】我单纯，所以我快乐。【C1.4】很现实的问题，多快好省，服从领导。【C2.1】继承必须拥有父类的所有属性和方法。private的呢，从哪个角度讲。【C2.2】OO5大原则之里氏替换，09图灵奖女得主。例子很好，尤其CS那个。【2010年04月15日 四 21:00】5. 第二里程【2010年04月16日 五 18:08】【P24】表面类型和实际类型，头一次听说，为何不用大众叫法。【C3.x】DIP，真的是讲了不少。【C4.1】实例接口，这种讲法，长见识。【C4.2】星探找美女，不如改成相亲，更有杀伤力。【C4.3】IBookSearcher例子不错。【C5.1】你知道的太多了，所以越单纯越好。【C5.2】commond通假command? 4层含义讲的很透彻。【BTW】不太习惯，下划线开始的变量。



## 《设计模式之禅》

【2010年04月16日 五 21:30】6. 第三里程【2010年04月17日 六 12:11】【C6.x】开闭原则，拥抱变化，第一思考的原则。【小结】六大原则很是受用，程序员进化之必备。例子多样，很大众化。建议增设第七原则，就是奥康姆剃刀----“如无必要，勿增实体”。【P59】say()还是不要static了吧。【P60】构造函数private只是确保了非本类不能new，而不是仅产生一个实例。【P60】“类中其他方法，尽量是static”，public static 违反LoD，隐患多。【C7.3】对单例讲的很全面。除了clone外，反序列化也值得注意。【C7.4】场景假设的好，还普及历史知识，但代码有点不妥。【P62】countNumOfEmperor，static太糟糕，钰有时会说他是镇，都不用多线程。【P63】直接 Emperor.say()试试看。【C7.5】单例会被JVM的GC么？何种情况，理论依据或证据呢？豆腐认为有Ref就不GC。【C8.1】大话女娲造人的故事比较有趣。【P67】”其中的’？‘表示的是，……“没看到代码里使用。言之何物？【P67】Class.forName(c.getName()).newInstance();为何不直接 c.newInstance();【P67】使用泛型T，为何要Human强制转换一下呢。【P77】Map<String,Product> prMap = new HashMap();建议HashMap parameterized【C8.x】GOF说的很好。此节的例子不太恰当，有点为了工厂而工厂。【C9.x】略读。还是女娲娘娘这块比较有趣，想必费了不少脑筋。【2010年04月17日 六 15:20】7. 第四里程【2010年04月17日 六 18:20】【C10.1】本小三，纯属虚构。【C10.2】final防覆盖，这个提示很到位。【C10.x】模板这块讲的很细腻。记得JIC系统，就问过袜子，如何限定子类行为。【C11.1】变化是永恒的，Builder+Templet的例子很有代表性，值得研习。【P108】显式调用Collection的clear()，不仅可避免意外惊喜，还可避免泄漏。【小结】模板和创建这两节非常好，一个场景贯穿，一气呵成。【C12.x】用游戏代练类比Proxy，用“审计”点出AOP，本节的看点。【P146】this.arrayList改成thing.arrayList，笔误。【C13.4.3】冤家是因为final赋值，浅clone可以，深clone曲线救国也是可以的。【C13.x】电子账单的场景有来路，带有实际业务的影子，比虚构的系列好很多。【2010年04月17日 六 21:33】8. 第五里程【2010年04月18日 日 08:00】【C14.1】两个“库存情况”。后者应该是“采购情况”。【P148】“折半采购”少了stock.increase(buyNumber);【C14.x】贴近生活，容易理解。【P170】命令模式通用类图，Client关联Receiver还是Invoker？【C15.5】原来伏笔在此揭开，算我读的很仔细。【C15.x】有别于GOF，从新的角度讲解了Command模式。【C16.x】责任链，全体妇女同志站起来……了。【2010年04月18日 日 09:30】9. 第六里程【2010年04月18日 日 13:00】【C17.x】成绩单大话的很热闹，JDK中例子很多。【C18.x】略读。策略枚举慎用。【C19.x】略读。RMI慎用。【C20.x】如书中所说，太普遍了。研究研究Collection框架有好处。【C21.x】组合模式这么讲有点晕。树这么整不合适。【C22.x】先看反面例子，再看注意事项，再研究“暴露狂”。【C23.1】letterInotoEnvelope() 通假 Into。【C23.x】略读。应该谈谈JDBC。【2010年04月18日 日 14:40】10. 第七里程【2010年04月18日 日 17:30】【P307】宽接口，窄接口，讲的很好。【C24.x】备忘录，有用的东西。注意不同场景下的策略略和细节。【P317】图25-5，这个类图怪怪的，method首字大写。【BTW】类图有些没标返回值，图25-6，25-5，25-4。【P328】双反派，双分派。【C25.x】以邻居访问为故事讲解的很好。【BTW】很多方法论，都有很贴切的例子，在生活中活灵活现。【P340】代码26-13，还是setLiftState(Context.closeingState);好。【BTW】closeing,opening这两个ing很smilence(笑而不语)。【C26.x】电梯的例子很好，明了贴切，值得细品。【C27.x】堆栈计算器，适合学习。【C28.1】对厂商的分析工具感兴趣。【P360】工厂是200多个并发，咋没进行线程安全控制呢。【P365】讲了。【P369】10万次，多按了2个零 10000000【C28.x】故事场景不错。对象在内存的大小可以通过成员变量估算的。【P378】眼中无黑体部分，心中有黑体部分。印丢了，呵呵。【C30.x】又是一个很有趣的例子，山寨公司。【小结】23个模式总算过完了，场景设计的很用心，老少皆宜。【2010年04月18日 日 20:15】11. 第八里程【2010年04月19日 一 20:20】【C30】PK阿，刺激，类比再对比，是深入了解事物的最佳实践。【C30.1】第一回合，PK的不错，有点内裤外穿的势头。【C30.2】第二回合，不太地道，应该工厂PK抽象工厂，工厂欺负Builder。【C30.x】这么P如何：超人工厂PK抽象工厂，抽象工厂PK建造者于汽车。【C31.1】场景合适，级别相当，恰到好处。【C31.2】丑小鸭的例子很好很强大，目前为止最赞的一个，是如何想到的呢。【C32.1】命令模式在这场PK中状态不好，晕乎乎的，结论很清楚。【C32.2】压缩对策略有力，对命令不利，容易漂移。【C32.3】DNS这段PK也到位，买一赠一，看PK，送DNS原理。【2010年04月19日 一 21:40】12. 第九里程【2010年04月20日 二 09:10】【C33】各种职业互P，法师对战士。【C33.1】略读，最佳实践总结了，尤其是抓到耗子就是好猫。【C33.2】看点是类图和最佳实践，可以扩展下场景应用。【C33.3】五大高手，阵容强大，从头读到尾，必有收获。【P474】全书中，代码34-9(好像是第一个关键字加粗的)。【C34.x】连横合纵，天下

## 《设计模式之禅》

一统。要是作为上机考试题如何。【C35.1】貌似这个例子背景强大，眼睛一亮。过程很Mini。【C35.x】过程很Mini，点到为止。【C36.x】行，过，有所获。【C37.x】规格模式，AND，OR，NOT的结构是看点。【BTW】SSH曾害我找ssh资料费了不少搜商，简历上也见到最多。【C38.x】MVC火的不得了，略读。【小结】一不小心，看完了，回头写总评。【2010年04月20日 二 11:20】13. 终于拿下好久没这么有压力，有计划，有步骤的读书了。限时读书是个好套路。免费总是有魔力的。看着柜里间歇冬眠的书，真是非借不能读也。回顾一下读书过程，9个里程，累计23小时，感觉3个小时一里程效果很好。书评将要发出，心如跳兔。不是见仁见智，就是贱人贱智，O( \_ )O。有人的地方就有江湖，有评的地方就有争辩，评论员不是什么好差事( o )。最后，支持真原创。大家积极修书立传，授业解惑。-----更详细评论参阅

: <http://www.trydofor.com/a9w3-auhome/trydofor/article/2010/0420133401/body.htm>

5、前一段时间终于领到了我期待已久的《设计模式之禅》一书，但是由于工作的原因，一直没有时间静下心来细细品味作者那些来自自己工作实践中的禅语。我把这本书放在我的床前，每当我临睡前，我都会翻翻此书。在此之前我也看过一些设计模式方面的书，但是看看就想睡觉。有些设计模式的书看的是云里雾里的，看完了都不知道人家说的是什么。《设计模式之禅》这本书更多的是从实践和应用的角来讲设计模式，作者根据自己的项目实践总结出一些使用设计模式的方法和应注意事项。再加上作者通俗易懂的语言描述，读起这本书来就像读武侠小说一样过瘾。翻看了一下本书的目录，我觉得本书不仅仅是对23种设计模式的解读，更多的是对这23中设计模式进行比较，比较各设计模式的优缺点，以便在实际应用中进行合理的使用。《设计模式之禅》确实是一本值得读的好书，我会好好的读这本书的。

6、本来应该早些写出来的，不过最近家里杂事太多，一直抽不出时间了，所以耽搁了，还请大家见谅。活动里面说提倡原创，吓得我都不敢看其他已经贴出来的书评，生怕自己的思路受到影响。好了，言归正传。最深印象：这本书绝对是有深厚编程功底的人才能写出来的。作者相当的务实，他不是简单摆弄理论，而是用自己的实际经验，用相当切合实际的例子对理论进行深入且深刻的阐述。这些例子不复杂，但确实能够比较好的说明问题。相信作者在例子的选择上是很下了一番功夫的。最值得欣赏的是作者是真正的做到了活学活用，而不是读死书。作者在对理论加以灵活运用的时候真正考虑了让人纠结的问题：度的问题。任何理论的运用都有一个度，不及或过之都是欠缺。而对度的把握则是难中之难，因为它没有可遵循的固定规律，只能依赖个人的经验和智慧综合考虑，这是任何机器都做不了的事情。相信大家能够从作者对度的把握上收益良多。阅读建议：1.如果有一定coding功底，建议看到作者举出的例子时，先自己加以重构/重新设计，然后再和作者改良后的代码进行比较。这样会收益多多。甚至可以在看到作者用汉字描述的例子时，就可以开始尝试用程序实现，进而作两次比较，一次代码抽象原型比较，一次是改善后的代码的比较。这样更锻炼人。2.尽信书则无书，适合别人的东西不一定适合你。所以用怀疑的态度去读书，尽可能尝试去推翻作者。这样你才能将作者写出来的东西变成你自己的东西，也就是做到活学活用。至于不足之处，呵呵，相对于这本书的优点而言，基本上都可以忽略，而且关于通俗和幽默本身也是一个度的问题，其间的斟酌也是相当费时，所以我也就不再鸡蛋里挑骨头了。

7、在一小段时间前，有点空闲时间，想学习一下设计模式。几年前就一直听“高人”们提到设计模式，当时也着实想认真研习一下，并认为不明白设计模式，始终是比较瘸腿。但找到的读物都令人费解，读起来很吃力，所以每次都中途而废。在网络上无意中发现了秦小波先生关于设计模式的一个PDF，感觉写的非常好，很对胃口，有眼前一亮的感觉。然后追根溯源就找到了这本《设计模式之禅》。这本书相对于其他讲模式的书，还是非常容易理解的，也比较容易记住。这本书让我重新拾回学习设计模式的信心，非常感谢作者。我感觉这本书可以这样读并从而学习设计模式：1、像读小说一样耐心读完，读的过程中要集中精力，尽量记住每一种模式的描述2、对照目录，认真回忆每种模式是怎么描述的，用于何种场景3、对于不能记清楚的模式，立刻找到对应章节快速浏览，从而加强记忆4、有几个模式会容易记混，所以要认真比较和加强记忆5、书本只是对模式进行了容易理解的描述，关键还在于理解每个模式的思想，然后使用于自己的设计中6、针对每个模式，都重新举个例子，并用代码实现，从而更加强化记忆和理解7、书中大部分的例子，都是先写不完全正确的例子，然后循序渐进，给出正确的例子。应该增加一个特殊的索引，直接指向正确的例子，便于读过的人可以快速定位。不过这本书很不错的一个做法是在最后提供了一个很大篇幅的关于各个模式的彩页，我觉得个人可以在这个彩页上直接标记出每个模式对应的页号，这样以后就方便快速索引了。8、对于手

## 《设计模式之禅》

头有源码并且设计复杂的系统或者软件，可以对照学习，对号入座，从而也可以加强理解。久而久之，以后再看到“高人”的代码就再也不迷惑了，可以一目十行的快速阅读代码了。特别是一些架构级别的代码。

8、实例丰富。全书545页，很少连着两面或三页只见文字不见代码，实际上基本每页都有代码。这些例子是作者九年的工作总结啊，其价值是不言而喻的！通俗易懂。先是提出问题，给出一个相对简单的解决方案，然后不断完善，循序渐进，层层深入，比如在讲工厂模式时，先由女娲造人的神话引出一个简单工厂的模式，然后根据变化过渡到多个工厂类的模式，再到抽象工厂，用贴近生活的例子和语言娓娓道出高深的模式，让人感觉自己就像在读白居易的诗一样，让一个不懂设计的人感受到了学习与设计的快乐。客观、全面。在由浅入深地给出相对完善的模式后，还从优缺两面进行总结，能让人有更深入的了解。同时，该书的讲解是很全面的。比如将唯一的单例扩展成固定数量的“单例”，再如原型模式中，讲解了浅拷贝与深拷贝后，还讲解了String的特殊性以及clone与final的冲突等，真是知无不言，言无不尽！来于现实，最后又回归现实。来书每章开始都从生活中常见事例引出某类经典问题，当我们怀着好奇之心一步步地看完解决方案后，发现自己已经又学习一种模式了，此时再去看那相对“抽象”的模式定义，已经感觉不到抽象了。最后，在每章的结束部分，又给出了该运用该模式的具体场景，将读者的思绪从定义又拉回了现实，给了我们巨大的思考与想象的空间。这也正是我们学习模式的原因啊！

9、说实在的，我根本不懂设计模式，本来看到好多人在力荐这本书，心里就痒的难受，但是，“书，不读，如废纸”，心里还是隐隐的告诉自己“再想想，买那么多书了，有几本彻底的看完过？！”恰巧有pdf格式的下，下来看看吧，再次说实在啊，看这本书的前言“如何阅读本书”，一个字“贼拉的震撼”，银生第二次看到发自内心的话语啊，而不是商人般的唯好必说、逢疵必让，第一次是看《21天学会C++》，作者前言说道“没有任何一个人能够在21天内掌握一门结构严谨的语言”，人家把道都给你指明了。第二次，就是本书，为什么本书会让我娇弱的小心眼儿为之震撼呢？我说过，我是一只小鸟，小phper，知道成长的艰辛，绞尽脑汁去参悟作者的意思、看花了眼睛去阅读手册，成之路真是辛苦啊！可能是受尽以往资料生硬语言的束缚，本书叙事方式和以往书籍截然相反，举例说明：大家都有几个要好的朋友吧，和朋友谈话时候什么感觉，和朋友谈心时什么感觉，和朋友扯淡是什么感觉？看这本书就是啥感觉，还是一个字“贼舒服”，换句话说，你和杨教授聊天，啥感觉？（你和杨教授关系老铁了，另当别论）；嘿嘿，上学的时候语文没学好，想到哪说到哪，倒是全部发自内心，谢谢：）

10、纸质书第16页下面刚改的代码，在17页仍然输出“父类被执行“而不是” Map转Collection被执行“，难道我理解有误？

11、“设计模式领域又一里程碑之作”，这个评价夸大其词了。买了这本书，还是有点后悔的。如果需要入门设计模式，还是看那本head first（例子java写的）更好一点。然后四人帮的那本经典，也是必读的。这本书名字起的很大气，但是内容并没有达到这个高度。

12、很荣幸能成为《设计模式之禅》的试读员，以前自己也看过一些设计模式的书籍，都是在以导弹火箭作为题材阐述，有两点，第一、理解非常难，第二，都是抄的，很是郁闷抱着尝试的心态去读了这本书，因为我自己读书本身比较慢，所以2周的时间只看到设计模式中的装饰模式那里。非常想说的一句话是，这本书不是抄的，比较容易理解，这本书的结构我很喜欢，从写代码和设计代码的原则入手，直至设计模式到设计模式的混合搭配还有最后的大量设计模式的应用案例，很感谢作者，这本书对我的帮助很大

13、这几年来，前后读过3本设计模式的书：1.《大话设计模式》2.《HeadFirst设计模式》3.《设计模式之禅》前两本书出版的时间比较早，也比较畅销，于是我都买了，总体而言，我能给《大话设计模式》打60分，因为它有的仅仅只是对设计模式的解读，而且是最基础部分的解读，试图用一些很能让读者容易懂的方式去解读，但在我看来，这方面其实做得并不好。缺乏比较核心的内容。我能给《HeadFirst设计模式》打75分，这本书在设计模式领域还是相当有地位的，因为它开创了讲解和学习设计模式的新方法，这本书的优点就在于通俗易懂，精准到位，不足之处在于没有能很好地把这23种设计模式进行融会贯通和综合比较，也没有太多关于如何才能更好地应用这些设计模式的讲解，属于入门价的，对提高没有太多帮助。《设计模式之禅》算是后起之秀了，这本书刚出版时在社区里引起了轰动，身边的同事也有几个人买了，我借来看了看，一下子被吸引住了，于是自己也买了1本，到现在为止，我已经看了2遍了，感觉自己设计模式的理解加深了很多。这本书几乎继承了前面两本

## 《设计模式之禅》

书的所有优点，而且避免了前面两本书的不足。强烈向想学习设计模式的朋友推荐这本《设计模式之禅》

14、在我的印象里，技术类书籍一向是相当枯燥的，至少我之前看的一直是这样。眼睛死盯着码起来的文字一个个地啃下去，遇到难理解的地方，自己看不明白，往往还得回头再精读一遍，就是神仙也没了兴趣。学习本应是一个快乐的过程，相信那些技术类书籍的作者也不愿意看到读者把自己的著作看做一个个纸疙瘩，那就真杯具了。《设计模式之禅》（以下简称《设禅》），很厚的一本书，跟我之前看的一本相当枯燥的《Web信息架构》略厚。当时一拿到手，心想：完了，这要看到什么时候！我这人看书本身比较慢，非得要把书里面的每一个字都要读透（这是个不好的习惯）。但是拿起《设禅》真正读起来的时候，彻底打消了我之前的疑虑，书中通过一个个实例给我们讲了一个个编程的故事，平和的语言，滑稽的比喻，使得整个阅读过程都相当流畅，一点没有之前《Web信息架构》那般拖泥带水的感觉……（惭愧，难为了那本很专业的书。）完全可以理解华章图书对这本书的重视，书的作者的确是花了脑筋的。作者的想象力和创造力也在书中有很好的体现，经常会在代码示例中加入一些很幽默的注释，例如“运行的时候开电梯门？你疯了！电梯不会给你开的”，“这是绝对合理的，只运行不停止还有谁敢坐这个电梯？！估计只有上帝了”等等。书中也经常把一个个编程对象比作成现实中的例子，犹如看小说一般行云流水。其实，每个软件在设计者的心中，都是一本小说，一个故事……能把深奥枯燥的理论知识变得风趣幽默通俗易懂，没有丰富的经验是不可能办到的，也体现了此书的难能可贵。

15、很言过其实的一本书。第一：作者说了，是用咱们的母语讲解设计模式的书，可是每次下定义的时候都先用英文下，然后再用母语重复一遍，估计是为了凑字数的。建议：如果能看懂这本书中的英文，建议直接看HeadFirst Design pattern原版，该书比本书至少要好三个档次。如果看不懂本书的英文，请看HeadFirst Design pattern的中文版，虽然有很多经典被翻译糟蹋了，但HeadFirst Design pattern翻译的还是不错的。书中错误百出，我昨天稍微翻了翻，就发现几处很明显的错误：例如第11页，“直接把sanMao.killEnemy(new Rifle())改为sanMao.killEnemy(new MachineGun()),”简直不知所云，原来每开一枪就要换一把枪。第217页，“你要一个研究生，你派了一个高中生给我，那算什么事”，第一个你应该是吧，要不你要一个研究生和你派了一个高中生给我有什么关系。还有一些明显的错别字。当然，这种情况的主要责任在编辑。建议以后编辑将主要精力放在提高书籍质量，避免低级错误身上。不要光顾吹牛，设计模式的里程碑，按照我的看法只能是四人帮的原著，虽然该书有些晦涩，而且翻译的很糟糕，但里程碑就是里程碑，不是随便挑块石头就可以当里程碑的。

16、转卓越网读者评论：是程序员都知道设计模式很重要，几乎所有还没有悟透设计模式的程序员都指导设计模式很难搞，学习的成本非常高，过来人都说要一边实践，一边理解，然后一边总结。关于设计模式的书非常多，有GOF的，也有其他解读版的，我个人在学习设计模式的过程中也接触过几本，比如GOF的《设计模式》，书是很经典，但是太难懂了；《Head First设计模式》也读过了，还不错；《大话设计模式》在书店里也翻了翻，能打个60分吧，远没有大家说的那么好；《设计模式之禅》这本书我读了快一半了，应该说的确弥补了这几本书的不足之处，浅显易读但是又不肤浅是这本书最大的特点，能从书中设计的案例看得出来作者的道行比较深，而且写这本书时很用心，实在是读者的福气。建议想学设计模式的朋友首选这本书，应该不会让你失望。

17、原来借阅过阎宏的《Java与模式》，当初看时，刚入门，对模式设计的兴趣不大，而且对传统文化了解浅薄，感觉读不懂，就放下了。所以，拿到这本书，看到名字的时候，就是怕自己再次看不懂。但在认真的看了几天之后，就明显感觉这本书讲述的比较朴实，比较结合现实。工作了这么多年，或多或少的用到了些模式，但对它们之间区别和运用往往不够重视，所以开发中往往犹如芒刺在背，本书开头用了六章来讲解最基本原则。本书定位读者应该是对模式有所了解和项目联合开发经验的，不过没有这些经验，也可以作为指导性书籍来看。我所在公司不大，项目基本都是单兵作战，往往是一个人同时扛多个项目，而每个项目的参与者顶多不过3-5人（包括页面设计和客服等），开发上往往只有一两个人，所以，接口用的不多，就是有也放着不用。但看了这本书以后，感觉受益良多，发现很多东西如果设计的好，可以避免重复工作的浪费。设计模式，不是说看过明白了就可以把书扔掉，此书也介绍了一下常见的组合模式，看起来，就不费劲。推荐给大家此书，时常看看。另外，特别感谢哲思社区和华章公司，给我这个机会做此书的评论员，书评有点晚了，非常抱歉！

18、读这本书的起因源于csdn学生大本营的一次活动《设计模式之禅》试读员招募，身为程序员兼之学生大本营的老师没有道理不踊跃参加了（参加时可没走任何后门），佛祖显灵，真的能有幸成

## 《设计模式之禅》

为了试读员。从得知寄出样书，居然用了半个多月才拿到手，长了些。却也将我急切等待读书的浮躁心情洗涤为一种平和的心情，此时读这本《设计模式之禅》未尝不是一件好事，正应了“禅”的心境了。

翻开书 赞誉映入眼帘，这么多还是直奔主题吧，我这人历来读书都是先看目录，读懂目录，书也就有机会读懂了。在参加活动时，在秦少波老师主页中，简单阅读了部分样章（我这人比较懒，看书还可以，让我在电脑上看书向来懒惰），作者将要阐述的知识分成了四大块。从面向对象的6大设计原则入手，逐步演绎了GOF23个设计模式的一招一式，到后来更是演练了一番海陆空联合作战（设计模式混编）。每个设计模式用一个小故事引出该模式的定义，随之展开、深入。结构分块清晰，正是我喜欢并常建议我的学生学习时采用的学习和思考方式，由原则规矩入手- 一招一式- 组合拳。

清楚了这本书的内容的组织结构，闭眼那些知识走马灯似在脑中过了一遍，设计模式就是这样啊，禅之何来。往后看下来，不对，不同。章节中深入浅出的故事让初学者对设计模式通俗易懂；几句话的定义又将知识点透彻清晰的总结了；而随之的应用、经验技巧正是唐三藏要求取的真经啊！！！快速翻过跳到“第三部分设计模式PK”，慢慢的读下来，PK相似、PK不同，不是调侃似的PK，不是理论似的PK，好好好，设计模式在心中越来越清晰的勾画出来。书读到这里还没完，作者居然在最后的增加几个“开发项目”。好生动的演练。通过前边知识的准备，潜移默化的深入再深入地探讨了几个主要模式。阳春白雪下里巴人。好佩服作者深厚的编程开发功力和文字处理能力，居然在一部书里层次清晰的实现了雅俗共赏。再次闭目冥想，设计模式如春雪入土，慢慢消溶；土中有雪，雪中有土；不着于痕迹——何时悟禅？初读，事多，时间紧，未及细读，很多未及文字，不通处见谅，欲知后事，再读，再再读有感时……

19、书是在再次读完 Head First Design Patterns 后读的，易于做横向比较，估计接下来会把《大话设计模式》也一并扫读了。我是看完后随手把书评发到微博上，整理到这里，就不再添字了。扫完「设计模式之禅」，读的是PDF版本，缺了几节。整体质量一般，最值得看就是对SOLID解说这一章。作者说读过 Head First Design Patterns，但不喜欢书中的西方幽默，但我觉得 Head First Design Patterns 这本书更好。Head First Design Patterns 较之「设计模式之禅」的优点：两者都在章节起始导入问题，但前者更深入地通过文字引导读者思路，并且伴随着教导读者些许测试驱动和重构的方法。此外，Head First Design Patterns 会比较类似模式之间的异同优劣。Head First Design Patterns 没有说尽23个设计模式，它有针对性地选择、编排设计模式讲述的先后，并在设计模式的讲述中插入对设计原则的讲解。相反，「设计模式之禅」只是简单介绍各个模式，代码篇幅太多，甚至只给出代码就让读者自个领悟去。引用「设计模式之禅」中的一句，看官自个评质量：“策略模式的好处就是：体现了高内聚低耦合的特性呀，缺点嘛，这个那个，我回去再查查”、“这个就不多说了，自己领悟去”（第7页）。注：我看的是PDF版，与实体书可能有些许不同。但大体内容和风格应该是一致的。总结：有闲情可以翻一翻，不值得入手。Head First Design Patterns 更适合用于了解设计模式。这篇简评很中肯：

<http://book.douban.com/review/3219343/>

20、先说点题外话，“感谢国家～”，感谢哲思，感谢华章给我这次机会～一开始以为是报名者有份，后来才知道是抽取五名幸运网友，俺从小从没中过奖，所以说这次被抽中试读真的吓到我了～没拿到书以前，以为是纯技术流，设计模式本身就是软件工程学精英和前辈们总结出来的智慧，再加上个“禅”字……呃，于是乎鄙人使劲使劲看了又看《Head First Design Patterns》和《Thinking in Design Patterns》以免露怯诚惶诚恐的拿到书，打开一看，我的第一反应是”shocked“，印象最深的就是那行”孙悟空……“之类的，再回头一看封皮，没错啊是这本书，我承认，初见这本书，哥被吓着了但是！仔细通读全书，深深被作者驾驭软件工程学知识的能力所折服，如果能够继续延续这种状态写下去的话，那么XXXX之禅系列绝对可以变成中国人自己的Head First～作者在每个点，都以有趣的故事开头，深入浅出的讲解软件工程学原理，including design principles and design patterns, which is extremely suitable for novices AND experts.我想，对于novices来说，这些故事可能就是领他们进门的钥匙，对于experts来说，这些有趣的比喻也会让他们会心一笑吧设计模式向来是和设计原则紧密联系的，作者首先就在书中介绍了软件工程的六大基本原则，对比head first系列把设计模式和设计原则分到两本书里的XX行为，我只有一个评价，那就是orz.记得当时上软件工程学课程的时候，老师就是先讲设计原则，再讲设计模式的，对于我这样的学生来说，这样一本书让我倍感亲切，也让我回想起充满笑声（our lecturer was a funny guy who always makes jokes in class）的软件工程课。当然，从纯学习的方面来说，这样的编排应该说比较合理。再往下就是经典的设计模式的讲解，可以说，作者对于设计模式的掌握和讲解的能力，应该达到了高三老师的水平（我的意思是，他可以让没基础的人在短期内学会，然

## 《设计模式之禅》

后走向“高考”嗯嗯)。这一点，从书后附送的彩图——设计模式总结即可看出，这张表是一个极好的cheat sheet，上面对于所有的设计模式的总结精辟而周到，UML画得也相当明了按说，说完了设计模式就该讲混合设计模式了吧，作者做了一件灰常灰常牛X的事情，那就是对于经典的设计模式们进行了分类的比较，这个部分其实是最吸引我的，因为很少有人对于设计模式进行系统的比较然后写到书里，因此老师们经典的做法都是告诉你这个设计模式用在什么情况下，然后你就像机器人一样碰到啥情况你就用哪个呗。本书的这个部分，可能就是所谓的“禅”的精华了吧，这个部分，让码工开始思考问题，让代码机器们开始对自己的东西有所思索，如果你得到这本书，一定一定要好好关注这一部分~最后一部分，设计模式混编，是作者传授经验的部分，就像僧人诵经之后的经验交流，作者依据专业经验，通过之前的比较，对于几个关键的混合设计模式进行了翔实的讲解，但是个人认为，如果不积累一定经验，可能看这部分不会有太多感觉吧总体来说，读本书绝对不会浪费时间，当你读完本书开始码代码的时候，一定会想起书中那些鲜活的故事，想起作者的讲解，相信你会对你所做的事情有更深入的理解。

21、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。（来自卓越）

22、对与设计模式的认识源于2年前的一本书《设计模式》，GoF（“四人帮”，指Gamma, Helm, Johnson & Vlissides, Addison-Wesley四人）的《设计模式》第一次将设计模式提升到理论高度，并将之规范化，提出了23种基本设计模式，自此，在可复用面向对象软件的发展过程中，新的大量的设计模式不断出现。但是对于一个初学者来说，这本书毕竟是比较专业化的视角来描述，难免有些苦涩难懂，自从前段时间在51cto读书频道，接触到“设计模式之禅”这本书后，使我的想法完全发生了改变，作者以亲切自然的风格阐述了设计模式的核心思想，潜移默化地提升可我们面向对象的架构和编程能力，带我们进入了“物我合一，见性成佛”的最高设计境界。全书以简单通俗易懂但又不肤浅的描写方法，让我们在静静的思考中，慢慢的会意中，把设计模式的思维无声地融入自己的思维中。就像作者在前言中所说：“会看的看门道，不会看的看个热闹”本书对无论有无编程经验的人来说都是一本书中的尚品。对于经验欠缺的编程人员，完全可以把它当成一本小书看。书中通过对通过六大设计原则的全新解读、23种设计模式的完美演绎、设计模式的pk、设计模式混编等，向我们展示了作者的想象力，创造力和对设计模式的深刻理解，通过具体的开发场景，以讲故事的方式向我们阐释了设计模式的灵魂思想。通过这两天的阅读，建议大家反复研读，必将使自己的设计思想得以升华，使我们再次学习设计模式之时，领悟到“禅”的境界，禅的思想。总之，这本书无论对初学者也好，资深编程人员也好，都是一本难得的好书，在细细玩味之后，在潜移默化之中，我们对设计模式的理解必将富有禅意！就和大家分享这么多，希望大家在读这本书时，能共同交流，共同提高，是我们无论对技术的学习还是对生活的领悟都有一个更高的境界。（我的邮箱daiyuxi110@sina.com，希望和大家分享！）

23、我是个刚刚入行半年的小鸟，只读完了前六章，因为答应了要在收到书的2周内写出书评，所以断章取义的写了些文字....ok，切入正题：本书前6章比较详细的介绍了6大设计原则，相对其他设计模式的书籍而言我觉得这种方式比较能让我这种小菜鸟入门；作者在每章首先抛出定义，然后是比较有趣实例和一些容易理解的代码，接着阐述为什么要用，优缺点和一些经验建议；但是总感觉有点意犹未尽，换言之我觉得如果能再添加一些点睛之笔那句更好啦，对于这点我觉得&lt;冒号课堂&gt;的文章写的就不错（不过T-T太深了，有很多地方我功力不够...），就个人而言我觉得在阐述优缺点和经验的地方如果能更深入一些或者说更细致一些就好了:)个人觉得如果想掌握一种新的事物，要了解他的定义和后才可能知道这种事物怎么用，而经过对这种新事物应用并总结优缺点后，才可能真正开始了解这种事物的“价值观”，才可能无招胜有招。其实之所以提及这个，我觉得设计模式只是一种经验总结，换句话说，即使你没看过，也可能自己不知不觉的用过了，问题关键就是怎么去读懂设计模式的核心思想，因为最近很忙所以就选择仔细的读前6章，我认为前六章才是我的重中之重！如果你问我作者是否将这种精炼的思想提取并融入全书中，我暂时还没法回答，当然也可能是我太菜，不识庐山真面目；等等！难倒这就是作者题目所提及的禅吗？恩，等读后面我再好好的体会吧.....嘿嘿，我仔细的看前言了，作为一本将自己多年工作经验总结，并通过写成书的方式来共享自己知识，单从这一点，哥们！额，不！前辈，小菜鸟挺感动的，希望以后国内能看到更多类似这

类书籍,而不是那些胡乱粘贴的垃圾书籍.呼,可以去睡觉了...

24、刚拿到书,看的不多,感觉该书与其他书的一个最大的区别就在于语言的通俗易懂,书中的例子也大多是平时开发工作中遇到的,感觉很亲切,作者将设计模式的禅潜移默化的融入在简单的语言和生动的实例中,不是说教,不会让你感觉枯燥,我觉得这起码是好书的一个标准,禅,其实就这么简单!原文地址:<http://www.china-pub.com/196412>

25、刚知道自己有机会拿到这本书时,真是喜出望外,倍感兴奋。因为以前下了n次的决心想学设计模式,四人帮的《设计模式》也从图书馆借了换,换了又借。一次都没看完过,并且都是翻了几页就放下了,感觉设计模式太可怕了。但他却又是程序员不得不学的东西,于是为之深感烦恼。有时候时常安慰自己说“现在还没到学习设计模式的时候呢,等做过几个项目再去看自然也就懂了。”说是这样说的,那只不过是安慰自己的话罢了,仍想早些了解他,并能很好的运用它。这时候我极想能找到一本设计模式入门的书籍,先对设计模式有一个初步的了解,让后再去看那所谓的经典,但一直没找到合适的。就在这愁人之际看到了这本书,一下子就被这本书的介绍吸引住了: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。这介绍正合我意,心想我终于找到了它。我终于可以参悟设计模式中的奥秘了。之后便是每天都在期待着这本书的来临,可是之后一个月都没拿到书,那可恨的邮局啊。还好华章的张艳又赶紧快递了一本给我。真是太感谢她了。虽然收到了书,但此时已步入毕业设计的繁忙时期,白天工作,晚上回来做毕业设计,这本书只有等熄灯后每天看上半个小时了。到今天已收到书恰好两周了,不过至今仍没看完,但我已了解本书的写作风格,下面说说我对这本书的感受吧:本书使用JAVA语言描述的小例子向大家展示了6大设计原则和23中设计模式。之后又做了扩展来了个设计模式大PK,更令人欣喜的是最后一部分讲解设计模式在实际中的运用,若真没有这一章,我还真对本书有点小失望呢。呵呵。本书的优点:1 纸张质量这本书的质量没得说,华丽精美的封面,书的纸张也相当的不错,绝对适合收藏。2 读者范围广这本书的读者范围可谓相当广,不管您有没有编程经验,这本书您都可以去读,就像作者说的,没有编程经验的您可以把它当本小说去读。但说是这么说,我相信有一定的编程基础的同学都能看懂些。3 通俗易懂,风趣幽默作者之所以敢说没有编程经验的同学可以把本书当小说读,完全在于作者那章章节节的小故事,非常风趣幽默,能完全吸引住读者跟着作者的思路走,再加上作者使用相当简单的程序去描述每个模式,非常通俗易懂,我相信只要有点编程经验的人都能看懂一些的。4 讲解细致相比200多页的“四人帮”经典,作者可谓大方之极,看来“惜墨如金”这四个字在作者身上没起作用哈。^\_^,作者结合实例细致讲解了每个设计模式,之后仍感觉不过瘾,又来了个设计模式大PK,通过PK能读者能更好的理解各个设计模式。PK完之后作者貌似体力仍十足有余啊,又从实际编程的角度给大家讲解设计模式的综合运用,真是一盘绝顶大餐啊。正和我口味,呵呵。本书的缺点:俗话说世上没有完美的东西,如果完美了,他肯定不是来自地球。呵呵!而每个人对美丑的欣赏角度也不一样,观点定不会统一,我认为本书的缺点是:1 看本书需要一定的UML基础,这点真是有点令人不悦,虽然我是学C\C++的,对JAVA的了解为0,但代码我基本上都能看懂,只可惜作者的类图有些实在让人摸不着头脑,有些类图画的是在让我搞不懂有些类的关系到底是什么,目前实在懒得去看UML了,我想作者能对此进行一下解释就更好了,我认为在这个地方,作者设计好小例子之后,可以专门拿出一小节根据类图讲述类之间的关系,并由此引申一下设计模式,我想会更好。如果所有的东西都让读者从代码中看的话,没有编程经验的确实也只能看个热闹了,只能留给有2年以上工作经验的人看了。2 我感觉作者的例子可以在深入一些,我感觉有些例子稍微有些牵强,能深入些会更好。总之,这本书是一本学习设计模式的很好书籍,如果您还觉得设计模式枯燥无味,难学之极,那请你来看本书吧,它能在你不知不觉中把你带入设计模式的领地,并不断浸润设计模式之精华。看完本书并做一些练习之后,我相信你不但不会再对设计模式有所恐怖,反而还可以能在自己的项目中运用自如的使用设计模式,使自己的代码设计更加灵活。

26、代码太多了,特别是BO代码,完全没必要,唯一的好处就是让人有看书一目十行的感觉。。另外,作者废话挺多,有时候对基础婆婆妈妈,比如讲组合模式的时候的还介绍什么是树叶,什么是根,有时候却对关键的内容一笔带过,比如在第一次讲到ssh的时候,连个简称说明都没有。。不过,书中的例子都还是不错的,最后两章关于不同模式的对比和混合写的也很好,相信作者还是很费心思的。

## 《设计模式之禅》

27、从程序员到初涉软件架构设计，对于我，“设计模式”仿佛从“Hello World”开始就是一个谜，面对完整的项目，复杂的需求，如何设计模型，这的确是一个问题。完美的设计出自对模式的理解和丰富的项目经验，不得不承认，这需要我们花费很多时间和精力于此，并付诸于实践中去。但是最近笔者阅读了机械工业出版社的《设计模式之禅》一书，对设计模式的“禅理”娓娓道来，并且有很多精彩的例子做辅，语言平淡直白，通俗易懂，就像是在与深谙其道的朋友聊天。纵览全书，本书分为四部分：设计原则，针对六个设计原则进行阐述，其中不乏经验之谈；设计模式演绎，分别描述了23中设计模式，并且每一种都有例程在里边，每种模式应用的场景被形象的描述出来，比如第十六章的责任链模式，形象的用我国古代“三从四德”来举例，描述责任链模式是怎样应用的，等；设计模式比较，分别就创建类模式，结构类模式和行为类模式等进行PK，设计模式混编，列举几例模式嫁接应用，和MVC框架。总之，该书语言平实，实例精准，内容翔实，涵盖面广，是软件设计模式中不可多得的一本书。简单的几行文字不足以概括其精妙，还需细细品味。

28、酒吧以高薪聘“男女公关”为名 骗了广州市快腾贸易有限公司近百人“不许动，统统举起手来！”3月8日下午2时30分，在深圳南山区粤海街道大冲村的飞马酒吧内，“内应”侦查员配合数十便衣警察，将酒吧内50多名涉案人员全部控制起来。至此，一个以高薪招聘“男女公关”为名的18人诈骗团伙全部落网。为“两万元月薪”交出500元血汗钱今年20岁的小林来自江西，在深圳找工作期间看到写有“内部直招，非中介，免押金”字样的广告，一下子就被其“男女公关月薪2万元”“保安一职不算小费也有2700元工资”的优厚条件吸引住了，于是到约定地点飞马酒吧面试。酒吧守门人直至小林报出联系人贺小姐手机后面4位号码“9662”的暗号后，才允许其进入酒吧包间接受“面试”。“面试”时，接待人黄某告知小林保安一职已招满，而经理助理也早有人选了。正当失望的小林准备离去时，黄某改口说：“现在紧缺的是高级服务员和公关。”当得知“公关”就是陪富婆聊聊天喝喝酒，一个月能轻松赚三四万元的职业时，小林毫不犹豫地按要求将身上仅有的500元交了出来，在酒吧里一心等消息。“入围者”还要再交4000元“继续包装”此时，酒吧里突然冲进来数十名便衣警察，把小林吓了一跳。随着一句“不许动，统统举起手来”的大喝，酒吧里两名“望风”人员、一名“咨客”，酒吧大厅内30余人及“包房”内“面试”的10多人迅速被控制，民警还搜出账本、传单、赃款等证据一批。警方在围捕过程中，酒吧里又先后涌进了15名应聘者。经过辨认、清点、勘查，18名涉嫌诈骗的疑犯无一漏网。经查，酒吧内存在一个特大的招工诈骗团伙。这个大团伙又分为两个相互独立的团伙，其中以王×勇、刘×明等人为首的团伙，以7000元的月租租得场地。他们将其中一半场地租给以方×萍、王某等人为首的另一个团伙共同“营业”。男性应聘者交纳300元到1500元不等的“包装费”后，会被带到某些夜总会“绕一圈”，然后告知“富婆”没看上，想要继续做“公关”，还需交4000元的“继续包装费”。而少数容貌姣好的女应聘者则成为“坐台”小姐，大部分女子所交的入职费都落入了诈骗团伙成员的腰包。据警方初步统计，该团伙总“营业额”在10万元以上，受骗者多达上百人。

29、例子很好,读着比较轻松,就是讲解的不是很深入,设计模式入门,讲的比较浅显.真要学习设计模式还是得有大量的代码经验比较好,能看懂例子,但是看使用场景没有多深的印象.还有java的一点基础运用,就当了解这门OOP语言了.有个别java知识点还是蛮难的,对于不会java的人而言...

30、简单、通俗、易懂，但又不肤浅，这是本书的最大特色。尤为值得一提的是，本书还有设计模式PK和混编设计模式两大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“四人帮”的那本书也不例外。（来自卓越）



## 章节试读

### 1、《设计模式之禅》的笔记-第1页

单一职责原则的英文名称是Single Responsibility Principle，简称是SRP

### 2、《设计模式之禅》的笔记-第275页

```
class HanfeiZi extends Oberverable, IHanFeiZi
```

本来书是比较容易读的，不过出现这种低级错误，让人有点不放心啊，好感度大减

### 3、《设计模式之禅》的笔记-多看阅读

单一职责原则提出了一个编写程序的标准，用“职责”或“变化原因”来衡量接口或类设计得是否优良，但是“职责”和“变化原因”都是不可度量的，因项目而异，因环境而异。

### 4、《设计模式之禅》的笔记-多看

在类中调用其他类时务必要使用父类或接口，如果不能使用父类或接口，则说明类的设计已经违背了LSP原则。

### 5、《设计模式之禅》的笔记-第8页

只要父类能出现的地方子类就可以出现，而且替换为子类也不会产生任何错误或异常

### 6、《设计模式之禅》的笔记-前言 Preface

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)。设计模式是什么？它是一套理论，由软件界的前辈们 (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。... .. 只要知道在Java中 (//) 代表注释就足够了

### 7、《设计模式之禅》的笔记-第32页

优化后的模型行不通吧.请问作者如何实现,AbstractSearcher::Show()?

其实整个用例的设计都不合适.非常不可理解,IPettyGirl应该包含Show()方法,同时goodLooking(),niceFigure()等这些具体的判定条件应该是私有函数.

但如果这样,这个例子明显还是不适于用来说明接口隔离.大家再想想有没有更好的案例.

### 8、《设计模式之禅》的笔记-多看

只要父类能出现的地方子类就可以出现，而且替换为子类也不会产生任何错误或异常，使用者可能根本就不需要知道是父类还是子类。但是，反过来就不行了，有子类出现的地方，父类未必就能适应。

# 《设计模式之禅》

## 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:[www.tushu111.com](http://www.tushu111.com)