

# 重构与模式



[重构与模式 下载链接1](#)

著者:科瑞夫福斯盖

出版者:人民邮电出版社

出版时间:2010-12

装帧:

isbn:9787115241177

《重构与模式》开创性地深入揭示了重构与模式这两种软件开发关键技术之间的联系,说明了通过重构实现模式改善既有的设计,往往优于在新的设计早期使用模式。《重构与模式》不仅展示了一种应用模式和重构的创新方法,而且有助于读者透过实战深入理解重构和模式。书中讲述了27种重构方式。

《重构与模式》适于面向对象软件开发人员阅读，也可作为高等学校计算机专业、软件工程专业师生的参考读物。

点击链接进入新版：重构与模式(修订版)

作者介绍：

目录: 第1章 本书的写作缘由 1 1.1 过度设计 1 1.2 模式万灵丹 2 1.3 设计不足 2 1.4 测试驱动开发和持续重构 3 1.5 重构与模式 5 1.6 演进式设计 6 第2章 重构 7 2.1 何谓重构 7 2.2 重构的动机 8 2.3 众目睽睽 9 2.4 可读性好的代码 10 2.5 保持清晰 11 2.6 循序渐进 11 2.7 设计欠账 12 2.8 演变出新的架构 13 2.9 复合重构与测试驱动的重构 13 2.10 复合重构的优点 15 2.11 重构工具 15 第3章 模式 17 3.1 何谓模式 17 3.2 模式痴迷 18 3.3 实现模式的方式不止一种 20 3.4 通过重构实现、趋向和去除模式 22 3.5 模式是否会使代码更加复杂 24 3.6 模式知识 25 3.7 使用模式的预先设计 26 第4章 代码坏味 28 4.1 重复代码(Duplicated Code) 30 4.2 过长函数(Long Method) 30 4.3 条件逻辑太复杂(Conditional Complexity) 31 4.4 基本类型偏执(Primitive Obsession) 32 4.5 不恰当的暴露(Indecent Exposure) 32 4.6 解决方案蔓延(Solution Sprawl) 33 4.7 异曲同工的类(Alternative Classes with Different Interfaces) 33 4.8 冗赘类(Lazy Class) 33 4.9 过大的类(Large Class) 33 4.10 分支语句(Switch Statement) 34 4.11 组合爆炸(Combinatorial Explosion) 34 4.12 怪异解决方案(Oddball Solution) 34 第5章 模式导向的重构目录 36 5.1 重构的格式 36 5.2 本目录中引用的项目 37 5.2.1 XML Builder 38 5.2.2 HTML Parser 38 5.2.3 贷款风险计算程序 39 5.3 起点 39 5.4 学习顺序 39 第6章 创建 41 6.1 用Creation Method替换构造函数 43 6.1.1 动机 43 6.1.2 做法 45 6.1.3 示例 45 6.1.4 变体 49 6.2 将创建知识搬移到Factory 51 6.2.1 动机 51 6.2.2 做法 54 6.2.3 示例 55 6.3 用Factory封装类 60 6.3.1 动机 60 6.3.2 做法 61 6.3.3 示例 62 6.3.4 变体 65 6.4 用Factory Method引入多态创建 67 6.4.1 动机 67 6.4.2 做法 68 6.4.3 示例 70 6.5 用Builder封装Composite 74 6.5.1 做法 76 6.5.2 示例 77 6.5.3 变体 87 6.6 内联Singleton 90 6.6.1 动机 90 6.6.2 做法 92 6.6.3 示例 93 第7章 简化 96 7.1 组合方法 97 7.1.1 动机 97 7.1.2 做法 99 7.1.3 示例 99 7.2 用Strategy替换条件逻辑 102 7.2.1 动机 102 7.2.2 做法 104 7.2.3 示例 105 7.3 将装饰功能搬移到Decorator 115 7.3.1 动机 115 7.3.2 做法 118 7.3.3 示例 119 7.4 用State替换状态改变条件语句 133 7.4.1 动机 133 7.4.2 做法 134 7.4.3 示例 135 7.5 用Composite替换隐含树 143 7.5.1 动机 143 7.5.2 做法 146 7.5.3 示例 147 7.6 用Command替换条件调度程序 155 7.6.1 动机 155 7.6.2 做法 157 7.6.3 示例 158 第8章 泛化 164 8.1 形成Template Method 165 8.1.1 动机 166 8.1.2 做法 167 8.1.3 示例 167 8.2 提取Composite 172 8.2.1 动机 172 8.2.2 做法 173 8.2.3 示例 174 8.3 用Composite替换一/多之分 180 8.3.1 动机 180 8.3.2 做法 182 8.3.3 示例 183 8.4 用Observer替换硬编码的通知 190 8.4.1 动机 190 8.4.2 做法 191 8.4.3 示例 192 8.5 通过Adapter统一接口 199 8.5.1 动机 199 8.5.2 做法 200 8.5.3 示例 201 8.6 提取Adapter 208 8.6.1 动机 208 8.6.2 做法 210 8.6.3 示例 210 8.6.4 变体 216 8.7 用Interpreter替换隐式语言 217 8.7.1 动机 217 8.7.2 做法 219 8.7.3 示例 220 第9章 保护 230 9.1 用类替换类型代码 231 9.1.1 动机 231 9.1.2 做法 233 9.1.3 示例 234 9.2 用Singleton限制实例化 240 9.2.1 动机 240 9.2.2 做法 241 9.2.3 示例 241 9.3 引入Null Object 244 9.3.1 动机 244 9.3.2 做法 246 9.3.3 示例 247 第10章 聚集操作 252 10.1 将聚集操作搬移到Collecting Parameter 253 10.1.1 动机 253 10.1.2 做法 254 10.1.3 示例 255 10.2 将聚集操作搬移到Visitor 259 10.2.1 动机 259 10.2.2 做法 263 10.2.3 示例 267 第11章 实用重构 274 11.1 链构造函数 275 11.1.1 动机 275 11.1.2 做法 276 11.1.3 示例 276 11.2 统一接口 278 11.2.1 动机 278 11.2.2 做法 279 11.2.3 示例 279 11.3 提取参数 280 11.3.1 动机 280 11.3.2 做法 280 11.3.3 示例 281 跋 282 参考文献 283 索引 286

• • • • • (收起)

## 标签

重构

设计模式

程序设计

软件工程

模式

编程

软件开发

面向对象

## 评论

结合《重构》一书阅读，更好理解设计模式在重构过程中的实现与应用。普通程序员与优秀程序员的差别不在于是否能够实现功能，而是在实现功能后能否继续对所写代码进行思考与改善。

---

重构，没看懂

---

经典，只有找原版了，翻译不给力。

## 重构的补充

好书。“重构的目的是设计模式”简直是一语中的。不过里面不少说明有些依赖Java。有些概念的描述过于书面、晦涩和啰嗦。

另：第100页有笔误。数组扩容应该是增加10，而不是增加10倍。

## [重构与模式 下载链接1](#)

## 书评

之前看过不少有关设计模式的书，《java与模式》，《Head First Design Patterns》，《重构：改善既有代码的设计》。

但是这本《重构与模式》，确实最实在的一本。《重构》从最基本的最常用最简单的重构方法开始介绍，实在是受益良多，对以前对设计模式的理解加深了很多，终于...

最近刚看完<重构与模式>这本书，这本书很适合我的一直以来的观点，大多数情况下，模式的应用是一个渐进演变的过程，坏味道也不是一开始就出现的，而在一开始就想到用这个模式，那个模式的，很可能导致设计过度。貌似重构与模式有一种天然的关系。重构是手段，模式是目的，同时...

这本书更多介绍了复合重构和测试驱动重构，如果对一般的重构手法和设计模式都不了解，读这本书会比较吃力。如果有过不少的项目经验，读这本书更合适。

重构与模式还有极限编程，我根本不会指望从书本中入门。

一个项目，几个项目之间总结出的经验，然后回过头来看看这些书，也许会有些共鸣，或者修补一些经验的遗漏，至于入门，领会了皮毛只会造成滥用，最近在看ajax设计模式，尤其有这种感觉，一种理论，从形成到滥用的...

借鉴《设计模式》与《重构既有代码》，对其中一些见解提出自己的看法，同时对《重构》中的大部分例子进行了详细的说明，是一本通过实际例子提升重构能力的好书。心得：加深了对《设计模式》的理解：一个特定的环境(上下文)，一系列类似的问题，还有一个解决办法的结构方式。重...

这本书中的内容不太好懂，即使你看了Martin的《重构》和四人帮的《设计模式》，最好还是把里面的代码敲一遍，并仔细的重构，体会整个过程。在重构的过程，会发现一些新的体会。关于设计模式，我推荐《设计模式之禅》和 Head First 的那本设计模式。1...

之前还没看过模式那本书，所以不知道那些模式。现在看完只是对重构有一部分的了解，因为里面的代码为Java，而我是用objective-c的，但代码都是通用的。只能在之后看完了重构和模式两本书以后再回头看，应该会有更深刻的了解。现在对项目的重构有了一点的帮助，很好。

又一经典被蛋疼的翻译给毁了，我越看越不淡定，实在忍不住在书上画满删除线，大爆shit，第六章开始讲正题就充斥大量没标点的长句，带语病，好不容易加个标点，还能标错。看到译者序里写：“在准备写这篇译者序的时候，我终觉得很难，因为译者序类似读后感，是要到出翻译时的特...

总的来说，这是一本好书。“重构的目的是设计模式”作为全书的核心，简直是一语中的。设计模式应该是一门深入面向对象思想骨髓的内功，而不是表露其外的外功。为设计而设计的思路并不适合大项目。逐渐重构到设计模式才是设计模式的正道。读此书应该有面向对象语言的扎实功底...

只是看了一半，因为其中每个例子都含有作者实践过的真实代码，非常适合我们跟着作者的思路去思考，运用设计模式进行重构，解决眼前存在的难题！

不要在设计之初就采用复杂的设计模式，这一点我非常认同作者。使用设计模式是为了应对变化。在设计阶段难以预见系统变化。推荐采用简单的设计模式，例如工厂方法和组合方法。这本书告知读者，复杂的设计模式（例如访问者）产生于不断重构。在重构过程中完善系统设计，应对未来...

---

这本书断断续续看过几次，感觉距离head first  
那本有一定的距离，无法让我很投入，或者看后有所思考，可能是我经验不足。

---

[重构与模式](#) [下载链接1](#)