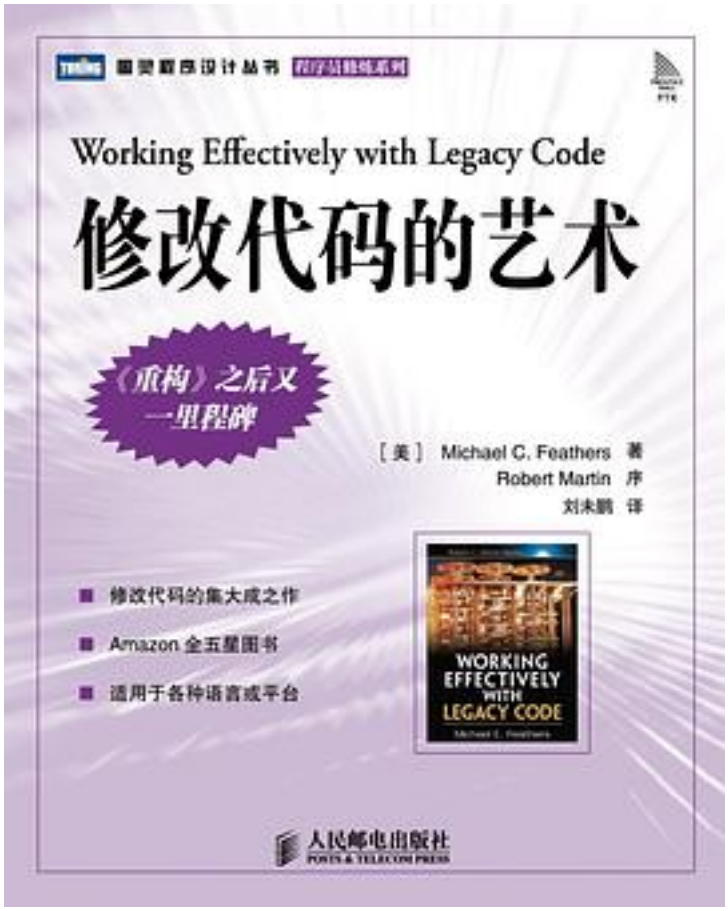
修改代码的艺术



修改代码的艺术_下载链接1_

著者:（美）Michael C. Feathers

出版者:机械工业出版社

出版时间:2014-6-15

装帧:平装

isbn:9787111466253

世界级计算机专家Michael C. Feathers的经典之作，软件开发大师Robert C. Martin作序倾情推荐，修改遗留代码的权威指南。深入剖析修改遗留代码的各种方法和策略，从理解遗留代码、为其编码测试、重构及增加特性等方面给出大量实用建议，是所有程序开发人员必读之作。

修改代码时，你觉得容易吗？当你修改代码时，能否几乎即时地获得反馈？你理解那些代码吗？如果对于这些问题的答案是否定的，那么你面对的就是遗留代码，它们正在浪费你开发工作的时间和金钱。

在本书中，作者为更有效地处理大规模、缺少测试的遗留代码提供了自始至终的策略。本书内容来自Michael创建的非常知名的Object Mentor公司的研习会，Michael使用那些技术来指导并帮助了成千上万位开发者、技术经理和测试人员，让他们的遗留系统处于可控状态。

本书主要内容：

理解修改软件的机制：添加特性、修正缺陷、改进设计、优化性能

把遗留代码放到测试用具之中

编写测试，防止引入新的问题

包含Java、C++、C和C#的示例，其中介绍的大多数技术适用于其他任何语言或平台

精确地确定要在哪些地方修改代码

处理非面向对象的遗留代码

处理看起来没有任何结构的应用程序

作者介绍:

Michael C. Feathers 世界级软件开发大师，就职于Object Mentor公司（这是一家世界领先的提供软件领域的指导、技能开发、知识传播和领导力服务的公司）。他是ACM和IEEE成员，也是CppUnit（从JUnit移植到C++上的单元测试框架）和FitCpp（FIT集成测试框架在C++上的实现）的最初作者，曾3次担任OOPSLA会议的CodeFest主席。目前他在世界范围内提供测试驱动开发、重构、面向对象设计、Java、C#、C++以及极限编程方面的培训和指导。

译者简介

侯伯薇
中荷人寿保险有限公司高级系统分析师，InfoQ中文站翻译团队主编，拥有十多年开发经验，目前致力于技术与业务的融合，让开发出来的程序能够真正提高业务人员的工作效率。热衷于通过翻译和演讲的方式与广大程序员分享和交流，曾翻译过多本技术书籍和几百篇技术短文，并在Scrumgathering、QClub、敏捷之旅等活动上做过技术演讲

目录: 目 录

修改代码的艺术_下载链接1_

标签

重构

软件工程

程序设计

计算机

编程

代码重构

软件开发

计算机科学

评论

1. 原书绝对是一本经典 2. 候伯薇翻译的这版简直太烂了，成吨的烂翻译和文字错误 3. 想看看刘未鹏翻译的版本。

实践性很浓的一本书，看过之后更有助于写出易于测试的系统

经典之作，改bug的境界

这里再重复一遍重构的定义——在保持代码行为的基础上，提升代码的质量。重构专注于第二步，即如何提升代码的质量，而修改代码的艺术专注于第一步，即如何保持代码的行为。
提升代码质量并不困难，但保持代码行为就难多了，尤其是对没有测试的遗留代码（Legacy
Code）而言——你需要首先引入测试，但遗留代码往往可测试性（Testability）很差，这时你就需要把代码变的可测试。修改代码的艺术包含大量的实用建议，用来把代码变的可测试（Testable），从而使重构变为可能，使提高代码质量变为可能。

稍微有些繁琐，可以需要使用的时候再细翻。拆分超大的类那篇，需要的时候，发现并没有讲细节，还是自己想了一些安全透明的方法来处理工作中的一些超大类的拆分（3300+行）

深入，透彻，具体

很OO，很Java（虽然也有适用于C这种过程式语言以及C++这种常用语言的建议）
不全是教条，值得参考。

修改代码的艺术_下载链接1_

书评

作为一个程序员，获取知识是让我不断前进的动力，而读书是我获取知识的一条重要途径。在这个“经典”、“必读”过剩的年代里，大多数的书都仅仅扮演着传播知识的角色，真正改变自己对某些问题看法的书其实少之有少。限于读书时的眼界和能力，在我列表中，让我拍案惊奇的书只有...

当软件系统的规模随着时间不断增长时，我们怎么构建和维护它？面对别人写好的大量的代码基，如何进行后续的可持续开发？TDD，单元测试，重构，设计模式这些看上去很美的技术，是如何应用的?
毫无疑问，这本书里不可能提供上诉问题的所有答案，但是它至...

这本书看的时间非常长, 断断续续有3个星期了吧, 不错的书, 至少对我来说是这样,
因为我现在就碰到了书中列出的种种问题:对已有的没有完善的单元测试的核心系统进行重构.为了保证少出乱子, 不出乱子, 我必须小心的对超大类,
巨型方法采用各种重构手段进行修改, 没有单元测试作保...

《修改代码的艺术》看完了
这本书很薄，但是看起来还是很吃力，里面介绍了很多重构的知识，而且有很多c++的内容，有的地方也是似懂非懂的，如果了解设计模式和重构，就会轻松很多，可能艺术这东西，本身就不容易懂吧。里面对单元测试的依赖性很强，其实还是一本不错的书，你完...

买这本书的原因一是这本书确实是一本关于修改老代码的经典，二来翻译者是中国地区 InfoQ 的主编。 但是入手看了大概到100多页之后实在是忍不住要上来吐槽一下。
首先是翻译的通畅性，应该说是比较烂的水准~~只能说是将将达到合格的水准，这个可能是个人的偏见。但是...~~

14h:05 in 6
days。我的“重构三部曲”之三，（另外两本是《重构》，《从重构到模式》，这三本书让我对代码的理解有重生之感。大部分书都是教你怎么从0开始写好代码，但是现实是经常从接手已有的项目开始，所以这三本就很有价值。）这本书压箱底8，9年了，前些年有次囫囵吞枣看...

Java重构的必读书，非常实用，但有的时候我想，Java代码的重构如此复杂，是否说明面向对象的设计思路在很多场合并不适用呢？比如很多服务端的逻辑本身是典型的函数转换，如果使用FP范型开发会简单的多。
推而广之，如果一种技术在大多数程序员手里都越用越复杂，以至于需要专家...

很好的实战经验，快来取道。在最近的开发项目中经常想起本书讲解的一些技术，受益匪浅。虽然我并不是 working on legacy code
，但是项目代码从无到有到完善也是经历几个阶段的，在不断演化，不断修正。另一方面，一边写单元测试，也参考了本书。 以前以为测试只是为了保...

一两个月前看到了这本书，那时候正对编写高质量的代码很感兴趣，于是借来读。这一个月断断续续的读完，实际上读书的时间仅有10天左右的业余时间。读的很浅，但也有小小的收获。
这本书讲解如何在不漂亮的旧代码下写漂亮的新代码，依照先有测试后有功能的思想，作者全书都围绕...

如果你想重构,重要的前提就是有强力的测试.哪怕你有自动化重构工具在手.
如果你想对既有代码进行测试,你就必须先重构,因为代码根本就没有办法在测试工具中实例化. ……
新写的代码大多是可以先进行测试,然后再挂接到原有代码中.而对付遗留的代码,我们则需要一点点地把代码抠出...

如果你想重构,重要的前提就是有强力的测试.哪怕你有自动化重构工具在手.
如果你想对既有代码进行测试,你就必须先重构,因为代码根本就没有办法在测试工具中实例化. ……
新写的代码大多是可以先进行测试,然后再挂接到原有代码中.而对付遗留的代码,我们则需要一点点地把代码抠出...

我发现很多网页里卓越的报价常常比当当的高，可是实际情况是点击链接后卓越比当当低！ 比如这本书实际报价：卓越是46.5，当当是46.6
不知道是不是最近卓越大范围调整了价格？！
顺便说一下，在csdn读书频道上也有类似情况。

修改代码的艺术_下载链接1_