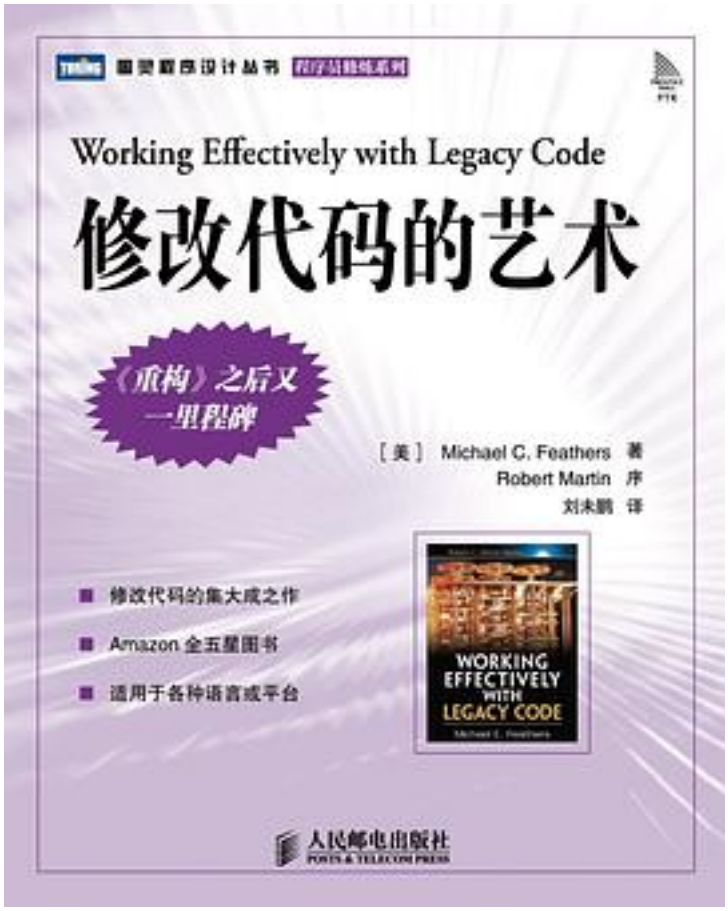
修改代码的艺术



修改代码的艺术_下载链接1_

著者:Michael Feathers

出版者:人民邮电出版社

出版时间:2007-09-25

装帧:平装

isbn:9787115163622

我们都知道，即使是最训练有素的开发团队，也不能保证始终编写出清晰高效的代码。如果不积极地修改、挽救，随着时间流逝，所有软件都会不可避免地渐渐变得复杂、难以理解，最终腐化、变质。因此，理解并修改已经编写好的代码，是每一位程序员每天都要面对的工作，也是开发程序新特性的基础。然而，与开发新代码相比，修改代码更加令人生畏，而且长期以来缺乏文献和资料可供参考。

本书是继《重构》和《重构与模式》之后探讨修改代码技术的又一里程碑式的著作，而且从涵盖面和深度上都超过了前两部经典。书中不仅讲述面向对象语言（Java、C#和C++）代码，也有专章讨论C这样的过程式语言。作者将理解、测试和修改代码的原理、技术和最新工具（自动化重构工具、单元测试框架、仿对象、集成测试框架等），与解依赖技术和大量开发和设计优秀代码的原则、最佳实践

相结合，许多内容非常深入，而且常常发前人所未发。书中处处体现出作者独到的洞察力，以及多年开发和指导软件项目所积累的丰富经验和深厚功力。通过这部集大成之作，你不仅能掌握最顶尖的修改代码技术，还可以大大提高对代码和软件开发的领悟力。

作者介绍:

Michael Feathers
世界级面向对象技术专家，以丰富的软件项目开发经验著称。目前在世界顶尖的软件咨询公司Object Mentor从事敏捷方法/极限编程、测试驱动开发、重构、面向对象设计、Java、C#和C++等方面的培训和项目指导。他是著名测试框架CppUnit和FitCpp的开发者，已经主持了三次面向对象界盛会OOPSLA上的CodeFest比赛。

刘未鹏，热爱编程技术，长期关注C++，现在南京大学计算机系攻读硕士学位，译有《Imperfect C++中文版》、《Exceptional C++
Style中文版》（人民邮电出版社出版）。个人blog：http://blog.csdn.net/pongba。

目录:

修改代码的艺术_下载链接1_

标签

重构

编程

代码

软件开发

计算机

软件工程

程序设计

programming

评论

基本上每周都要在几个长达2000行以上的函数里面漫游，学会控制情绪是一件很重要的事。本书的第24章”当你绝望时”只有短短的两页，可见作者也写不下去了，只好草草地以“混口饭吃”这样的理

由来安慰阅读此书的码农。

=。= 真的是鹏鹏翻的?

原来我写的全都是Legacy Code...>

以后干活顺便把测试的活也揽下来算了。另外，看到以前乱七八糟的命名，因为我太蠢而看不懂的逻辑，还有在我后面猛塞重复代码的，都很想把人揪出来揍一顿。

: TP311.52/5813

不推荐。

从测试角度出发的重构

尽管吹捧的很厉害，但是和《重构与模式》一样，充其量只是一本案例集，无法达到《重构》的高度。

先求一本实体啊，被china-pub晃点了!
作者靠谱、译者靠谱、内容靠谱，各种硬货都足够消化一段时间。大量内容和编写、修改、组织测试用例相关。虽然重构和代码重写说起来简单：减少（坏的）重复、减少耦合、增加内聚，但真正写出来每一点展开都是一本砖头书。

很多修改的技巧，更像一本手册，经常查阅....

Feathers出品

Day 47 和《单元测试艺术》的主题其实很接近，非常实操的一本书 #百日早起学习挑战

#就那样。

值得一读

很多耳目一新的测试/修改方式

太死板了。

最近的修改以前的项目了。赶紧把刘老大翻的这本书拿来瞧瞧，以免陷入泥沼之中。 God Bless Me!Don't get me fucked.

远远不及 Refactoring 经典

编写可添加单元测试的代码，让代码有更少的依赖，可能由于各种原因没法实践每个功能都有单元测试，但往这个方向去了，会让工程有更少的依赖，可扩展变强了。

分享了很多做法，可以说是干货

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_