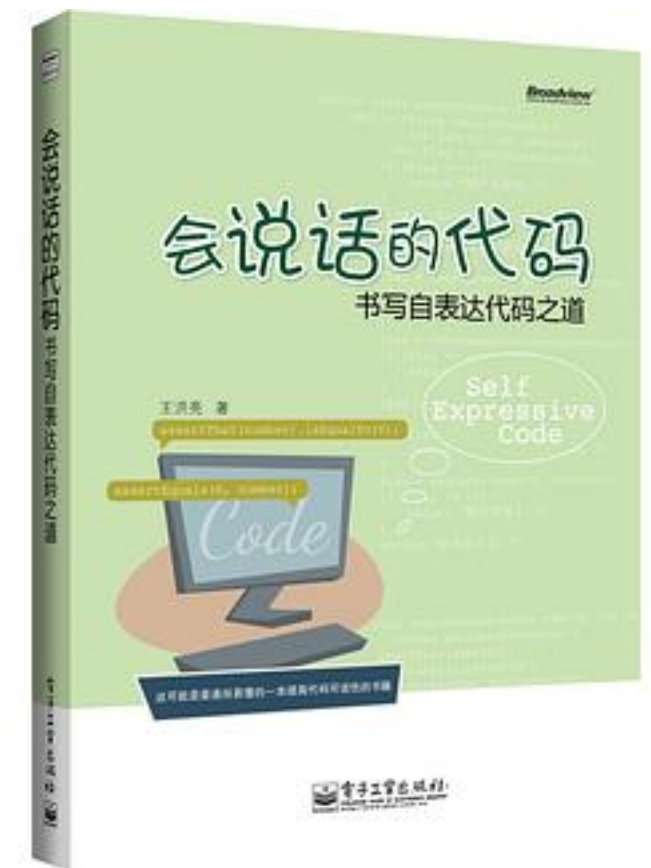


会说话的代码



[会说话的代码_下载链接1](#)

著者:王洪亮

出版者:电子工业出版社

出版时间:2013-8

装帧:平装

isbn:9787121208768

写得不好的代码有各种各样的问题，会给读者带来不好的阅读体验，并且如果代码写得不够好，文档和代码不一致，注释和代码不一致，那么对维护人员来说，理解代码和进入项目组都是有困难的。如果代码写得不够好，就需要扩展功能，修改Bug时所需要花费的时间也较长。

自表达代码是一种采用接近于自然语言的方式书写代码的主张。

代码可读性、可扩展性、可测试性是关联代码质量的重要参考指标。本书从各个角度来分析产生劣质代码的原因，并从代码的命名、注释、风格、结构、架构等方面着手，对提高设计能力、熟悉开发环境、了解编程语言、提高英语能力等诸多方面提出了具体的改善建议。

作者介绍:

目录: 第1章 劣质代码带来的劣质体验 1

1.1 代码的可读性问题 1

1.1.1 命名类问题 2

1.1.2 注释类问题 6

1.1.3 风格类问题 8

1.1.4 结构类问题 11

1.1.5 架构类问题 18

1.2 代码的可测试性问题 22

1.3 代码的可维护性问题 22

1.3.1 需求变更难以应对 23

1.3.2 纠缠不清的Bug 23

第2章 劣质代码是怎么产生的 25

2.1 理论知识匮乏 25

2.2 对编程语言不熟悉 29

2.3 对开发环境不熟悉 31

2.4 对设计方法不了解 32

2.5 编程习惯不佳 32

2.6 英语能力不足 35

2.7 管理人员误导 36

第3章 自表达代码 39

第4章 理论知识的补充 41

4.1 面向对象的基础知识 41

4.1.1 封装 41

4.1.2 继承 42

4.1.3 多态 44

4.2 设计的基本原则 45

4.2.1 单一职责原则 45

4.2.2 开放封闭原则 46

4.2.3 里氏代换原则 48

4.2.4 接口分离原则 48

4.2.5 依赖倒置原则 49

4.2.6 迪米特法则 50

4.2.7 不要重复你自己 51

4.3 圈复杂度 52

第5章 编程环境的熟悉 53

5.1 保存动作与自动格式化 53

5.2 快速修复 58

5.2.1 缺少类 59

5.2.2 缺少方法 60

5.2.3 缺少变量 61

5.2.4 类型不匹配 62

5.3 重构 63

5.3.1 改名 64

- 5.3.2 改变方法签名 68
- 5.3.3 提取方法 69
- 5.3.4 移动 73
- 5.3.5 引入参数对象 76
- 5.3.6 引入接口 77
- 5.3.7 向上提取 78
- 5.3.8 提取常量 81
- 5.3.9 局部变量变为域变量 82
- 5.3.10 引入工厂 82
- 5.4 自动完成 83
- 5.5 大纲 84
- 5.6 Eclipse快捷键 86
- 第6章 编程语言的学习 88
 - 6.1 注解 88
 - 6.2 异常 94
 - 6.3 反射 97
 - 6.4 泛型 100
- 第7章 设计方法的学习 101
 - 7.1 设计模式 101
 - 7.2 依赖注入 102
 - 7.3 Map的妙用 103
 - 7.4 采用位掩码减少类的个数 107
 - 7.5 List处理Z-Order 110
- 第8章 英文能力的提升 113
 - 8.1 词性 113
 - 8.2 时态 115
 - 8.3 语法 116
 - 8.4 否定词 117
 - 8.5 成对词 119
 - 8.6 近义词 120
 - 8.7 前缀、后缀 120
 - 8.8 缩写 122
 - 8.9 正确用词 122
- 第9章 编程习惯的改进 124
 - 9.1 命名方面的改进 124
 - 9.1.1 命名的基本原则 124
 - 9.1.2 包的命名 125
 - 9.1.3 接口的命名 125
 - 9.1.4 类的命名 125
 - 9.1.5 注解的命名 126
 - 9.1.6 枚举的命名 126
 - 9.1.7 方法的命名 127
 - 9.1.8 变量 129
 - 9.1.9 常量的命名 130
 - 9.1.10 相似的命名 130
 - 9.2 注释方面的改进 131
 - 9.2.1 JavaDoc 131
 - 9.2.2 TODO、FIXME和XXX 132
 - 9.2.3 其他注释 133
 - 9.2.4 方法的注释 133
 - 9.2.5 类的注释 133
 - 9.2.6 版权声明 134
 - 9.3 风格方面的改进 135
 - 9.3.1 缩进与对齐 135

- 9.3.2 空格与空行 138
- 9.3.3 行长与换行 140
- 9.4 结构方面的改进 140
 - 9.4.1 缩短长判定 140
 - 9.4.2 缩短长分支 144
 - 9.4.3 减少参数个数 148
 - 9.4.4 减少嵌套的层数 150
 - 9.4.5 去掉重复代码 152
 - 9.4.6 分离暧昧关系 155
 - 9.4.7 多维度变化 156
 - 9.4.8 缩短长方法 158
 - 9.4.9 正则表达式 158
 - 9.4.10 自相似对象 160
 - 9.4.11 同质化处理 161
- 9.5 架构方面的改进 162
 - 9.5.1 MVC 162
 - 9.5.2 包含关系 164
 - 9.5.3 继承关系 164
 - 9.5.4 包的划分 165
 - 9.5.5 内部类、内部接口及内部枚举 166
 - 9.5.6 匿名类 167
 - 9.5.7 有限取值范围 168
- 第10章 测试代码的自表达改进 174
 - 10.1 测试代码的可读性 175
 - 10.2 测试代码的可维护性 179
 - 10.3 测试代码的可扩展性 179
 - 10.4 测试语法 180
- 第11章 Lambda表达式 182
 - 11.1 filter 183
 - 11.2 forEach 183
 - 11.3 findAny与findFirst 184
 - 11.4 sorted 184
 - 11.5 uniqueElements 185
- 附录 186
 - • • • • (收起)

[会说话的代码_下载链接1](#)

标签

编程

代码可读性

软件工程

Programming

软件开发

程序设计

软件

计算机

评论

有一章专门讲英文能力的，如何取名。各种介词的使用，of, with, on 在 guava 和 jdk里经常看到

新手如何写代码显得有文化。

同样是写代码，你愿意写冗长的还是简洁的？

可能是我希望高了

书店里面随便翻了一下，新手看看就行了，Eclipse 快捷键都收录进去，囧。觉得还是 clean code 和重构就够了，老油条就不用买了，书还不便宜。

主要原因是java，对我不太适用

代码以java为例，有点小懵

p话多；有错误，流式api并没有可读性上的优越性；我觉得类似的代码规范的书，作者必须是非常有项目经验的，你得写过很多大型的，成功的项目；这本书读下来的体验就是特别naive

Java的代码规范

没什么干货，随便翻翻就行了

平时开发不好找资料的一些小问题的汇总！代码的命名，重构，注释的编写、改进还有代码的可测试都有。甚至还有英文时态和eclipse的快捷键。书不长，读起来很快。对了，里面的代码以Java为例

[会说话的代码 下载链接1](#)

书评

自表达代码自然具有好的可读性。代码可读的重要性已经得到几乎所有程序员的认可，特别是具有丰富经验的程序员。而对于新手，或许认识得还不够。但是如何书写自表达代码？不一定所有的程序员都知道，尤其是程序员新手，包括一些像我这样具有三四年项目经验的程序员，面对命名...

很多人都说,你们程序员写的代码像天书似的,都看懂,不过当你读到这本书的时候,你会发现,原来代码可以这样写:
hero.fight(enemy).with("Fist").当然想要写成这样不是一天就能学成的,这本书能够把你领进-书写自表达代码的道路上来. 想知道自己的代码有哪些坏味道,你也可以来读...

一个良好的编程习惯，会让你以后减少很多的麻烦，尤其是在工作初期，这本书会对自

己有很大的帮助，其实有很多的时候，我们的忙碌是因为自己给自己制造了很大的麻烦，在这一点上，我想，做对日工作的会有更多体会吧，曾经看到过一个极品，对着式样书的要求自己写代码，IFELSE写...

这本书从编程实践中总结编程技巧和经验，从艺术角度总结提高代码规范性和可读性的经验，提出了很多编程工作者编程时的问题并给出解决方法。很多编程人员在编程时只注重程序的实现，而忽略了很多可以提高编程效率和调试bug的方法。本书系统地介绍了编写漂亮代码和养成良好编程习...

这本书通俗易懂，语句简练，内容很丰富，但是要想真正掌握其中的内容，还是要需要花费一定的功夫，这本书对于自己提高的代码整洁及简练度非常有帮助，这对以后自己的代码提高有很大的帮助。
最近一直在读这本书，也掌握了一些方法，使自己的代码读起来更通顺了...

[会说话的代码 下载链接1](#)