

<<编写可读代码的艺术>>

图书基本信息

书名：<<编写可读代码的艺术>>

13位ISBN编号：9787111385448

10位ISBN编号：7111385446

出版时间：2012-7-10

出版时间：机械工业出版社华章公司

作者：Boswell, D.,Foucher, T.

页数：178

译者：尹哲,郑秀雯

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：<http://www.tushu007.com>

<<编写可读代码的艺术>>

前言

<<编写可读代码的艺术>>

内容概要

细节决定成败，思路清晰、言简意赅的代码让程序员一目了然；而格式凌乱、拖沓冗长的代码让程序员一头雾水。

除了可以正确运行以外，优秀的代码必须具备良好的可读性，编写的代码要使其他人能在最短的时间内理解才行。

本书旨在强调代码对人的友好性和可读性。

本书关注编码的细节，总结了很多提高代码可读性的小技巧，看似都微不足道，但是对于整个软件系统的开发而言，它们与宏观的架构决策、设计思想、指导原则同样重要。

编码不仅仅只是一种技术，也是一门艺术，编写可读性高的代码尤其如此。

如果你要成为一位优秀的程序员，要想开发出高质量的软件系统，必须从细处着手，做到内外兼修，本书将为你提供有效的指导。

- 简化命名、注释和格式的方法，使每行代码都言简意赅。
- 梳理程序中的循环、逻辑和变量来减小复杂度并理清思路。
- 在函数级别解决问题，例如重新组织代码块，使其一次只做一件事。
- 编写有效的测试代码，使其全面而简洁，同时可读性更高。

<<编写可读代码的艺术>>

作者简介

Dustin

Boswell毕业于加州理工大学，资深软件工程师，在Google就职多年，负责Web爬虫和程序设计相关的工作。

他专注于前端、后端，服务器架构、机器学习、大数据、系统和网站等技术领域的研究和实践，经验十分丰富。

他现在是MyLikes的软件工程师。

Trevor

Foucher资深软件工程师和技术经理，先后在Microsoft和Google工作了数十年，在Microsoft担任软件工程师、技术经理以及安全产品技术主管，在Google从事广告应用开发和搜索基础结构研发相关的工作。

。

<<编写可读代码的艺术>>

书籍目录

前言

第1章 代码应当易于理解

- 是什么让代码变得“更好”

- 可读性基本定理

- 总是越小越好吗

- 理解代码所需的时间是否与其他目标有冲突

- 最难的部分

第一部分 表面层次的改进

第2章 把信息装到名字里

- 选择专业的词

- 避免像tmp和retval这样泛泛的名字

- 用具体的名字代替抽象的名字

- 为名字附带更多信息

- 名字应该有多长

- 利用名字的格式来传递含义

- 总结

第3章 不会误解的名字

- 例子：Filter()

- 例子：Clip(text, length)

- 推荐用first和last来表示包含的范围

- 推荐用begin和end来表示包含/排除范围

- 给布尔值命名

- 与使用者的期望相匹配

- 例子：如何权衡多个备选名字

- 总结

第4章 审美

- 为什么审美这么重要

- 重新安排换行来保持一致和紧凑

- 用方法来整理不规则的东西

- 在需要时使用列对齐

- 选一个有意义的顺序，始终一致地使用它

- 把声明按块组织起来

- 把代码分成“段落”

- 个人风格与一致性

- 总结

第5章 该写什么样的注释

- 什么不需要注释

- 记录你的思想

- 站在读者的角度

- 最后的思考——克服“作者心理阻滞”

- 总结

第6章 写出言简意赅的注释

- 让注释保持紧凑

- 避免使用不明确的代词

- 润色粗糙的句子

<<编写可读代码的艺术>>

- 精确地描述函数的行为
- 用输入/输出例子来说明特别的情况
- 声明代码的意图
- “具名函数参数”的注释
- 采用信息含量高的词
- 总结

第二部分 简化循环和逻辑

第7章 把控制流变得易读

- 条件语句中参数的顺序
- if/else语句块的顺序
- ?:条件表达式(又名“三目运算符”)
- 避免do/while循环
- 从函数中提前返回
- 臭名昭著的goto
- 最小化嵌套
- 你能理解执行的流程吗
- 总结

第8章 拆分超长的表达式

- 用做解释的变量
- 总结变量
- 使用德摩根定理
- 滥用短路逻辑
- 例子：与复杂的逻辑战斗
- 拆分巨大的语句
- 另一个简化表达式的创意方法
- 总结

第9章 变量与可读性

- 减少变量
- 缩小变量的作用域
- 只写一次的变量更好
- 最后的例子
- 总结

第三部分 重新组织代码

第10章 抽取不相关的子问题

- 介绍性的例子：findClosestLocation()
- 纯工具代码
- 其他多用途代码
- 创建大量通用代码
- 项目专有的功能
- 简化已有接口
- 按需重塑接口
- 过犹不及
- 总结

第11章 一次只做一件事

- 任务可以很小
- 从对象中抽取值
- 更大型的例子

<<编写可读代码的艺术>>

总结

第12章 把想法变成代码

清楚地描述逻辑

了解函数库是有帮助的

把这个方法应用于更大的问题

总结

第13章 少写代码

别费神实现那个功能——你不会需要它

质疑和拆分你的需求

保持小代码库

熟悉你周边的库

例子：使用Unix工具而非编写代码

总结

第四部分 精选话题

第14章 测试与可读性

使测试易于阅读和维护

这段测试什么地方不对

使这个测试更可读

让错误消息具有可读性

选择好的测试输入

为测试函数命名

那个测试有什么地方不对

对测试较好的开发方式

走得太远

总结

第15章 设计并改进“分钟/小时计数器”

问题

定义类接口

尝试1：一个幼稚的方案

尝试2：传送带设计方案

尝试3：时间桶设计方案

比较三种方案

总结

附录 深入阅读

<<编写可读代码的艺术>>

章节摘录

<<编写可读代码的艺术>>

编辑推荐

《编写可读代码的艺术》写出的代码能让人快速理解、轻松维护、容易扩展的程序员才是专业的程序员。

《编写可读代码的艺术》简化命名、注释和格式的方法，使每行代码都言简意赅；梳理程序中的循环、逻辑和变量来减小复杂度并理清思路；编写有效的测试代码，使其全面而简洁，同时可读性更高。

<<编写可读代码的艺术>>

名人推荐

"软件开发的一个重要部分是要意识到你的代码以后将如何影响查看这些代码的人。两位作者高屋建瓴，带你领略这一挑战的各个方面，并且使用有指导意义的例子来解释细节。" ——Michael Hunger，软件开发人员

<<编写可读代码的艺术>>

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:<http://www.tushu007.com>