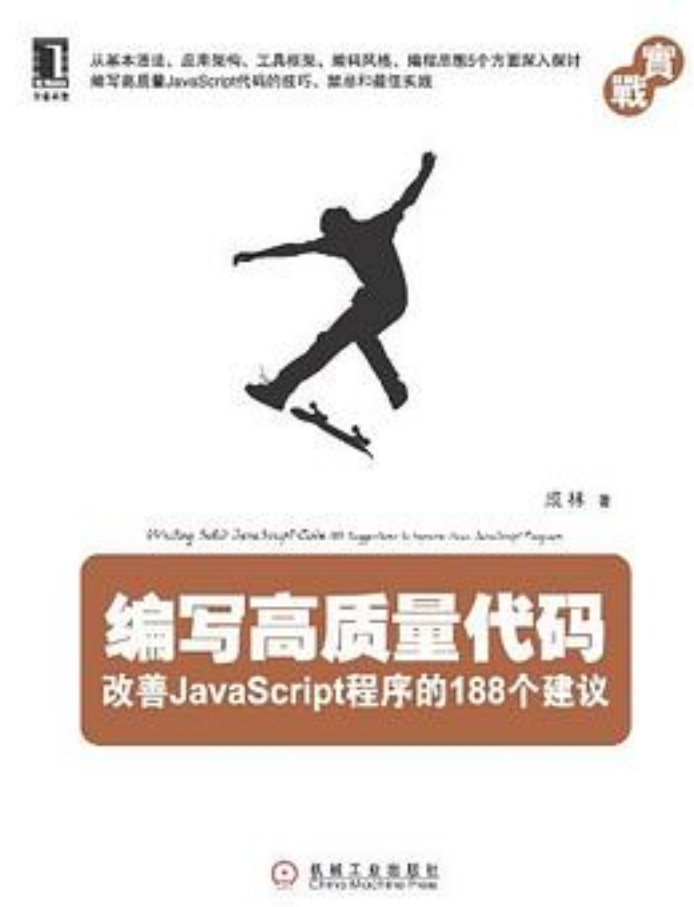


编写高质量代码



[编写高质量代码_下载链接1](#)

著者:刘一道

出版者:机械工业出版社

出版时间:2015-9

装帧:平装

isbn:9787111514633

《编写高质量代码：改善Objective-C程序的61个建议》是一本关于OBJECTIVE-C很好实践的集大成之作，它从OBJECTIVE-C语言本身、OBJECTIVE-C程序的架构设计和思想、OBJECTIVE-C程序的编码规范和习惯等三大方面对200个经典的问题给出了解决方案，

为OBJECTIVE-C开发者提高开发效率和编写高质量的OBJECTIVE-C代码提供了宝贵的建议。对于每一个建议，作者不仅给出了被实践证明为比较出色的解决方案，而且还给出了被误用或被错误理解的不好的解决方案，形成了鲜明对比。

在通往“iOS技术殿堂”的路上，本书将为你编写健壮、优雅、高质量的Objective-C代码提供极为宝贵的经验！内容全部由Objective-C编码的很好实践组成，从从基本原则、惯用法、语法、内存管理、设计、实现、设计模式、兼容性和性能优化的等方面深入探讨了编写高质量Objective-C代码的技巧、禁忌和很好实践，一共总结出61条宝贵的建议。每条建议对应Objective-C程序员可能会遇到的一个问题。本书不仅以建议的方式从正反两方面给出了被实践证明为十分出色的解决方案或非常糟糕的解决方案，而且分析了问题产生的根源，会使人有一种醍醐灌顶的感觉，豁然开朗。

本书针对每个问题所选择的应用场景都非常典型，给出的建议也都与实践紧密结合。书中的每一条建议都可能在你的下一行代码、下一个应用或下一个项目中显露锋芒。建议你將本书搁置在手边，随时查阅，相信这么做一定能使你的学习和开发工作事半功倍。

作者介绍:

刘一道

专业Mac/iOS开发工程师，有近10年软件开发经验，精通iOS、Objective-C、Xcode等与苹果开发相关的各项技术，同时对Android和Windows

Phone等移动开发技术也有深入的研究，并积累了丰富的实战经验。出版了《Objective-C程序设计入门与实践》和《iOS

7: iPhone/iPad应用开发技术详解》等著作。

目录: 前言

第1章 让自己习惯Objective-C 1

建议1: 视Objective-C为一门动态语言 1

建议2: 在头文件中尽量减少其他头文件的引用 6

建议3: 尽量使用const、enum来替换预处理#define 10

建议4: 优先使用对象字面量语法而非等效方法 13

建议5: 处理隐藏的返回类型，优先选择实例类型而非id 17

建议6: 尽量使用模块方式与多类建立复合关系 19

建议7: 明解Objective-C++中的有所为而有所不为 23

第2章 数据类型、集合和控制语句 28

建议8: C语言与Objective-C语言的关系是充分而非必要条件 28

建议9: 高度警惕空指针和野指针的袭击 31

建议10: 在64位环境下尽可能利用标记指针 35

建议11: 谨记兼容32位和64位环境下代码编写事项 38

建议12: 清楚常量字符串和一般字符串的区别 43

建议13: 在访问集合时要优先考虑使用快速枚举 44

建议14: 有序对象适宜存于数组，而无序对象适宜存于集 48

建议15: 存在公共键时，字典是在对象之间传递信息的绝佳方式 53

建议16: 明智而审慎地使用BOOL类型 55

第3章 内存管理 57

建议17: 理解内存和Objective-C内存管理规则 57

建议18: 内存管理讲究“好借好还，再借不难” 61

建议19: 区别开alloc、init、retain、release和dealloc之间的差异 63

建议20: 优先选用存取方法来简化内存管理 66

建议21: 对象销毁或者被移除一定考虑所有权的释放 70
建议22: 明智而审慎地使用dealloc 73
第4章 设计与声明 75
建议23: 编写代码要遵守Cocoa API约定 75
建议24: 洞悉实例变量 77
建议25: 透彻了解属性的里里外外 81
建议26: 存取方法是良好的类接口必要组成部分 85
建议27: 明晓类公共领域的方法都是虚方法 87
建议28: 初始化还是解码取决于是否支持归档和解档 92
建议29: 利用键-值机制访问类的私有成员变量和方法 93
建议30: 浅复制适宜指针而深复制适宜数据 101
建议31: 明智而审慎地使用NSCopying 103
建议32: 使用协议来实现匿名对象的提供 106
第5章 实现 108
建议33: 使用类别把类的实现拆分成不同的文件 108
建议34: 明智地使用内省可使程序更加高效和健壮 109
建议35: 尽量使用不可变性对象而非可变性对象 113
建议36: 利用复合能巧妙地把两个类或两个对象融合 115
建议37: 使用类扩展来隐藏实现的细节 120
建议38: 使用内联块应注意避免循环引用 122
建议39: 利用类别把方法添加到现有的类 124
建议40: 通过强弱引用来管理对象的所有权 127
第6章 继承与面向对象设计 133
建议41: 明确isa在继承上的作用 133
建议42: 利用类别和协议实现类似多重继承的机制 136
建议43: 类别和类扩展是类继承的延续性拓展 139
建议44: 继承基类的实现行为勿忘调用super 141
第7章 设计模式与Cocoa编程 145
建议45: 设计模式是特定环境下的特定问题的解决方案 145
建议46: MVC模式是一种复合或聚合模式 147
建议47: 对象建模在数据库中也广泛使用 155
建议48: 类簇可简化框架的公开架构而又不减少功能的丰富性 160
建议49: 委托用于界面控制, 而数据源用于数据控制 165
第8章 定制init...和dealloc 171
建议50: 了解对象的alloc和init... 171
建议51: 直接访问实例变量的init...方法 174
建议52: 初始化方法必须以“init”字母开头 176
建议53: 从init...方法得到的对象可能是不想要的 177
建议54: 实现init...方法的唯一性或者指定性并非“不可能” 179
建议55: init...方法有“轻重级别”之分 181
第9章 Objective-C与Swift的兼容性 184
建议56: Objective-C和Swift的互用性基于映射机制 184
建议57: 利用Swift的特性可增强已有的Objective-C代码 191
建议58: 洞悉Objective-C和Swift类型转换的处理机制 194
建议59: C语言的数据类型在Swift中“有所变有所不变” 199
建议60: Swift和Objective-C兼容性是基于混搭机制 204
建议61: 利用迁移机制实现Objective-C代码的重生 209
• • • • • ([收起](#))

[编写高质量代码_下载链接1](#)

标签

iOS

计算机

技术

评论

烂书

间断性地在读，反复读，希望能够读完，基础得到强化

61个建议里有小一半都是敷衍，但是其中有 10 个真的不错的建议，学到了不少～

Objective-C编写的程序不能直接编译成令机器读懂的机器语言？？？谁告诉你的

书是16年出的，但是书中有些内容还停留在12、13年的技术上。。。

不适合初中级开发者阅读，错误较多

[编写高质量代码_下载链接1](#)

书评

[编写高质量代码_下载链接1](#)