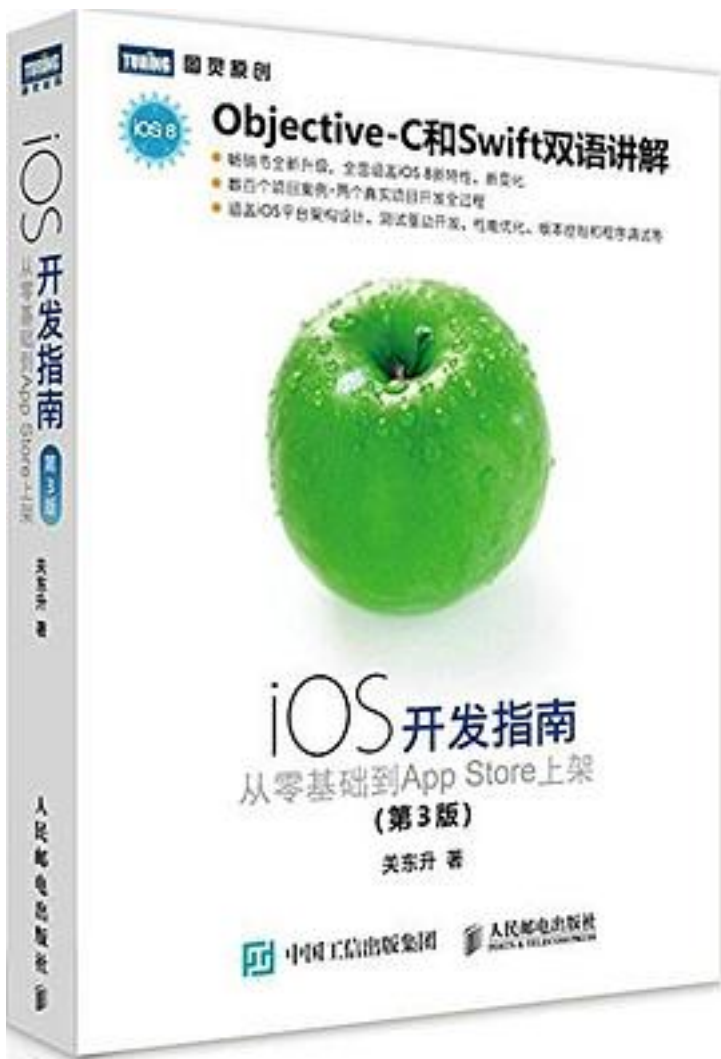


iOS开发指南 从零基础到App Store上架（第3版）



[iOS开发指南 从零基础到App Store上架（第3版） 下载链接1](#)

著者:关东升

出版者:人民邮电出版社

出版时间:2015-4-1

装帧:平装

isbn:9787115387769

本书以Swift和Objective-C语言为基础进行讲解，共分4篇：第一篇为基础篇，主要讲解iOS的一些基础知识；第二篇为网络篇，主要讲解iOS网络开发的相关知识；第三篇为进阶篇，介绍了iOS高级内容、商业思考等；最后一篇是实战篇，通过两个真实的项目，带领读者身临其境地完成一个项目的开发，使读者了解整个开发过程以及开发中的技巧。

作者介绍:

关东升

智捷课堂创始人，国内知名iOS技术作家，iOS技术顾问，高级培训讲师，移动开发专家。担任51CTO社区iOS技术顾问，精通iOS、Android和Windows Phone及HTML5等移动开发技术。曾先后主持开发大型网络游戏《神农诀》的iOS和Android客户端开发，国家农产品追溯系统的iPad客户端开发，酒店预订系统的iPhone客户端开发，金融系统微博的iOS、Windows Phone及Android客户端开发。在App Store上发布多款游戏和应用软件，擅长移动平台的应用和游戏类项目开发。长期为中国移动研究院、云南移动、东软、方正科技、惠普、NTT、深圳康拓普、上海财富168、天津港务局、福州软件园、广州软件园等企事业单位授课。除本书外，还著有《Swift开发指南》《iOS网络编程与云端应用最佳实践》《iOS传感器应用开发最佳实践》《iOS图形图像、动画和多媒体编程技术最佳实践》《iPhone与iPad开发实战——iOS经典应用剖析》《品味移动设计》《交互设计的艺术》《Android开发案例驱动教程》、《Android网络游戏开发实战》，以及《Cocos2d-x实战》系列丛书等。

目录: 第一部分 基础篇

第1章 开篇综述 2

1.1 iOS概述 2

1.1.1 iOS介绍 2

1.1.2 iOS 8的新特性 2

1.2 开发环境及开发工具 3

1.3 本书中的约定 4

1.3.1 案例代码约定 4

1.3.2 图示的约定 5

1.3.3 方法命名约定 6

1.3.4 构造器命名约定 7

第2章 第一个iOS应用程序 9

2.1 创建HelloWorld工程 9

2.1.1 创建工程 9

2.1.2 Xcode中的iOS工程模板 14

2.1.3 应用剖析 15

2.2 HelloWorld工程中的故事板文件 17

2.2.1 故事板的导航特点 17

2.2.2 故事板中的Scene和Segue 19

2.3 应用生命周期 19

2.3.1 非运行状态——应用启动场景 21

2.3.2 点击Home键——应用退出场景 22

2.3.3 挂起重新运行场景 24

2.3.4 内存清除——应用终止场景 25

2.4 视图生命周期 25

2.5 设置产品属性 27

2.5.1 Xcode中的PROJECT和TARGET 27

2.5.2 设置常用的产品属性 29

2.6 iOS API简介 30

2.6.1 API概述	31
2.6.2 如何使用API帮助	33
2.7 小结	36
第3章 UIView与视图	37
3.1 视图“始祖”——UIView	37
3.1.1 UIView“家族”	37
3.1.2 应用界面的构建层次	39
3.1.3 视图分类	40
3.2 标签与按钮	40
3.2.1 标签控件	41
3.2.2 按钮控件	42
3.3 动作与输出口	44
3.4 TextField和TextView	48
3.4.1 TextField控件	48
3.4.2 TextView控件	49
3.4.3 键盘的打开和关闭	50
3.4.4 关闭和打开键盘的通知	50
3.4.5 键盘的种类	51
3.5 开关控件、滑块控件和分段控件	52
3.5.1 开关控件	53
3.5.2 滑块控件	53
3.5.3 分段控件	54
3.6 网页视图WebView	54
3.7 等待相关的控件与进度条	57
3.7.1 活动指示器ActivityIndicatorView	58
3.7.2 进度条ProgressView	59
3.8 警告框和操作表	60
3.8.1 警告框	60
3.8.2 操作表ActionSheet	62
3.8.3 使用iOS 8 UIAlertController实现	63
3.9 工具栏和导航栏	64
3.9.1 工具栏	64
3.9.2 导航栏	67
3.10 小结	70
第4章 iOS 8多分辨率屏幕适配	71
4.1 iOS 8屏幕的多样性	71
4.1.1 iOS 8屏幕介绍	71
4.1.2 iOS 8的三种分辨率	72
4.1.3 获得iOS设备屏幕信息	73
4.2 iOS屏幕布局	74
4.2.1 iOS中的“栏”	74
4.2.2 传统布局	75
4.3 Auto Layout布局	77
4.3.1 在Interface Builder中管理Auto Layout约束	77
4.3.2 实例：Auto Layout布局	78
4.4 Size Class与iOS 8多屏幕适配	83
4.4.1 在Interface Builder中使用Size Class	83
4.4.2 Size Class的九宫格	84
4.4.3 实例：使用Size Class	85
4.5 在屏幕滚动视图中使用Auto Layout和Size Class	90
4.5.1 屏幕滚动视图的重要属性	90
4.5.2 屏幕滚动视图实例	92
4.6 使用资源目录（asset catalog）管理图片	97
4.7 小结	100

第5章 委托协议、数据源协议与高级视图 101

5.1 视图中的委托协议和数据源协议 101

5.2 选择器 101

5.2.1 日期选择器 101

5.2.2 普通选择器 103

5.2.3 数据源协议与委托协议 106

5.3 集合视图 108

5.3.1 集合视图介绍 108

5.3.2 实例：奥运会比赛项目 109

5.3.3 添加集合视图控制器 110

5.3.4 添加集合视图单元格 113

5.3.5 数据源协议与委托协议 116

5.4 小结 117

第6章 表视图 118

6.1 概述 118

6.1.1 表视图的组成 118

6.1.2 表视图的相关类 119

6.1.3 表视图分类 119

6.1.4 单元格的组成和样式 121

6.1.5 数据源协议与委托协议 123

6.2 简单表视图 123

6.2.1 创建简单表视图 123

6.2.2 自定义单元格 128

6.2.3 添加搜索栏 131

6.3 分节表视图 137

6.3.1 添加索引 137

6.3.2 分组 140

6.3.3 静态表与控制界面布局 141

6.4 修改单元格 144

6.4.1 删除和插入单元格 144

6.4.2 移动单元格 151

6.5 表视图UI设计模式 153

6.5.1 分页模式 153

6.5.2 下拉刷新模式 154

6.5.3 下拉刷新控件 154

6.6 小结 157

第7章 视图控制器与导航模式 158

7.1 概述 158

7.1.1 视图控制器的种类 158

7.1.2 导航模式 158

7.1.3 模态视图 160

7.2 平铺导航 166

7.2.1 应用场景 166

7.2.2 基于分屏导航的实现 168

7.2.3 基于电子书导航的实现 173

7.3 标签导航 178

7.3.1 应用场景 178

7.3.2 实现 179

7.4 树形结构导航 182

7.4.1 应用场景 182

7.4.2 实现 183

7.5 组合使用导航模式 190

7.5.1 应用场景 190

7.5.2 实现 191

- 7.6 小结 198
- 第8章 iOS常用设计模式 199
 - 8.1 单例模式 199
 - 8.1.1 问题提出 199
 - 8.1.2 Objective-C实现原理 199
 - 8.1.3 Swift实现原理 200
 - 8.1.4 应用案例 201
 - 8.2 委托模式 202
 - 8.2.1 问题提出 202
 - 8.2.2 实现原理 204
 - 8.2.3 应用案例 207
 - 8.3 观察者模式 211
 - 8.3.1 问题提出 211
 - 8.3.2 实现原理 211
 - 8.3.3 通知机制和KVO机制 212
 - 8.4 MVC 模式 218
 - 8.4.1 MVC模式概述 218
 - 8.4.2 Cocoa Touch中的MVC模式 219
 - 8.5 小结 221
- 第9章 iPhone与iPad应用开发的差异 222
 - 9.1 概述 222
 - 9.1.1 应用场景差异 222
 - 9.1.2 设计和开发需注意的问题 222
 - 9.2 iPad专用API 225
 - 9.2.1 UIPopoverController控制器 225
 - 9.2.2 UISplitViewController控制器 229
 - 9.3 小结 234
- 第10章 iOS分层架构设计 235
 - 10.1 低耦合企业级系统架构设计 235
 - 10.2 iOS分层架构设计 236
 - 10.2.1 工程与工作空间 236
 - 10.2.2 静态链接库 237
 - 10.2.3 框架 237
 - 10.2.4 10种分层模式 238
 - 10.3 实例：MyNotes应用 238
 - 10.4 基于同一工程的分层 240
 - 10.4.1 采用纯Swift语言实现——PSSS模式 240
 - 10.4.2 采用纯Objective-C语言实现——POOO模式 243
 - 10.4.3 采用Swift与Objective-C混合搭配实现——PSOO模式 247
 - 10.5 基于同一工作空间静态链接库实现的WLSOO模式 247
 - 10.5.1 使用工作空间 247
 - 10.5.2 添加数据持久层工程 248
 - 10.5.3 建立业务逻辑层与数据持久层的依赖关系 249
 - 10.5.4 建立表示层与业务逻辑层的依赖关系 251
 - 10.6 基于同一工作空间框架实现的WFSSS模式 251
 - 10.6.1 创建框架工程 251
 - 10.6.2 依赖关系建立 253
 - 10.6.3 代码重构 254
 - 10.7 基于同一工作空间框架实现的WFOOO模式 255
 - 10.7.1 设置Public头文件 255
 - 10.7.2 建立表示层与业务逻辑层的依赖关系 256
 - 10.7.3 代码重构 256
 - 10.8 基于同一工作空间框架实现的WFSOO模式 256
 - 10.8.1 设置Public头文件 256

10.8.2 建立表示层与业务逻辑层的依赖关系	257
10.8.3 代码重构	257
10.9 基于同一工作空间框架实现的WFOSS模式	258
10.9.1 设置Public头文件	258
10.9.2 代码重构	258
10.10 小结	258
第11章 数据持久化	259
11.1 概述	259
11.1.1 沙箱目录	259
11.1.2 持久化方式	260
11.2 属性列表	260
11.3 对象归档	264
11.4 使用SQLite数据库	268
11.4.1 SQLite数据类型	268
11.4.2 创建数据库	269
11.4.3 查询数据	271
11.4.4 修改数据	273
11.5 Core Data	276
11.5.1 ORM	276
11.5.2 Core Data堆栈	276
11.5.3 建模和生成实体	280
11.5.4 采用Core Data分层架构设计	283
11.5.5 查询数据	285
11.5.6 修改数据	286
11.6 小结	288
第12章 应用程序设置	289
12.1 概述	289
12.1.1 设置	289
12.1.2 配置	290
12.2 应用程序设置包	291
12.3 设置项目种类	293
12.3.1 文本字段	295
12.3.2 开关	298
12.3.3 滑块	299
12.3.4 值列表	300
12.3.5 子界面	301
12.4 读取设置	303
12.5 小结	304
第13章 国际化	305
13.1 概述	305
13.1.1 需要国际化的内容	305
13.1.2 国际化目录结构	307
13.2 文本信息国际化	308
13.2.1 系统按钮和信息国际化	308
13.2.2 应用名称国际化	309
13.2.3 程序代码输出的静态文本国际化	311
13.2.4 使用genstring工具	312
13.3 xib和故事板文件国际化	313
13.3.1 使用Base国际化技术	313
13.3.2 Auto Layout与国际化	315
13.4 资源文件国际化	317
13.4.1 图片资源文件国际化	317
13.4.2 声音资源文件国际化	318
13.5 小结	318

第14章 访问联系人	319
14.1 概述	319
14.2 读取联系人信息	320
14.2.1 查询联系人记录	321
14.2.2 读取单值属性	324
14.2.3 读取多值属性	326
14.2.4 读取图片属性	328
14.3 写入联系人信息	329
14.3.1 创建联系人	330
14.3.2 修改联系人	332
14.3.3 删除联系人	333
14.4 高级API	335
14.4.1 选择联系人	335
14.4.2 显示和修改联系人	336
14.4.3 创建联系人	339
14.5 小结	342
第二部分 网络篇	
第15章 访问Web Service	344
15.1 概述	344
15.2 数据交换格式	344
15.2.1 XML文档结构	346
15.2.2 解析XML文档	347
15.2.3 JSON文档结构	355
15.2.4 JSON数据解码	356
15.3 REST Web Service	357
15.3.1 HTTP和HTTPS协议	358
15.3.2 同步GET请求方法	358
15.3.3 异步GET请求方法	361
15.3.4 POST请求方法	362
15.3.5 调用REST Web Service的插入、修改和删除方法	363
15.4 使用轻量级网络请求框架MKNetworkKit	368
15.4.1 ASIHTTPRequest、AFNetworking和MKNetworkKit比较	368
15.4.2 安装和配置MKNetworkKit框架	369
15.4.3 网络请求	370
15.4.4 下载数据	371
15.4.5 上传数据	373
15.5 反馈网络信息改善用户体验	375
15.5.1 使用下拉刷新控件改善用户体验	375
15.5.2 使用活动指示器控件	378
15.5.3 使用网络活动指示器	380
15.6 小结	380
第16章 定位服务与地图应用	381
16.1 定位服务	381
16.1.1 定位服务编程	382
16.1.2 地理信息反编码	386
16.1.3 地理信息编码查询	388
16.1.4 关于定位服务的测试	390
16.2 使用iOS苹果地图	393
16.2.1 显示地图	393
16.2.2 添加标注	395
16.2.3 跟踪用户位置变化	398
16.3 使用程序外地图	399
16.3.1 调用iOS苹果地图	400
16.3.2 调用谷歌Web地图	402

16.4 小结	404
第三部分 进阶篇	
第17章 iOS中的商业模式	406
17.1 收费策略	406
17.1.1 iOS如何赚钱	406
17.1.2 避免定价策略误区	406
17.1.3 免费软件的艺术	407
17.1.4 在适当的时间、适当的地点植入广告	407
17.1.5 尝试不同的盈利模式	407
17.2 使用苹果iAd 广告	408
17.2.1 横幅广告	408
17.2.2 插页广告	411
17.2.3 查看你的收入	417
17.3 使用谷歌AdMob广告	419
17.3.1 注册AdMob账号	419
17.3.2 管理AdMob广告	419
17.3.3 AdMob广告类型	422
17.3.4 下载AdMob Ads SDK	423
17.3.5 配置AdMob开发环境	424
17.3.6 AdMob横幅广告	426
17.3.7 添加AdMob插页广告	428
17.3.8 为广告提交用户和位置信息	431
17.3.9 搜索广告	431
17.4 小结	433
第18章 找出程序中的bug——调试	434
18.1 Xcode调试工具	434
18.1.1 定位编译错误	434
18.1.2 查看和显示日志	435
18.1.3 设置和查看断点	437
18.1.4 调试工具	442
18.1.5 输出窗口	443
18.1.6 变量查看窗口	444
18.1.7 查看线程	445
18.2 日志与断言输出	446
18.2.1 使用NSLog函数	446
18.2.2 使用断言	447
18.3 异常堆栈报告分析	448
18.3.1 跟踪异常堆栈	448
18.3.2 分析堆栈报告	450
18.4 在iOS设备上调试	450
18.4.1 创建开发者证书	451
18.4.2 设备注册	455
18.4.3 创建App ID	456
18.4.4 创建配置概要文件	459
18.4.5 设备调试	460
18.5 Xcode设备管理工具	461
18.5.1 查看设备上的应用程序	462
18.5.2 设备日志	464
18.6 小结	464
第19章 测试驱动下的iOS应用开发	465
19.1 测试驱动的软件开发概述	465
19.1.1 测试驱动的软件开发流程	465
19.1.2 测试驱动的软件开发案例	466
19.1.3 iOS单元测试框架	467

19.2 使用XCTest测试框架	468
19.2.1 添加XCTest到工程	468
19.2.2 编写XCTest测试方法	469
19.2.3 运行测试程序	472
19.2.4 分析测试报告	473
19.3 异步单元测试	475
19.4 性能测试	478
19.4.1 配置测试环境	478
19.4.2 测试代码	479
19.4.3 分析测试结果	480
19.5 iOS单元测试最佳实践	481
19.5.1 测试数据持久层	481
19.5.2 测试业务逻辑层	485
19.6 小结	486
第20章 让你的程序“飞”起来——性能优化	487
20.1 内存优化	487
20.1.1 内存管理	487
20.1.2 使用Analyze和Instruments工具解决内存泄漏问题	487
20.1.3 查找和解决僵尸对象	494
20.1.4 autorelease的使用问题	497
20.1.5 响应内存警告	497
20.2 优化资源文件	499
20.2.1 图片文件优化	499
20.2.2 音频文件优化	500
20.3 延迟加载	501
20.3.1 资源文件的延迟加载	501
20.3.2 故事板文件的延迟加载	505
20.3.3 xib文件的延迟加载	507
20.4 数据持久化的优化	509
20.4.1 使用文件	509
20.4.2 使用SQLite数据库	513
20.4.3 使用Core Data	514
20.5 可重用对象的使用	516
20.5.1 表视图中的可重用对象	516
20.5.2 集合视图中的可重用对象	518
20.5.3 地图视图中的可重用对象	519
20.6 并发处理与多核CPU	520
20.6.1 主线程阻塞问题	520
20.6.2 选择NSThread还是GCD	521
20.7 小结	521
第21章 管理好你的程序代码——代码版本控制	522
21.1 概述	522
21.1.1 版本控制历史	522
21.1.2 基本概念	523
21.2 Git代码版本控制	523
21.2.1 服务器搭建	523
21.2.2 Gitolite服务器管理	525
21.2.3 Git常用命令	527
21.2.4 Git分支	529
21.2.5 Git协同开发	533
21.2.6 Xcode中Git的配置与使用	535
21.3 GitHub代码托管服务	542
21.3.1 创建和配置GitHub账号	542
21.3.2 创建代码库	544

- 21.3.3 删除代码库 546
- 21.3.4 派生代码库 546
- 21.3.5 使用GitHub协同开发 549
- 21.3.6 管理组织 555
- 21.4 小结 558
- 第22章 把你的应用放到App Store上 559
 - 22.1 收官 559
 - 22.1.1 在Xcode 6中添加图标 559
 - 22.1.2 在Xcode 6中添加启动界面 562
 - 22.1.3 调整Identity和Deployment Info属性 564
 - 22.1.4 为发布进行编译 567
 - 22.2 发布流程 571
 - 22.2.1 创建应用 572
 - 22.2.2 基本信息输入 574
 - 22.2.3 应用定价 577
 - 22.2.4 上传应用 577
 - 22.2.5 提交审核 579
 - 22.3 常见审核不通过的原因 581
 - 22.3.1 功能问题 581
 - 22.3.2 用户界面问题 581
 - 22.3.3 商业问题 581
 - 22.3.4 不当内容 581
 - 22.3.5 其他问题 582
 - 22.4 小结 582
- 第四部分 实战篇
- 第23章 重构MyNotes应用——iOS网络通信中的设计模式与架构设计 584
 - 23.1 移动网络通信应用的分层架构设计 584
 - 23.2 Objective-C版本：在工作空间中添加和配置MKNetworkKit 585
 - 23.3 基于委托模式实现 587
 - 23.3.1 网络通信与委托模式 587
 - 23.3.2 在异步网络通信中使用委托模式实现分层架构设计 587
 - 23.3.3 类图 588
 - 23.3.4 时序图 591
 - 23.3.5 数据持久层重构 593
 - 23.3.6 业务逻辑层的代码实现 597
 - 23.3.7 表示层的代码实现 599
 - 23.4 基于观察者模式的通知机制实现 603
 - 23.4.1 观察者模式的通知机制回顾 603
 - 23.4.2 异步网络通信中通知机制的分层架构设计 604
 - 23.4.3 类图 605
 - 23.4.4 时序图 606
 - 23.4.5 数据持久层的重构 609
 - 23.4.6 业务逻辑层的代码实现 611
 - 23.4.7 表示层的代码实现 612
 - 23.5 小结 616
- 第24章 iOS敏捷开发项目实战——2016里约热内卢奥运会应用开发及App Store发布 617
 - 24.1 应用分析与设计 617
 - 24.1.1 应用概述 617
 - 24.1.2 需求分析 617
 - 24.1.3 原型设计 618
 - 24.1.4 数据库设计 619
 - 24.1.5 架构设计 619
 - 24.2 iOS敏捷开发 620
 - 24.2.1 敏捷开发宣言 620

24.2.2 iOS适合敏捷开发? 621
24.2.3 iOS敏捷开发最佳实践 621
24.3 任务1: 创建应用工作空间和工程 623
24.3.1 迭代1.1: 创建Swift版本的工程 623
24.3.2 迭代1.2: 创建Objective-C版本工作空间 623
24.3.3 迭代1.3: 发布到GitHub 623
24.4 任务2: 信息系统层与持久层开发 624
24.4.1 迭代2.1: 编写数据库DDL脚本 624
24.4.2 迭代2.2: 插入初始数据到数据库 625
24.4.3 迭代2.3: 编写实体类 625
24.4.4 迭代2.4: DAO类XCTest单元测试 626
24.4.5 迭代2.5: 数据库帮助类DBHelper 629
24.4.6 迭代2.5: 编写DAO类 632
24.4.7 迭代2.6: 发布到GitHub 635
24.5 任务3: 业务逻辑层开发 636
24.5.1 迭代3.1: 对比赛项目业务逻辑类进行XCTest单元测试 636
24.5.2 迭代3.2: 编写比赛项目业务逻辑类 637
24.5.3 迭代3.3: 对比赛日程业务逻辑类进行XCTest单元测试 638
24.5.4 迭代3.4: 编写比赛日程业务逻辑类 639
24.5.5 迭代3.5: 发布到GitHub 640
24.6 任务4: 表示层开发 641
24.6.1 迭代4.1: 使用资源目录管理图片和图标资源 641
24.6.2 迭代4.2: 根据原型设计初步设计iPad故事板 642
24.6.3 迭代4.3: 根据原型设计初步设计iPhone故事板 644
24.6.4 迭代4.4: 首页模块 644
24.6.5 迭代4.5: 比赛项目模块 645
24.6.6 迭代4.6: 比赛日程模块 649
24.6.7 迭代4.7: 倒计时模块表示层 652
24.6.8 迭代4.8: 关于我们模块表示层 653
24.6.9 迭代4.9: 发布到GitHub 654
24.7 任务5: 收工 654
24.7.1 迭代5.1: 添加图标 654
24.7.2 迭代5.2: 设计和添加启动界面 654
24.7.3 迭代5.3: 植入谷歌AdMob横幅广告 655
24.7.4 迭代5.4: 性能测试与改善 657
24.7.5 迭代5.5: 发布到GitHub上 658
24.7.6 迭代5.6: 在App Store上发布应用 658
24.8 小结 658
• • • • • (收起)

[iOS开发指南 从零基础到App Store上架（第3版） 下载链接1](#)

标签

iOS

编程

ios

iOS入门

Objective-C

计算机

技术

已有纸书

评论

胜在双语，定位有点尴尬，不高不低的

肯定不值这个价……

说是入门书，讲的并不深，真是入门又看不太懂，好尴尬。

入门的时候看过，现在又溜了一遍，用来找找那些已经忘记的不常用的知识点。讲的浅，有些许过时

这本书根本不是给零基础看的，啥都说，一些iOS的、设计模式、单元测试、开发理念，就是不说oc或者swift语法，泛泛而谈，而且泛而不精，水的内容很多，只能一星差评了

一般

讲的内容比较泛泛，比较简单，只是把一些用法简单的demo出来，入门教材中的初级水平

这书真的没意思 太简单了

有些知识点讲的很简单，但是真的太简单了

[iOS开发指南 从零基础到App Store上架（第3版） 下载链接1](#)

书评

[iOS开发指南 从零基础到App Store上架（第3版） 下载链接1](#)