

《iOS开发指南》

图书基本信息

书名：《iOS开发指南》

13位ISBN编号：9787115423180

出版时间：2016-6

作者：关东升

页数：816

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《iOS开发指南》

内容概要

本书是iOS开发权威指南，以Swift和Objective-C语言为基础，分5部分讲解如何从零起步编写并上线iOS 9应用。第一部分介绍iOS开发基础知识，第二部分介绍设计与架构的相关知识，第三部分介绍了iOS的一些高级知识，第四部分介绍测试、调试和优化等相关知识，第五部分为实战篇，涵盖代码版本管理、项目依赖管理、App Store发布流程，以及两个真实iOS应用的分析设计、编程、测试与分布过程

作者简介

关东升

国内著名iOS和Cocos技术作家、iOS技术顾问、Cocos最有价值专家（CVP）、智捷课堂首席培训专家、51CTO社区iOS技术顾问。他精通iOS、Android和Windows及HTML5等移动开发技术，擅长应用和游戏类项目开发，曾先后主持开发：大型网络游戏《神农诀》的iOS和Android客户端，国家农产品追溯系统的iPad客户端，酒店预订系统的iOS客户端，金融系统的iOS、Windows及Android客户端。与此同时，他长期为中国移动研究院、方正科技、大唐电信、中国石油、工商银行、南方航空、惠普、东软集团、NTT等企事业单位提供技术咨询及员工培训。

另著有：

《iOS开发指南》

《从零开始学Swift》

《iOS传感器应用开发最佳实践》

《iOS网络编程与云端应用最佳实践》

《iOS图形图像、动画和多媒体编程技术最佳实践》

《品味移动设计》

《交互设计的艺术》

《iPhone与iPad开发实战》

《Android开发案例驱动教程》

《Android网络游戏开发实战》

《Cocos2d-x实战：C++卷》

《Cocos2d-x实战：JS卷》

《Cocos2d-x实战：Lua卷》

《Cocos2d-x实战：工具卷》

书籍目录

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 9新特性	2
1.2 开发环境及开发工具	3
1.3 本书约定	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 构造函数命名约定	7
1.3.5 错误处理约定	8
第2章 第一个iOS应用程序	10
2.1 创建HelloWorld工程	10
2.1.1 创建工程	10
2.1.2 Xcode中的iOS工程模板	14
2.1.3 应用剖析	15
2.2 应用生命周期	17
2.2.1 非运行状态——应用启动场景	19
2.2.2 点击Home键——应用退出场景	20
2.2.3 挂起重新运行场景	22
2.2.4 内存清除：应用终止场景	23
2.3 设置产品属性	23
2.3.1 Xcode中的工程和目标	23
2.3.2 设置常用的产品属性	26
2.4 iOS API简介	27
2.4.1 API概述	27
2.4.2 如何使用API帮助	29
2.5 小结	32
第3章 Cocoa Touch框架与构建应用界面	33
3.1 视图控制器	33
3.1.1 视图控制器种类	33
3.1.2 视图的生命周期	33
3.2 视图	35
3.2.1 UIView继承层次结构	35
3.2.2 视图分类	37
3.2.3 应用界面构建层次	37
3.3 使用故事板构建界面	38
3.3.1 什么是故事板	39
3.3.2 场景和过渡	41
3.4 使用XIB文件构建界面	42
3.4.1 重构HelloWorld	42
3.4.2 XIB与故事板比较	45
3.5 使用纯代码构建界面	45
3.5.1 重构HelloWorld	46
3.5.2 视图的几个重要属性	47
3.6 3种构建界面技术讨论	49

3.6.1	所见即所得	49
3.6.2	原型驱动开发	49
3.6.3	团队协作开发	49
3.7	小结	50
第4章	UIView与视图	51
4.1	标签与按钮	51
4.1.1	Interface Builder实现	51
4.1.2	代码实现	55
4.2	事件处理	56
4.2.1	Interface Builder实现	56
4.2.2	代码实现	59
4.3	访问视图	60
4.3.1	Interface Builder实现	60
4.3.2	代码实现	62
4.4	TextField和TextView	63
4.4.1	Interface Builder实现	64
4.4.2	代码实现	65
4.4.3	键盘的打开和关闭	67
4.4.4	关闭和打开键盘的通知	68
4.4.5	键盘的种类	69
4.5	开关控件、分段控件和滑块控件	70
4.5.1	开关控件	71
4.5.2	分段控件	72
4.5.3	滑块控件	74
4.6	Web视图：WKWebView类	76
4.7	警告框和操作表	80
4.7.1	UIAlertController实现警告框	81
4.7.2	UIAlertController实现操作表	83
4.8	等待相关的控件与进度条	84
4.8.1	活动指示器ActivityIndicatorView	85
4.8.2	进度条ProgressView	87
4.9	工具栏和导航栏	90
4.9.1	工具栏	90
4.9.2	导航栏	94
4.10	小结	98
第5章	委托协议、数据源协议与高级视图	99
5.1	视图中的委托协议和数据源协议	99
5.2	选择器	99
5.2.1	日期选择器	99
5.2.2	普通选择器	104
5.2.3	数据源协议与委托协议	109
5.3	集合视图	111
5.4	实例：Interface Builder实现奥运会比赛项目	112
5.4.1	添加集合视图	113
5.4.2	添加集合视图单元格	114
5.4.3	数据源协议与委托协议	118
5.5	实例：代码实现奥运会比赛项目	120
5.5.1	添加集合视图	120
5.5.2	自定义集合视图单元格	122

5.6	小结	123
第6章	表视图	124
6.1	概述	124
6.1.1	表视图的组成	124
6.1.2	表视图的相关类	125
6.1.3	表视图分类	125
6.1.4	单元格的组成和样式	127
6.1.5	数据源协议与委托协议	129
6.2	简单表视图	129
6.2.1	实现协议方法	130
6.2.2	UIViewController根视图控制器	130
6.2.3	UITableViewController根视图控制器	137
6.3	自定义表视图单元格	140
6.3.1	Interface Builder实现	141
6.3.2	代码实现	145
6.4	添加搜索栏	146
6.5	分节表视图	150
6.5.1	添加索引	150
6.5.2	分组	153
6.6	静态表与界面布局	154
6.7	插入和删除单元格	160
6.7.1	Interface Builder实现	162
6.7.2	代码实现	167
6.8	移动单元格	169
6.9	表视图UI设计模式	171
6.9.1	分页模式	171
6.9.2	下拉刷新模式	171
6.9.3	下拉刷新控件	172
6.10	小结	174
第7章	界面布局与屏幕适配	175
7.1	界面布局概述	175
7.2	iOS界面布局设计模式	176
7.2.1	表单布局模式	176
7.2.2	列表布局模式	176
7.2.3	网格布局模式	177
7.3	传统布局技术	177
7.4	Auto Layout布局技术	179
7.4.1	Interface Builder中管理Auto Layout约束	179
7.4.2	实例：Auto Layout布局	180
7.5	iOS 9堆视图StackView	185
7.5.1	堆视图与布局	185
7.5.2	实例：堆视图布局	186
7.6	iOS屏幕的多样性	191
7.6.1	iOS屏幕介绍	191
7.6.2	iOS的3种分辨率	192
7.6.3	获得iOS设备屏幕信息	193
7.7	Size Class与iOS多屏幕适配	194
7.7.1	Interface Builder中使用Size Class	195
7.7.2	Size Class的九宫格	195

7.7.3	实例：使用Size Class	196
7.8	资源目录与图片资源适配	201
7.9	小结	204
第8章	视图控制器与导航模式	205
8.1	概述	205
8.1.1	视图控制器的种类	205
8.1.2	导航模式	205
8.2	模态视图	207
8.2.1	Interface Builder实现	208
8.2.2	代码实现	214
8.3	平铺导航	215
8.3.1	应用场景	215
8.3.2	基于分屏导航的实现	217
8.3.3	基于电子书导航的实现	223
8.4	标签导航	228
8.4.1	应用场景	229
8.4.2	Interface Builder实现	230
8.4.3	代码实现	234
8.5	树形结构导航	235
8.5.1	应用场景	235
8.5.2	Interface Builder实现	236
8.5.3	代码实现	244
8.6	组合使用导航模式	245
8.6.1	应用场景	246
8.6.2	Interface Builder实现	246
8.6.3	代码实现	254
8.7	小结	257
第9章	iPad应用开发	258
9.1	iPad与iPhone应用开发的差异	258
9.1.1	应用场景不同	258
9.1.2	导航模式不同	258
9.1.3	API不同	258
9.2	iPad树形结构导航	259
9.2.1	邮件应用中的树形结构导航	259
9.2.2	Master-Detail应用程序模板	260
9.2.3	Interface Builder实现SplitViewSample案例	266
9.2.4	代码实现SplitViewSample案例	270
9.3	iPad模态视图	273
9.3.1	邮件应用中的模态导航	273
9.3.2	iPad模态导航相关API	275
9.3.3	Interface Builder实现ModalView-Sample案例	277
9.3.4	代码实现ModalViewSample案例	281
9.4	Popover（浮动层）视图	284
9.4.1	Popover相关API	284
9.4.2	PopoverViewSample案例	285
9.5	iOS 9分屏多任务	289
9.5.1	Slide Over多任务	289
9.5.2	分屏视图多任务	290
9.5.3	画中画多任务	293

9.6	iPad分屏多任务适配开发	293
9.6.1	分屏多任务前提条件	293
9.6.2	分屏多任务适配	294
9.7	小结	295
第二部分 设计与架构篇		
第10章	iOS常用设计模式	298
10.1	单例模式	298
10.1.1	问题提出	298
10.1.2	Objective-C实现原理	298
10.1.3	Swift实现原理	299
10.1.4	应用案例	300
10.2	委托模式	301
10.2.1	问题提出	301
10.2.2	实现原理	303
10.2.3	应用案例	306
10.3	观察者模式	309
10.3.1	问题提出	309
10.3.2	实现原理	310
10.3.3	通知机制和KVO机制	311
10.4	再谈MVC模式	317
10.4.1	MVC模式概述	317
10.4.2	Cocoa Touch中的MVC模式	318
10.5	小结	320
第11章	iOS分层架构设计	321
11.1	低耦合企业级系统架构设计	321
11.2	iOS分层架构设计	322
11.2.1	工程与工作空间	322
11.2.2	静态链接库	323
11.2.3	框架	323
11.2.4	16种分层模式	324
11.3	实例：MyNotes应用	325
11.4	基于同一工程的分层	327
11.4.1	采用纯Swift语言实现：PSSS模式	327
11.4.2	采用纯Objective-C语言实现：POOO模式	330
11.4.3	采用Swift调用Objective-C的混合搭配实现：PSOO模式	334
11.5	基于同一工作空间静态链接库实现的WLSOO模式	335
11.5.1	使用工作空间	335
11.5.2	添加数据持久层工程	336
11.5.3	建立业务逻辑层与数据持久层依赖关系	337
11.5.4	建立表示层与业务逻辑层依赖关系	339
11.6	基于同一工作空间框架实现的WFSSS模式	339
11.6.1	创建框架工程	340
11.6.2	建立依赖关系	341
11.6.3	代码重构	342
11.7	基于同一工作空间框架实现的WFOOO模式	344
11.7.1	设置Public头文件	344
11.7.2	设置保护伞头文件	345
11.7.3	代码重构	345
11.8	基于同一工作空间框架实现的WFSOO模式	346

11.8.1	设置Public头文件	346
11.8.2	设置保护伞头文件	346
11.8.3	建立表示层与业务逻辑层依赖关系	346
11.8.4	代码重构	346
11.9	基于同一工作空间框架实现的WFOSS模式	346
11.9.1	设置Public头文件	346
11.9.2	设置Swift代码嵌入应用	347
11.9.3	代码重构	347
11.10	基于同一个工程不同目标框架实现的TFSSO模式	348
11.10.1	使用目标	348
11.10.2	添加框架目标	348
11.11	小结	349
第12章	数据持久化	350
12.1	概述	350
12.1.1	沙箱目录	350
12.1.2	持久化方式	351
12.2	属性列表	351
12.3	对象归档	357
12.4	使用SQLite数据库	361
12.4.1	SQLite数据类型	361
12.4.2	配置Objective-C框架工程环境	362
12.4.3	配置Swift框架工程环境	362
12.4.4	创建数据库	364
12.4.5	查询数据	365
12.4.6	修改数据	368
12.5	Core Data	370
12.5.1	ORM	370
12.5.2	Core Data栈	371
12.5.3	建模和生成实体	374
12.5.4	采用Core Data分层架构设计	380
12.5.5	查询数据	382
12.5.6	修改数据	384
12.6	小结	385
第三部分	进阶篇	
第13章	应用程序设置	388
13.1	概述	388
13.1.1	设置	388
13.1.2	配置	389
13.2	应用程序设置包	390
13.3	设置项目种类	392
13.3.1	文本字段	395
13.3.2	开关	398
13.3.3	滑块	399
13.3.4	值列表	400
13.3.5	子界面	401
13.4	读取设置	403
13.5	小结	405
第14章	本地化	406
14.1	概述	406

14.1.1	需要本地化的内容	406	
14.1.2	本地化目录结构	409	
14.2	文本信息本地化	410	
14.2.1	系统按钮和信息本地化	410	
14.2.2	应用名称本地化	411	
14.2.3	程序代码输出的静态文本本地化	413	
14.2.4	使用genstring工具	414	
14.3	故事板和XIB文件本地化	415	
14.3.1	使用Base Internationalization技术	415	
14.3.2	Auto Layout与本地化	417	
14.4	资源文件本地化	419	
14.4.1	图片资源文件本地化	419	
14.4.2	声音资源文件本地化	420	
14.5	小结	421	
第15章	iOS 9中访问通讯录	422	
15.1	通讯录的安全访问设置	422	
15.2	使用Contacts框架读取联系人信息	423	
15.2.1	查询联系人	423	
15.2.2	读取单值属性	426	
15.2.3	读取多值属性	428	
15.2.4	读取图片属性	430	
15.3	使用Contacts框架写入联系人信息	430	
15.3.1	创建联系人	431	
15.3.2	修改联系人	433	
15.3.3	删除联系人	434	
15.4	使用系统提供界面	435	
15.4.1	选择联系人	435	
15.4.2	显示和修改联系人	438	
15.4.3	创建联系人	441	
15.5	小结	444	
第16章	数据交换格式	445	
16.1	XML数据交换格式	446	
16.1.1	XML文档结构	446	
16.1.2	解析XML文档	447	
16.2	实例：MyNotes应用XML	448	
16.2.1	使用NSXML解析	450	
16.2.2	使用TBXML解析	453	
16.3	JSON数据交换格式	459	
16.3.1	JSON文档结构	459	
16.3.2	JSON数据编码/解码	460	
16.4	实例：MyNotes应用JSON解码	461	
16.5	小结	462	
第17章	REST Web Service	463	
17.1	概述	463	
17.2	使用NSURLSession	464	
17.2.1	NSURLSession API	464	
17.2.2	简单会话实现GET请求	465	
17.2.3	默认会话实现GET请求	468	
17.2.4	实现POST请求	469	

17.2.5	下载数据	470
17.3	实例：使用NSURLSession重构MyNotes案例	473
17.3.1	插入方法调用	473
17.3.2	修改方法调用	474
17.3.3	删除方法调用	475
17.4	使用AFNetworking框架	477
17.4.1	比较ASIHTTPRequest、AFNetworking和MKNetworkKit	477
17.4.2	安装和配置AFNetworking框架	478
17.4.3	实现GET请求	479
17.4.4	实现POST请求	480
17.4.5	下载数据	481
17.4.6	上传数据	482
17.5	使用为Swift设计的网络框架：Alamofire	484
17.5.1	安装和配置Alamofire框架	484
17.5.2	实现GET请求	485
17.5.3	实现POST请求	486
17.5.4	下载数据	487
17.5.5	上传数据	488
17.6	反馈网络信息改善用户体验	489
17.6.1	使用下拉刷新控件改善用户体验	489
17.6.2	使用活动指示器控件	492
17.6.3	使用网络活动指示器	494
17.7	小结	494
第18章	定位服务	495
18.1	定位服务概述	495
18.1.1	定位服务编程	496
18.1.2	测试定位服务	499
18.2	管理定位服务	503
18.2.1	应用启动与停止下的定位服务管理	504
18.2.2	视图切换下的定位服务管理	504
18.2.3	应用前后台切换下的定位服务管理	505
18.2.4	设置自动暂停位置服务	507
18.2.5	iOS 9后台位置服务管理	507
18.3	地理信息编码与反编码	508
18.3.1	地理信息反编码	508
18.3.2	实例：地理信息反编码	509
18.3.3	地理信息编码查询	510
18.3.4	实例：地理信息编码查询	510
18.4	微定位技术iBeacon	512
18.4.1	微定位与地理围栏	512
18.4.2	iBeacon技术概述	512
18.4.3	实例：使用iBeacon技术实现微定位	513
18.5	小结	519
第19章	苹果地图应用	520
19.1	使用iOS苹果地图	520
19.1.1	显示地图	520
19.1.2	显示3D地图	524
19.2	添加标注	525
19.2.1	实现查询	526

19.2.2	在地图上添加标注	528
19.3	跟踪用户位置变化	529
19.4	使用程序外地图	530
19.5	小结	533
第四部分 测试、调试和优化篇		
第20章	找出程序中的bug——调试	536
20.1	Xcode调试工具	536
20.1.1	定位编译错误	536
20.1.2	查看和显示日志	537
20.1.3	设置和查看断点	539
20.1.4	调试工具	544
20.1.5	输出窗口	545
20.1.6	变量查看窗口	546
20.1.7	查看线程	547
20.2	LLDB调试工具	548
20.2.1	断点命令	548
20.2.2	观察点命令	550
20.2.3	查看变量和计算表达式命令	552
20.3	日志与断言输出	554
20.3.1	使用NSLog函数	554
20.3.2	使用断言	555
20.4	异常栈报告分析	556
20.4.1	跟踪异常栈	556
20.4.2	分析栈报告	558
20.5	在iOS设备上调试	559
20.5.1	Xcode设置	559
20.5.2	设备设置	560
20.6	Xcode设备管理工具	562
20.6.1	查看设备上的应用程序	562
20.6.2	设备日志	564
20.7	小结	564
第21章	iOS测试驱动与单元测试	565
21.1	测试驱动的软件开发概述	565
21.1.1	测试驱动的软件开发流程	565
21.1.2	测试驱动的软件开发案例	566
21.1.3	iOS单元测试框架	567
21.2	使用XCTest测试框架	568
21.2.1	添加XCTest到工程	568
21.2.2	Swift版本中设置编译目标成员	570
21.2.3	编写XCTest测试方法	570
21.2.4	运行测试用例目标	573
21.2.5	分析测试报告	575
21.3	异步单元测试	576
21.4	性能测试	580
21.4.1	Swift版本中配置测试环境	580
21.4.2	测试用例代码	580
21.4.3	分析测试结果	581
21.5	iOS单元测试最佳实践	583
21.5.1	配置测试数据持久层	583

21.5.2	编写数据持久层测试用例	584
21.5.3	运行测试数据持久层测试用例	587
21.5.4	配置测试业务逻辑层	588
21.5.5	编写业务逻辑层测试用例	588
21.6	小结	589
第22章	iOS应用UI测试	590
22.1	UI测试概述	590
22.2	添加UI测试到工程	590
22.2.1	创建工程时添加UI测试框架	590
22.2.2	在现有工程中添加UI测试用例目标	591
22.3	录制脚本	593
22.3.1	录制之前的准备	593
22.3.2	录制过程	593
22.3.3	修改录制脚本	595
22.4	访问UI元素	595
22.4.1	UI元素层次结构树	595
22.4.2	UI测试中相关API	597
22.5	表示层测试最佳实践	598
22.5.1	配置UI测试用例目标	598
22.5.2	编写UI测试用例	599
22.6	小结	602
第23章	让你的程序“飞”起来——性能优化	603
23.1	内存优化	603
23.1.1	内存管理	603
23.1.2	使用Analyze工具检查内存泄漏	603
23.1.3	使用Instruments工具检查内存泄漏	608
23.1.4	使用Instruments工具检查僵尸对象	612
23.1.5	autorelease的使用问题	615
23.1.6	响应内存警告	616
23.2	优化资源文件	617
23.2.1	图片文件优化	618
23.2.2	音频文件优化	619
23.3	延迟加载	620
23.3.1	资源文件的延迟加载	620
23.3.2	故事板文件的延迟加载	624
23.3.3	XIB文件的延迟加载	626
23.4	数据持久化的优化	628
23.4.1	使用文件	628
23.4.2	使用SQLite数据库	631
23.4.3	使用Core Data	632
23.5	可重用对象的使用	634
23.5.1	表视图中的可重用对象	635
23.5.2	集合视图中的可重用对象	636
23.5.3	地图视图中的可重用对象	638
23.6	并发处理与多核CPU	638
23.6.1	主线程阻塞问题	638
23.6.2	选择NSThread还是GCD	639
23.7	小结	640
第五部分	实战篇	

第24章	管理好你的程序代码——代码版本控制	642
24.1	概述	642
24.1.1	版本控制历史	642
24.1.2	基本概念	643
24.2	Git代码版本控制	643
24.2.1	服务器搭建	643
24.2.2	Gitolite服务器管理	645
24.2.3	Git常用命令	647
24.2.4	Git分支	649
24.2.5	Git协同开发	653
24.2.6	Xcode中Git的配置与使用	656
24.3	GitHub代码托管服务	663
24.3.1	创建和配置GitHub账号	663
24.3.2	创建代码库	666
24.3.3	删除代码库	668
24.3.4	派生代码库	669
24.3.5	使用GitHub协同开发	671
24.3.6	管理组织	677
24.4	小结	680
第25章	项目依赖管理	681
25.1	使用CocoaPods工具管理依赖	681
25.1.1	安装CocoaPods	681
25.1.2	搜索库	682
25.1.3	项目与第三方库搭配形式	683
25.1.4	示例：静态链接库形式管理依赖	684
25.1.5	示例：框架形式管理依赖	686
25.2	使用Carthage工具管理依赖	687
25.2.1	安装Carthage	688
25.2.2	项目与第三方库搭配形式	688
25.2.3	Cartfile文件	688
25.2.4	示例：重构MyNotes依赖关系	689
25.3	小结	691
第26章	重构MyNotes应用——iOS网络通信中的设计模式与架构设计	692
26.1	移动网络通信应用的分层架构设计	692
26.2	Objective-C版本：在数据持久层中添加和配置AFNetworking	693
26.2.1	用CocoaPods工具管理依赖	693
26.2.2	测试依赖	693
26.3	Swift版本：在数据持久层中添加和配置Alamofire	694
26.3.1	用Carthage工具管理依赖	694
26.3.2	测试依赖	695
26.4	基于委托模式实现	695
26.4.1	网络通信与委托模式	696
26.4.2	在异步网络通信中使用委托模式实现分层架构设计	696
26.4.3	类图	697
26.4.4	时序图	699
26.4.5	数据持久层重构（Objective-C版本）	702
26.4.6	数据持久层重构（Swift版本）	705
26.4.7	业务逻辑层的代码实现	707
26.4.8	表示层的代码实现	710

26.5	基于观察者模式的通知机制实现	716
26.5.1	观察者模式的通知机制回顾	716
26.5.2	异步网络通信中通知机制的分层架构设计	716
26.5.3	类图	717
26.5.4	时序图	719
26.5.5	数据持久层重构 (Objective-C版本)	722
26.5.6	数据持久层重构 (Swift版本)	723
26.5.7	业务逻辑层的代码实现	725
26.5.8	表示层的代码实现	726
26.6	小结	731
第27章	把你的应用放到App Store上	732
27.1	收官	732
27.1.1	在Xcode中添加图标	732
27.1.2	在Xcode中添加启动界面	734
27.1.3	调整Identity和Deployment Info属性	738
27.2	为发布进行编译	739
27.2.1	创建开发者证书	739
27.2.2	创建App ID	744
27.2.3	创建描述文件	745
27.2.4	发布编译	748
27.3	发布上架	750
27.3.1	创建应用	750
27.3.2	应用定价	753
27.3.3	基本信息输入	754
27.3.4	上传应用	757
27.3.5	提交审核	759
27.4	常见审核不通过的原因	761
27.4.1	功能问题	761
27.4.2	用户界面问题	761
27.4.3	商业问题	761
27.4.4	不当内容	761
27.4.5	其他问题	762
27.5	小结	762
第28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	763
28.1	应用分析与设计	763
28.1.1	应用概述	763
28.1.2	需求分析	763
28.1.3	原型设计	764
28.1.4	数据库设计	765
28.1.5	架构设计	766
28.2	iOS敏捷开发	766
28.2.1	敏捷开发宣言	766
28.2.2	iOS适合敏捷开发?	767
28.2.3	iOS敏捷开发最佳实践	767
28.3	任务1: 创建应用工作空间和工程	769
28.3.1	迭代1.1: 创建工作空间	769
28.3.2	迭代1.2: 发布到GitHub	769
28.4	任务2: 信息系统层与持久层开发	770
28.4.1	迭代2.1: 编写数据库DDL脚本	770

28.4.2	迭代2.2：插入初始数据到数据库	770
28.4.3	迭代2.3：数据库版本控制	771
28.4.4	迭代2.4：配置持久层工程PersistenceLayer	771
28.4.5	迭代2.5：编写实体类	772
28.4.6	迭代2.6：编写DAO类单元测试用例类	773
28.4.7	迭代2.7：编写DAO类	775
28.4.8	迭代2.8：数据库帮助类DBHelper	779
28.4.9	迭代2.9：配置及运行持久层测试用例目标	781
28.4.10	迭代2.10：发布到GitHub	782
28.5	任务3：业务逻辑层开发	782
28.5.1	迭代3.1：比赛项目业务逻辑类XCTest单元测试	783
28.5.2	迭代3.2：编写比赛项目业务逻辑类	784
28.5.3	迭代3.3：比赛日程业务逻辑类XCTest单元测试	785
28.5.4	迭代3.4：编写比赛日程业务逻辑类	786
28.5.5	迭代3.5：发布到GitHub	787
28.6	任务4：表示层开发	788
28.6.1	迭代4.1：使用资源目录管理图片和图标资源	788
28.6.2	迭代4.2：根据原型设计初步设计故事板	789
28.6.3	迭代4.3：“首页”模块	790
28.6.4	迭代4.4：“比赛项目”模块	791
28.6.5	迭代4.5：“比赛日程”模块	795
28.6.6	迭代4.6：“倒计时”模块表示层	797
28.6.7	迭代4.7：“关于我们”模块表示层	800
28.6.8	迭代4.8：发布到GitHub	800
28.7	任务5：收工	800
28.7.1	迭代5.1：添加图标	800
28.7.2	迭代5.2：设计和添加启动界面	801
28.7.3	迭代5.3：性能测试与改善	801
28.7.4	迭代5.4：发布到GitHub上	802
28.7.5	迭代5.5：在App Store上发布应用	802
28.8	小结	802

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:www.tushu000.com