



Objective-C和Swift双语讲解

- 畅销书全新升级，全面涵盖iOS 8新特性、新变化
- 数百个项目案例+两个真实项目开发全过程
- 涵盖iOS平台架构设计、测试驱动开发、性能优化、版本控制和程序调试等



iOS 开发指南

从零基础到App Store上架

(第3版)

关东升 著



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

iOS 开发指南

从零基础到App Store上架

(第3版)

关东升 著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

iOS开发指南：从零基础到App Store上架 / 关东升
著. — 3版. — 北京：人民邮电出版社，2015.4
(图灵原创)
ISBN 978-7-115-38776-9

I. ①i… II. ①关… III. ①移动终端—应用程序—
程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2015)第055676号

内 容 提 要

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。

本书适合 iOS 开发人员阅读。

-
- ◆ 著 关东升
责任编辑 王军花
责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京艺辉印刷有限公司印刷
 - ◆ 开本：880×1230 1/16
印张：42
字数：1338千字 2015年4月第3版
印数：26 001 - 31 000册 2015年4月北京第1次印刷
-

定价：99.00元

读者服务热线：(010)51095186转600 印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京崇工商广字第 0021 号

前 言

北京时间 2014 年 6 月 2 日凌晨 1 点，苹果公司发布了最新的 iOS 8 系统。此次，最令人震撼的消息是苹果的新语言 Swift。而 iOS 开发者和感兴趣者立即开始研究和学习 Swift 语言，我们智捷课堂团队在 2014 年 6 月 9 日推出一套完整的 Swift 学习视频，接着又与图灵教育合作，推出国内第一本 Swift 图书——《Swift 开发指南》，并且在 2014 年 8 月 1 日推出了《Swift 开发指南（修订版）》。在这个大背景下，我们原来编写的《iOS 开发指南：从零基础到 App Store 上架》得到了广大读者的认可，很多读者希望我们将其升级为 iOS 8 版本，并且能够使用 Swift 和 Objective-C 两种语言进行讲解，并比较它们的不同之处。几个月过去了，我们终于在 2015 年元旦之前将书稿提交给出版社。

内容和组织结构

本书是我们团队编写的 iOS 系列丛书中的一本，目的是使有 Swift 或 Objective-C 基础的程序员通过学习本书，从零基础学习如何在 App Store 上发布一款应用。全书共 4 部分。

第一部分为基础篇，共 14 章内容，介绍了 iOS 的一些基础知识。

第 1 章介绍了 iOS 的开发背景以及本书约定。

第 2 章使用故事板技术创建了 HelloWorld，同时讨论了 iOS 工程模板、应用的运行机制和生命周期、视图的生命周期等，最后介绍了如何使用 API 帮助文档和官方案例。

第 3 章首先学习了视图和控件之间的关系以及应用界面的构建层次，然后介绍了标签、按钮、文本框和导航栏等基本控件。

第 4 章介绍了 iOS 8 多分辨率屏幕适配方法，其中涉及的技术主要是 Auto Layout 和 Size Class。此外，还介绍了 iOS 屏幕的多样性、iOS 屏幕布局相关的内容。

第 5 章首先向大家介绍了数据源协议和委托协议，然后介绍了高级视图：选择器和集合视图。

第 6 章探讨了表视图的组成、表视图类的构成和表视图的分类，使我们对表视图有了一个整体上的认识。接下来，介绍了如何实现简单表视图和分节表视图，以及表视图中索引、搜索栏和分组的使用，然后学习了如何对表视图单元格进行删除、插入、移动等操作，最后介绍了表视图 UI 设计模式方面的内容。

第 7 章讨论了如何判断应用是否需要一个导航功能，并且知道在什么情况下选择平铺导航、标签导航、树形结构导航，或者同时综合使用这 3 种导航模式。

第 8 章讨论了 iOS 开发中 4 种常用的设计模式，分别为单例模式、委托模式、观察者模式和 MVC 模式。在介绍每种设计模式时，我们按照问题提出、实现原理、应用案例的结构介绍了其适用情况、实现原理及其用法。

第 9 章首先介绍了 iPhone 和 iPad 设备使用场景上的差异，然后深入介绍了 iPad 专用 API，包括 UIPopoverController 控制器和 UISplitViewController 控制器。

第 10 章介绍了 iOS 平台的分层架构设计技术，归纳了 10 种分层模式，并且重点实现了 8 种模式，每一种模式都有其应用场景。

第 11 章讨论了 iOS 本地数据持久化的问题。首先分析了数据存取的几种方式以及每种数据存取方式适合什么样的场景，然后分别举例介绍了每种存取方式的实现。

第 12 章介绍了设置和配置的概念，然后通过对二者差异的探讨，介绍了什么样的项目适合放在设置里，什

么样的项目适合放在配置里。

第 13 章首先介绍了国际化的概念，接下来详细阐述了文本信息、xib 及故事板、资源文件的国际化。

第 14 章首先介绍了访问通讯录所需要的框架，然后介绍了使用 AddressBook 框架如何读取联系人的信息，具体包括联系人记录、单值多值属性、图片属性的读取；接着介绍了如何使用该框架将联系人信息写入数据库，具体包括联系人的创建、修改和删除；最后介绍了如何使用 AddressBook 高级 API 实现选择联系人、显示和修改联系人以及创建联系人的操作。

第二部分为网络篇，共两章，介绍了 iOS 网络开发的相关知识。

第 15 章介绍了数据交换格式，其中 XML 和 JSON 是主要的方式。这里重点介绍了 Web Service 的访问以及 MKNetworkKit 框架。

第 16 章讨论了 iOS 中的定位服务技术，包括地理信息编码和反编码查询。之后介绍了 iOS 苹果地图的使用，包括显示地图、在地图上添加标注以及跟踪用户位置的变化等。最后，介绍了程序外地图的使用，如何调用 iOS 苹果地图和谷歌 Web 地图。

第三部分为进阶篇，共 6 章，介绍了 iOS 高级内容和商业思考等，相关内容如下所示。

第 17 章介绍了 iOS 中的商业模式，其中的收费策略值得广大读者借鉴。此外，我们还介绍了植入广告和应用内购买的 API，其中植入广告包括苹果自己的 iAd 和谷歌的 AdMob 广告。

第 18 章首先介绍了调试工具、异常堆栈报告分析，接下来讲解了如何在真机上调试应用，最后分析了 Xcode 设备管理工具的用法。

第 19 章首先介绍了测试驱动的 iOS 开发，接着介绍了单元测试框架 XCTest、异步单元测试和性能测试，最后讨论了 iOS 单元测试最佳实践。

第 20 章介绍了 iOS 中的性能优化方法，其中包括内存优化、资源文件优化、延迟加载、持久化优化、使用可重用对象、并发处理与多核 CPU 等。

第 21 章介绍了如何使用 Git 进行代码版本控制，其中包括 Git 服务器的搭建、Git 常用命令和协同开发。此外，还介绍了在 Xcode 中如何配置和使用 Git 工具。

第 22 章探讨了如何在 App Store 上发布应用，介绍了应用的发布流程以及应用审核不通过的一些常见原因。

第四部分为实战篇，共两章，从无到有地介绍了两个真实的 iOS 应用：MyNotes 应用和 2016 里约热内卢奥运会应用。

第 23 章通过重构 MyNotes 应用，把 MyNotes 应用的数据由原来的本地存储变成云存储。在这个过程中，我们介绍了移动网络通信应用中分层架构设计的必要性和重要性。我们重点介绍了基于委托模式和观察者模式通知机制实现的分层架构设计。

第 24 章介绍了完整的 iOS 应用分析设计、编程、测试和发布过程，其中采用了敏捷开发方法。此外，该项目采用分层架构设计，这对于学习 iOS 架构是非常重要的。

书中并没有包括多媒体等知识，我们会在另外一本 iOS 开发书中介绍，具体进展请读者关注智捷 iOS 课堂官方网站 <http://www.51work6.com>。

本书服务网址

为了更好地为广大读者提供服务，我们专门为本书建立了一个服务网址 www.51work6.com/ios1.php，大家可以查看相关出版进度，并对书中内容发表评论，提出宝贵意见。

源代码

书中包括了 100 多个完整的案例项目源代码，大家可以到本书网站 www.51work6.com/ios1.php 下载或者到图灵社区本书主页（www.ituring.com.cn/book/1629）免费注册下载。

勘误与支持

我们在网站 www.51work6.com/ios1.php 中建立了一个勘误专区，可以及时地把书中的问题、失误和纠正反馈给广大读者。如果你发现了任何问题，均可以在网上留言，也可以发送电子邮件到 eorient@sina.com，我们会在第一时间回复你。此外，你也可以通过新浪微博与我们联系，我的微博为@tony_关东升。

致谢

在此感谢图灵的王军花责编给我们提供的宝贵意见，感谢智捷 iOS 课堂团队的李玉超和贾云龙参与内容讨论和审核，感谢赵大羽老师手绘了书中全部草图，并从专业的角度修改书中图片，力求更加真实完美地奉献给广大读者。此外，还要感谢我的家人容忍我的忙碌，以及对我的关心和照顾，使我能抽出这么多时间，投入全部精力专心编写此书。

由于时间仓促，书中难免存在不妥之处，请读者原谅。

关东升
2014年12月于北京

目 录

第一部分 基础篇

第 1 章 开篇综述	2	第 3 章 UIView 与视图	37
1.1 iOS 概述	2	3.1 视图“始祖”——UIView	37
1.1.1 iOS 介绍	2	3.1.1 UIView“家族”	37
1.1.2 iOS 8 的新特性	2	3.1.2 应用界面的构建层次	39
1.2 开发环境及开发工具	3	3.1.3 视图分类	40
1.3 本书中的约定	4	3.2 标签与按钮	40
1.3.1 案例代码约定	4	3.2.1 标签控件	41
1.3.2 图示的约定	5	3.2.2 按钮控件	42
1.3.3 方法命名约定	6	3.3 动作与输出口	44
1.3.4 构造器命名约定	7	3.4 TextField 和 TextView	48
第 2 章 第一个 iOS 应用程序	9	3.4.1 TextField 控件	48
2.1 创建 HelloWorld 工程	9	3.4.2 TextView 控件	49
2.1.1 创建工程	9	3.4.3 键盘的打开和关闭	50
2.1.2 Xcode 中的 iOS 工程模板	14	3.4.4 关闭和打开键盘的通知	50
2.1.3 应用剖析	15	3.4.5 键盘的种类	51
2.2 HelloWorld 工程中的故事板文件	17	3.5 开关控件、滑块控件和分段控件	52
2.2.1 故事板的导航特点	17	3.5.1 开关控件	53
2.2.2 故事板中的 Scene 和 Segue	19	3.5.2 滑块控件	53
2.3 应用生命周期	19	3.5.3 分段控件	54
2.3.1 非运行状态——应用启动场景	21	3.6 网页视图 WebView	54
2.3.2 点击 Home 键——应用退出场景	22	3.7 等待相关的控件与进度条	57
2.3.3 挂起重新运行场景	24	3.7.1 活动指示器 ActivityIndicatorView	58
2.3.4 内存清除——应用终止场景	25	3.7.2 进度条 ProgressView	59
2.4 视图生命周期	25	3.8 警告框和操作表	60
2.5 设置产品属性	27	3.8.1 警告框	60
2.5.1 Xcode 中的 PROJECT 和 TARGET	27	3.8.2 操作表 ActionSheet	62
2.5.2 设置常用的产品属性	29	3.8.3 使用 iOS 8 UIAlertController 实现	63
2.6 iOS API 简介	30	3.9 工具栏和导航栏	64
2.6.1 API 概述	31	3.9.1 工具栏	64
2.6.2 如何使用 API 帮助	33	3.9.2 导航栏	67
2.7 小结	36	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	6.3.1 添加索引	137
4.2 iOS 屏幕布局	74	6.3.2 分组	140
4.2.1 iOS 中的“栏”	74	6.3.3 静态表与控制界面布局	141
4.2.2 传统布局	75	6.4 修改单元格	144
4.3 Auto Layout 布局	77	6.4.1 删除和插入单元格	144
4.3.1 在 Interface Builder 中管理 Auto Layout 约束	77	6.4.2 移动单元格	151
4.3.2 实例: Auto Layout 布局	78	6.5 表视图 UI 设计模式	153
4.4 Size Class 与 iOS 8 多屏幕适配	83	6.5.1 分页模式	153
4.4.1 在 Interface Builder 中使用 Size Class	83	6.5.2 下拉刷新模式	154
4.4.2 Size Class 的九宫格	84	6.5.3 下拉刷新控件	154
4.4.3 实例: 使用 Size Class	85	6.6 小结	157
4.5 在屏幕滚动视图中使用 Auto Layout 和 Size Class	90	第 7 章 视图控制器与导航模式	158
4.5.1 屏幕滚动视图的重要属性	90	7.1 概述	158
4.5.2 屏幕滚动视图实例	92	7.1.1 视图控制器的种类	158
4.6 使用资源目录 (asset catalog) 管理图片	97	7.1.2 导航模式	158
4.7 小结	100	7.1.3 模态视图	160
第 5 章 委托协议、数据源协议与高级视图	101	7.2 平铺导航	166
5.1 视图中的委托协议和数据源协议	101	7.2.1 应用场景	166
5.2 选择器	101	7.2.2 基于分屏导航的实现	168
5.2.1 日期选择器	101	7.2.3 基于电子书导航的实现	173
5.2.2 普通选择器	103	7.3 标签导航	178
5.2.3 数据源协议与委托协议	106	7.3.1 应用场景	178
5.3 集合视图	108	7.3.2 实现	179
5.3.1 集合视图介绍	108	7.4 树形结构导航	182
5.3.2 实例: 奥运会比赛项目	109	7.4.1 应用场景	182
5.3.3 添加集合视图控制器	110	7.4.2 实现	183
5.3.4 添加集合视图单元格	113	7.5 组合使用导航模式	190
5.3.5 数据源协议与委托协议	116	7.5.1 应用场景	190
5.4 小结	117	7.5.2 实现	191
第 6 章 表视图	118	7.6 小结	198
6.1 概述	118	第 8 章 iOS 常用设计模式	199
6.1.1 表视图的组成	118	8.1 单例模式	199
6.1.2 表视图的相关类	119	8.1.1 问题提出	199
6.1.3 表视图分类	119	8.1.2 Objective-C 实现原理	199
6.1.4 单元格的组成和样式	121	8.1.3 Swift 实现原理	200
6.1.5 数据源协议与委托协议	123	8.1.4 应用案例	201
6.2 简单表视图	123	8.2 委托模式	202
6.2.1 创建简单表视图	123	8.2.1 问题提出	202
6.2.2 自定义单元格	128	8.2.2 实现原理	204
6.2.3 添加搜索栏	131	8.2.3 应用案例	207
6.3 分节表视图	137	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	10.7.2 建立表示层与业务逻辑层的依赖关系	256
8.4.1 MVC 模式概述	218	10.7.3 代码重构	256
8.4.2 Cocoa Touch 中的 MVC 模式	219	10.8 基于同一工作空间框架实现的 WFSOO 模式	256
8.5 小结	221	10.8.1 设置 Public 头文件	256
第 9 章 iPhone 与 iPad 应用开发的差异	222	10.8.2 建立表示层与业务逻辑层的依赖关系	257
9.1 概述	222	10.8.3 代码重构	257
9.1.1 应用场景差异	222	10.9 基于同一工作空间框架实现的 WFOSS 模式	258
9.1.2 设计和开发需注意的问题	222	10.9.1 设置 Public 头文件	258
9.2 iPad 专用 API	225	10.9.2 代码重构	258
9.2.1 UIPopoverController 控制器	225	10.10 小结	258
9.2.2 UISplitViewController 控制器	229	第 11 章 数据持久化	259
9.3 小结	234	11.1 概述	259
第 10 章 iOS 分层架构设计	235	11.1.1 沙箱目录	259
10.1 低耦合企业级系统架构设计	235	11.1.2 持久化方式	260
10.2 iOS 分层架构设计	236	11.2 属性列表	260
10.2.1 工程与工作空间	236	11.3 对象归档	264
10.2.2 静态链接库	237	11.4 使用 SQLite 数据库	268
10.2.3 框架	237	11.4.1 SQLite 数据类型	268
10.2.4 10 种分层模式	238	11.4.2 创建数据库	269
10.3 实例: MyNotes 应用	238	11.4.3 查询数据	271
10.4 基于同一工程的分层	240	11.4.4 修改数据	273
10.4.1 采用纯 Swift 语言实现——PSSS 模式	240	11.5 Core Data	276
10.4.2 采用纯 Objective-C 语言实现——POOO 模式	243	11.5.1 ORM	276
10.4.3 采用 Swift 与 Objective-C 混合搭配实现——PSOO 模式	247	11.5.2 Core Data 堆栈	276
10.5 基于同一工作空间静态链接库实现的 WLSOO 模式	247	11.5.3 建模和生成实体	280
10.5.1 使用工作空间	247	11.5.4 采用 Core Data 分层架构设计	283
10.5.2 添加数据持久层工程	248	11.5.5 查询数据	285
10.5.3 建立业务逻辑层与数据持久层的依赖关系	249	11.5.6 修改数据	286
10.5.4 建立表示层与业务逻辑层的依赖关系	251	11.6 小结	288
10.6 基于同一工作空间框架实现的 WFSSS 模式	251	第 12 章 应用程序设置	289
10.6.1 创建框架工程	251	12.1 概述	289
10.6.2 依赖关系建立	253	12.1.1 设置	289
10.6.3 代码重构	254	12.1.2 配置	290
10.7 基于同一工作空间框架实现的 WFOOO 模式	255	12.2 应用程序设置包	291
10.7.1 设置 Public 头文件	255	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	15.2.4 JSON 数据解码	356
12.4 读取设置	303	15.3 REST Web Service	357
12.5 小结	304	15.3.1 HTTP 和 HTTPS 协议	358
第 13 章 国际化	305	15.3.2 同步 GET 请求方法	358
13.1 概述	305	15.3.3 异步 GET 请求方法	361
13.1.1 需要国际化的内容	305	15.3.4 POST 请求方法	362
13.1.2 国际化目录结构	307	15.3.5 调用 REST Web Service 的插入、 修改和删除方法	363
13.2 文本信息国际化	308	15.4 使用轻量级网络请求框架 MKNetworkKit	368
13.2.1 系统按钮和信息国际化	308	15.4.1 ASIHTTPRequest、AFNetworking 和 MKNetworkKit 比较	368
13.2.2 应用名称国际化	309	15.4.2 安装和配置 MKNetworkKit 框架	369
13.2.3 程序代码输出的静态文本国际化	311	15.4.3 网络请求	370
13.2.4 使用 genstring 工具	312	15.4.4 下载数据	371
13.3 xib 和故事板文件国际化	313	15.4.5 上传数据	373
13.3.1 使用 Base 国际化技术	313	15.5 反馈网络信息改善用户体验	375
13.3.2 Auto Layout 与国际化	315	15.5.1 使用下拉刷新控件改善用户体验	375
13.4 资源文件国际化	317	15.5.2 使用活动指示器控件	378
13.4.1 图片资源文件国际化	317	15.5.3 使用网络活动指示器	380
13.4.2 声音资源文件国际化	318	15.6 小结	380
13.5 小结	318	第 16 章 定位服务与地图应用	381
第 14 章 访问联系人	319	16.1 定位服务	381
14.1 概述	319	16.1.1 定位服务编程	382
14.2 读取联系人信息	320	16.1.2 地理信息反编码	386
14.2.1 查询联系人记录	321	16.1.3 地理信息编码查询	388
14.2.2 读取单值属性	324	16.1.4 关于定位服务的测试	390
14.2.3 读取多值属性	326	16.2 使用 iOS 苹果地图	393
14.2.4 读取图片属性	328	16.2.1 显示地图	393
14.3 写入联系人信息	329	16.2.2 添加标注	395
14.3.1 创建联系人	330	16.2.3 跟踪用户位置变化	398
14.3.2 修改联系人	332	16.3 使用程序外地图	399
14.3.3 删除联系人	333	16.3.1 调用 iOS 苹果地图	400
14.4 高级 API	335	16.3.2 调用谷歌 Web 地图	402
14.4.1 选择联系人	335	16.4 小结	404
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

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	19.1.2	测试驱动的软件开发案例	466
17.2	使用苹果 iAd 广告	408	19.1.3	iOS 单元测试框架	467
17.2.1	横幅广告	408	19.2	使用 XCTest 测试框架	468
17.2.2	插页广告	411	19.2.1	添加 XCTest 到工程	468
17.2.3	查看你的收入	417	19.2.2	编写 XCTest 测试方法	469
17.3	使用谷歌 AdMob 广告	419	19.2.3	运行测试程序	472
17.3.1	注册 AdMob 账号	419	19.2.4	分析测试报告	473
17.3.2	管理 AdMob 广告	419	19.3	异步单元测试	475
17.3.3	AdMob 广告类型	422	19.4	性能测试	478
17.3.4	下载 AdMob Ads SDK	423	19.4.1	配置测试环境	478
17.3.5	配置 AdMob 开发环境	424	19.4.2	测试代码	479
17.3.6	AdMob 横幅广告	426	19.4.3	分析测试结果	480
17.3.7	添加 AdMob 插页广告	428	19.5	iOS 单元测试最佳实践	481
17.3.8	为广告提交用户和位置信息	431	19.5.1	测试数据持久层	481
17.3.9	搜索广告	431	19.5.2	测试业务逻辑层	485
17.4	小结	433	19.6	小结	486
第 18 章 找出程序中的 bug——调试			第 20 章 让你的程序“飞”起来——性能优化		
18.1	Xcode 调试工具	434	20.1	内存优化	487
18.1.1	定位编译错误	434	20.1.1	内存管理	487
18.1.2	查看和显示日志	435	20.1.2	使用 Analyze 和 Instruments 工具 解决内存泄漏问题	487
18.1.3	设置和查看断点	437	20.1.3	查找和解决僵尸对象	494
18.1.4	调试工具	442	20.1.4	autorelease 的使用问题	497
18.1.5	输出窗口	443	20.1.5	响应内存警告	497
18.1.6	变量查看窗口	444	20.2	优化资源文件	499
18.1.7	查看线程	445	20.2.1	图片文件优化	499
18.2	日志与断言输出	446	20.2.2	音频文件优化	500
18.2.1	使用 NSLog 函数	446	20.3	延迟加载	501
18.2.2	使用断言	447	20.3.1	资源文件的延迟加载	501
18.3	异常堆栈报告分析	448	20.3.2	故事板文件的延迟加载	505
18.3.1	跟踪异常堆栈	448	20.3.3	xib 文件的延迟加载	507
18.3.2	分析堆栈报告	450	20.4	数据持久化的优化	509
18.4	在 iOS 设备上调试	450	20.4.1	使用文件	509
18.4.1	创建开发者证书	451	20.4.2	使用 SQLite 数据库	513
18.4.2	设备注册	455	20.4.3	使用 Core Data	514
18.4.3	创建 App ID	456	20.5	可重用对象的使用	516
18.4.4	创建配置概要文件	459	20.5.1	表视图中的可重用对象	516
18.4.5	设备调试	460	20.5.2	集合视图中的可重用对象	518
18.5	Xcode 设备管理工具	461	20.5.3	地图视图中的可重用对象	519
18.5.1	查看设备上的应用程序	462	20.6	并发处理与多核 CPU	520
18.5.2	设备日志	464	20.6.1	主线程阻塞问题	520
18.6	小结	464	20.6.2	选择 NSThread 还是 GCD	521
第 19 章 测试驱动下的 iOS 应用开发			20.7 小结		
19.1	测试驱动的软件开发概述	465			521
19.1.1	测试驱动的软件开发流程	465			

第 21 章 管理好你的程序代码——代码版本控制.....	522	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
21.1 概述.....	522	23.1 移动网络通信应用的分层架构设计.....	584
21.1.1 版本控制历史.....	522	23.2 Objective-C 版本：在工作空间中添加和配置 MKNetworkKit.....	585
21.1.2 基本概念.....	523	23.3 基于委托模式实现.....	587
21.2 Git 代码版本控制.....	523	23.3.1 网络通信与委托模式.....	587
21.2.1 服务器搭建.....	523	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
21.2.2 Gitolite 服务器管理.....	525	23.3.3 类图.....	588
21.2.3 Git 常用命令.....	527	23.3.4 时序图.....	591
21.2.4 Git 分支.....	529	23.3.5 数据持久层重构.....	593
21.2.5 Git 协同开发.....	533	23.3.6 业务逻辑层的代码实现.....	597
21.2.6 Xcode 中 Git 的配置与使用.....	535	23.3.7 表示层的代码实现.....	599
21.3 GitHub 代码托管服务.....	542	23.4 基于观察者模式的通知机制实现.....	603
21.3.1 创建和配置 GitHub 账号.....	542	23.4.1 观察者模式的通知机制回顾.....	603
21.3.2 创建代码库.....	544	23.4.2 异步网络通信中通知机制的分层架构设计.....	604
21.3.3 删除代码库.....	546	23.4.3 类图.....	605
21.3.4 派生代码库.....	546	23.4.4 时序图.....	606
21.3.5 使用 GitHub 协同开发.....	549	23.4.5 数据持久层的重构.....	609
21.3.6 管理组织.....	555	23.4.6 业务逻辑层的代码实现.....	611
21.4 小结.....	558	23.4.7 表示层的代码实现.....	612
第 22 章 把你的应用放到 App Store 上.....	559	23.5 小结.....	616
22.1 收官.....	559	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
22.1.1 在 Xcode 6 中添加图标.....	559	24.1 应用分析与设计.....	617
22.1.2 在 Xcode 6 中添加启动界面.....	562	24.1.1 应用概述.....	617
22.1.3 调整 Identity 和 Deployment Info 属性.....	564	24.1.2 需求分析.....	617
22.1.4 为发布进行编译.....	567	24.1.3 原型设计.....	618
22.2 发布流程.....	571	24.1.4 数据库设计.....	619
22.2.1 创建应用.....	572	24.1.5 架构设计.....	619
22.2.2 基本信息输入.....	574	24.2 iOS 敏捷开发.....	620
22.2.3 应用定价.....	577	24.2.1 敏捷开发宣言.....	620
22.2.4 上传应用.....	577	24.2.2 iOS 适合敏捷开发?.....	621
22.2.5 提交审核.....	579	24.2.3 iOS 敏捷开发最佳实践.....	621
22.3 常见审核不通过的原因.....	581	24.3 任务 1：创建应用工作空间和工程.....	623
22.3.1 功能问题.....	581	24.3.1 迭代 1.1：创建 Swift 版本的工程.....	623
22.3.2 用户界面问题.....	581	24.3.2 迭代 1.2：创建 Objective-C 版本工作空间.....	623
22.3.3 商业问题.....	581	24.3.3 迭代 1.3：发布到 GitHub.....	623
22.3.4 不当内容.....	581		
22.3.5 其他问题.....	582		
22.4 小结.....	582		

24.4	任务 2: 信息系统层与持久层开发	624	24.6.1	迭代 4.1: 使用资源目录管理 图片和图标资源	641
24.4.1	迭代 2.1: 编写数据库 DDL 脚本	624	24.6.2	迭代 4.2: 根据原型设计初步 设计 iPad 故事板	642
24.4.2	迭代 2.2: 插入初始数据到 数据库	625	24.6.3	迭代 4.3: 根据原型设计初步 设计 iPhone 故事板	644
24.4.3	迭代 2.3: 编写实体类	625	24.6.4	迭代 4.4: 首页模块	644
24.4.4	迭代 2.4: DAO 类 XCTest 单元 测试	626	24.6.5	迭代 4.5: 比赛项目模块	645
24.4.5	迭代 2.5: 数据库帮助类 DBHelper	629	24.6.6	迭代 4.6: 比赛日程模块	649
24.4.6	迭代 2.5: 编写 DAO 类	632	24.6.7	迭代 4.7: 倒计时模块表示层	652
24.4.7	迭代 2.6: 发布到 GitHub	635	24.6.8	迭代 4.8: 关于我们模块表示层	653
24.5	任务 3: 业务逻辑层开发	636	24.6.9	迭代 4.9: 发布到 GitHub	654
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	24.7	任务 5: 收工	654
24.5.2	迭代 3.2: 编写比赛项目业务 逻辑类	637	24.7.1	迭代 5.1: 添加图标	654
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	24.7.2	迭代 5.2: 设计和添加启动界面	654
24.5.4	迭代 3.4: 编写比赛日程业务 逻辑类	639	24.7.3	迭代 5.3: 植入谷歌 AdMob 横幅 广告	655
24.5.5	迭代 3.5: 发布到 GitHub	640	24.7.4	迭代 5.4: 性能测试与改善	657
24.6	任务 4: 表示层开发	641	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

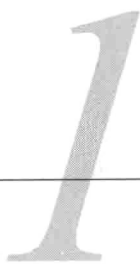
Part 1

第一部分

基础篇

本部分内容

- 第1章 开篇综述
- 第2章 第一个 iOS 应用程序
- 第3章 UIView 与视图
- 第4章 iOS 8 多分辨率屏幕适配
- 第5章 委托协议、数据源协议与高级视图
- 第6章 表视图
- 第7章 视图控制器与导航模式
- 第8章 iOS 常用设计模式
- 第9章 iPhone 与 iPad 应用开发的差异
- 第10章 iOS 分层架构设计
- 第11章 数据持久化
- 第12章 应用程序设置
- 第13章 国际化
- 第14章 访问联系人



自从App Store上线以来，它创造了很多神话，给我们这些程序员提供了展示自己的舞台，给了我们创意的空间，给了我们创业的机会。下面让我们从这里开始iOS开发之旅吧。

1.1 iOS 概述

在本节中，我们将了解什么是iOS以及iOS 8有哪些新特性。

1.1.1 iOS 介绍

iOS是由苹果公司开发的移动设备操作系统，这些移动设备包括iPhone、iPod touch、iPad和Apple TV等，目前最新的操作系统是iOS 8。

苹果公司最早于2007年1月9日的Macworld大会上公布了这个系统，最初是设计给iPhone使用的，后来陆续适用于到iPod touch、iPad和Apple TV等产品上。iOS与苹果的Mac OS X操作系统一样，都属于类Unix的商业操作系统。

原本这个系统的名字为iPhone OS，因为主要应用于iPhone和iPod touch设备，后来在2010 WWDC大会上宣布改名为iOS。

1.1.2 iOS 8 的新特性

苹果公司于2014年9月18日凌晨1点开放下载iOS 8的正式版，它支持iPhone 4s、iPhone 5、iPhone 5s、iPhone 5c、iPhone 6、iPhone 6 Plus、iPad 2、iPad 3、iPad mini 2、iPad Air、iPod touch 5和Apple TV等设备。据苹果发布的更新文档显示，iOS 8新增了多项功能，很多新特性或将成为将来的焦点。

现在我们先简要介绍一下iOS 8几个重要的变化。

- **家庭分享**。用户可以创建家庭分享，除创建者之外最多可以加入6个家庭成员。通过该功能，用户可以和家人分享位置、照片、日历、应用程序、音乐和视频等。
- **键盘**。苹果在iOS 8之后开放了键盘应用程序接口，相信iOS用户很快就能看到不少有趣的键盘应用。
- **Touch ID**。第三方应用可以使用Touch ID接口，这意味着未来的很多应用都可以用指纹识别功能了。
- **iCloud Drive**。为安装iOS 8或OS X Yosemite系统的设备提供了电子文档管理服务。这项新功能允许用户在iCloud上访问相片、视频、电子文档、音乐、应用数据等信息，也允许用户跨平台操控，实现多个苹果设备的数据同步。用户默认可以获得多达5GB的免费云端存储空间。当然，用户也可以另外购买更大的空间。
- **Handoff**。这为安装iOS 8或OS X Yosemite系统的设备之间通过Wi-Fi路由器或个人便携式热点来分享电子文档、E-mail和网站信息提供了便利。此外，该功能也能将Mac的数据进度同步到iOS设备上来。
- **HealKit**。这是一项提供给开发者的SDK和服务。通过HealKit，开发人员可以随时查看各种健康和健身相关的信息，比如血液、心率、水化、血压、营养、血糖、睡眠、呼吸频率、血氧饱和度和体重等。这些信息来自iOS设备内置的传感器以及第三方健康外设。HealKit可作为每位用户健康数据的存储中心，苹果为HealthKit提供第三方应用接入，用户需要时可以提供给医疗机构。

□ HomeKit。这也是一项提供给开发者的SDK和服务，通过HomeKit可以开发智能家居类应用。苹果在发布会上阐述了未来美好的愿望，利用手机控制家庭事物，例如开启关闭门窗、定时烹调、自动空调、扫地机器人等，所有的东西将会通过网络和接口平台实时统一管理。

此外，iOS 8还对已有的框架进行了不同程度的增强和删减。

1.2 开发环境及开发工具

苹果公司于2008年3月6日发布了iPhone和iPod touch的应用程序开发包，其中包括Xcode开发工具、iPhone SDK和iPhone手机模拟器。第一个Beta版本是iPhone SDK 1.2b1 (build 5A147p)，发布后立即就能使用，但是同时推出的App Store所需要的固件更新直到2008年7月11日才发布。编写本书时，iOS SDK 8.1.2版本已经发布。

iOS开发工具主要是Xcode。自从Xcode 3.1发布以后，Xcode就成为iPhone软件开发工具包的开发环境。Xcode可以开发Mac OS X和iOS应用程序，其版本是与SDK相互对应的。例如，Xcode 3.2.5与iOS SDK 4.2对应，Xcode 4.1与iOS SDK 4.3对应，Xcode 4.2与iOS SDK 5对应，Xcode 4.5和Xcode 4.6与iOS SDK 6对应，Xcode 5与iOS SDK 7对应，Xcode 6与iOS SDK 8对应。

在Xcode 4.1之前，还有一个配套使用的工具Interface Builder，它是Xcode套件的一部分，用来设计窗体和视图，通过它可以“所见即所得”地拖曳控件并定义事件等，其数据以XML的形式被存储在xib文件中。在Xcode 4.1之后，Interface Builder成为了Xcode的一部分，与Xcode集成在一起。

注意 本书所有案例采用Xcode 6.2编写。由于Swift语法和iOS API的变化，如果采用低版本或者高版本的Xcode运行案例，有些地方可能会有编译错误。

打开Xcode 6工具，看到的主界面如图1-1所示。该界面主要分成3个区域，①号区域是工具栏，其中的按钮可以完成大部分工作。②号区域是导航栏，主要对工作空间中的内容进行导航。③号区域是代码编辑区，我们的编码工作就是在这里完成的。在导航栏上面还有一排按钮，如图1-2所示，默认选中的是“文件”导航面板。关于各个按钮的具体用法，我们会在以后用到的时候详细介绍。

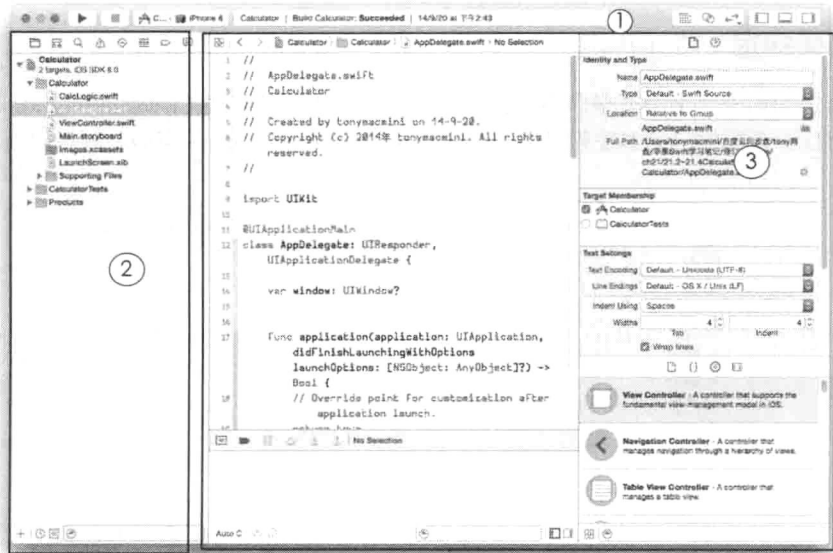


图1-1 Xcode主界面



图1-2 Xcode导航面板

在选中导航面板时，导航栏下面也有一排按钮，如图1-3所示。这是辅助按钮，它们的功能都与该导航面板的内容相关。对于不同的导航面板，这些按钮也是不同的。

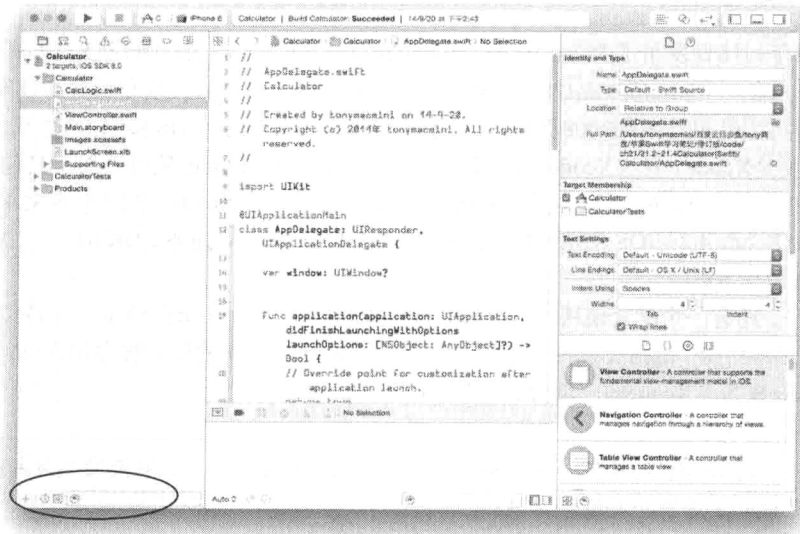


图1-3 导航面板的辅助按钮

1.3 本书中的约定

为了方便大家阅读本书，本节介绍一下书中案例代码和图示的相关约定。

1.3.1 案例代码约定

作为一本编程方面的书，书中有很多案例代码，我们可以从图灵网站（www.ituring.com.cn）本书主页免费注册下载或者从智捷课堂提供的本书服务网站（www.51work6.com/ios1.php）下载，解压后会看到如图1-4所示的目录结构。

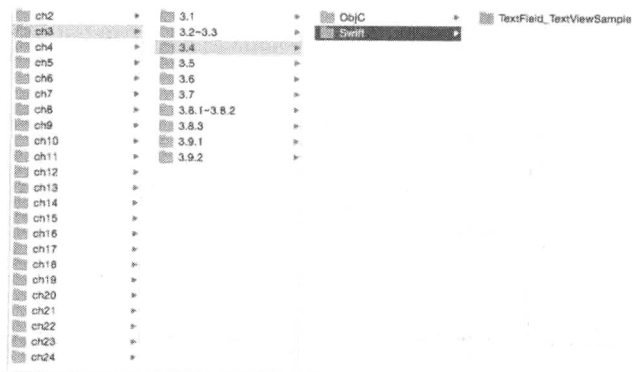


图1-4 源代码文件目录