



- 国内最新 iOS 7 开发详解，全面涵盖 iOS 7 新特性、新变化
- 数百个项目案例 + 两个真实商业项目开发全过程
- 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内容



# iOS 开发指南

## 从零基础到App Store上架 (第2版)

关东升 著

TURING 图灵原创

# iOS 开发指南

从零基础到App Store上架  
(第2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. — 2版. — 北京：人民邮电出版社，2014.4  
(图灵原创)  
ISBN 978-7-115-34802-9

I. ①i… II. ①关… III. ①移动终端—应用程序—  
程序设计—指南 IV. ①TN929.53-62

中国版本图书馆CIP数据核字(2014)第032993号

## 内 容 提 要

本书采用全新的 iOS 7 API, 详细介绍了最新的 iOS 7 开发相关的知识点。本书共分为 4 个部分: 第一部分为基础篇, 介绍了 iOS 的一些基础知识; 第二部分为网络篇, 介绍了 iOS 网络开发相关的知识; 第三部分为进阶篇, 介绍了 iOS 高级内容、商业思考等; 第四部分为实战篇, 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。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 <http://www.iosbook1.com> 下载。

本书适合 iOS 开发人员阅读。

- 
- ◆ 著 关东升  
责任编辑 王军花  
执行编辑 张 霞  
责任印制 焦志炜
  - ◆ 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>  
三河市海波印务有限公司印刷
  - ◆ 开本: 880×1230 1/16  
印张: 44.5  
字数: 1402千字 2014年4月第2版  
印数: 8 501 - 13 500册 2014年4月河北第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 2013 年 9 月 19 日凌晨 1 点，苹果公司发布了最新的 iOS 7。此次最大的变化是 UI 设计风格，苹果放弃了以前的拟物化设计，转而采用扁平化设计。《iOS 开发指南：从零基础到 App Store 上架》是在 iOS 7 发布之前出版的，上市后得到了广大读者的认可，同时我们也得到很多读者的反馈，希望将这本书升级为 iOS 7 版本。经过几个月的努力，我们终于在 2014 年元旦之前将书稿更新完毕。

## 内容和组织结构

本书是我们团队编写的 iOS 系列图书之一，目的是使具有 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 13 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、视图器的生命周期等，最后介绍了如何使用 API 帮助文档和官方案例。

第 3 章首先说明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，接着介绍了屏幕布局的内容以及一个较为复杂的控件——选择器，最后探讨了 iOS 中的集合视图。

第 4 章探讨了表视图的组成、表视图类的构成和表视图的分类，使读者对表视图有一个整体上的认识。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，然后讲解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 5 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 6 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 7 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 8 章重点介绍了 iOS 平台的分层架构设计。

第 9 章介绍了 iOS 7 中的文字排版和渲染引擎 Text Kit，以及如何利用 Text Kit 技术实现文本图片混合排版，设置动态字体。

第 10 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 11 章介绍了国际化概念、内容和目录结构，并详细阐述了文本信息、nib 及故事板、资源文件的国际化。

第 12 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 13 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 14 章介绍了数据交换格式，其中 XML 和 JSON 是主要的方式。这一章重点介绍了 Web Service 的访问以及 ASIHTTPRequest 框架。

第 15 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 6 苹果地图的使用，包括了显示地图、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 6 苹果地图和谷歌 Web 地图。

第三部分为进阶篇，共 7 章，主要介绍了 iOS 高级内容和商业思考等。

第 16 章介绍了从 iOS 6 升级到 iOS 7 带来的变化、遇到的困难以及解决的方法。

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，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 章首先介绍了一些常用的调试工具、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 章讨论了测试驱动的 iOS 开发，介绍了测试驱动开发流程。此外，还介绍了单元测试框架 OCMock、GHUnit 和 OCMock 的用法。

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、多线程以及程序编译参数等。

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，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，我们会在另外一本介绍 iOS 游戏开发的书中介绍，具体进展请读者关注我们智捷 iOS 课堂官方网站 <http://www.51work6.com>。

## 本书网站

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网站 <http://www.iosbook1.com>，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 <http://www.iosbook1.com> 下载，或者到图灵社区 ([www.ituring.cn](http://www.ituring.cn)) 本书主页免费注册下载。

## 勘误与支持

我们在网站 <http://www.iosbook1.com> 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](mailto: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

2013年12月于北京

欢迎加入

**图灵社区** ituring.com.cn

## ——最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

优惠提示：现在购买电子书，读者将获赠书款20%的社区银子，可用于兑换纸质样书。

## ——最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

## ——最直接的读者交流平台

在图灵社区，你可以十分方便地写作文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

你可以积极参与社区经常开展的访谈、乐译、评选等多种活动，赢取积分和银子，积累个人声望。

# 目 录

## 第一部分 基础篇

第 1 章 开篇综述	2	第 3 章 UIView 与控件	35
1.1 iOS 概述	2	3.1 视图“始祖”——UIView	35
1.1.1 iOS 介绍	2	3.1.1 UIView “家族”	35
1.1.2 iOS 7 新特性	2	3.1.2 应用界面的构建层次	37
1.2 开发环境及开发工具	3	3.1.3 视图分类	38
1.3 本书中的约定	4	3.2 标签控件和按钮控件	38
1.3.1 案例代码约定	5	3.2.1 标签控件	39
1.3.2 图示的约定	5	3.2.2 按钮控件	39
3.2.3 动作和输出口	41	3.3 TextField 控件和 TextView 控件	43
第 2 章 第一个 iOS 应用程序	7	3.3.1 TextField 控件	44
2.1 创建 HelloWorld 工程	7	3.3.2 TextView 控件	44
2.1.1 创建工程	7	3.3.3 键盘的打开和关闭	45
2.1.2 Xcode 中的 iOS 工程模板	12	3.3.4 关闭和打开键盘的通知	46
2.1.3 应用剖析	13	3.3.5 键盘的种类	47
2.2 HelloWorld 工程中故事板文件		3.4 开关控件、滑块控件和分段控件	48
Main.storyboard	16	3.4.1 开关控件	49
2.2.1 故事板的导航特点	16	3.4.2 滑块控件	49
2.2.2 故事板中的 Scene 和 Segue	17	3.4.3 分段控件	49
2.3 应用生命周期	17	3.5 网页控件 WebView	50
2.3.1 非运行状态——应用启动场景	19	3.5.1 WebView 介绍	50
2.3.2 点击 Home 键——应用退出场景	20	3.5.2 使用 WebView 构建 Hybrid 应用	53
2.3.3 挂起重新运行场景	21	3.6 屏幕滚动控件 ScrollView	58
2.3.4 内存清除——应用终止场景	22	3.6.1 ScrollView 属性的设置	58
2.4 视图生命周期	22	3.6.2 键盘与其他控件的协同	63
2.4.1 视图生命周期与视图控制器关系	23	3.7 等待相关的控件与进度条	65
2.4.2 iOS UI 状态保持和恢复	24	3.7.1 活动指示器 ActivityIndicatorView	66
2.5 设置产品属性	26	3.7.2 进度条 ProgressView	66
2.5.1 Xcode 中的 Project 和 Target	26	3.8 警告框和操作表	67
2.5.2 设置常用的产品属性	28	3.8.1 警告框 UIAlertView	67
2.6 iOS API 简介	30	3.8.2 操作表 ActionSheet	69
2.6.1 API 概述	30	3.9 工具栏和导航栏	70
2.6.2 如何使用 API 帮助	32	3.9.1 工具栏	70
2.7 小结	34	3.9.2 导航栏	73
		3.10 屏幕布局	77



3.10.1 iPad、iPhone 和 iPhone 5 屏幕 布局	77	5.2.2 基于分屏导航的实现	147
3.10.2 绝对布局和相对布局	78	5.2.3 基于分页导航的实现	150
3.10.3 使用 AutoLayout 布局	80	5.3 标签导航	156
3.10.4 旋转你的屏幕	82	5.3.1 应用场景	157
3.11 选择器	86	5.3.2 实现	157
3.11.1 日期选择器	86	5.4 树形结构导航	159
3.11.2 普通选择器	88	5.4.1 应用场景	160
3.11.3 数据源协议与委托协议	90	5.4.2 实现	161
3.12 集合视图	92	5.5 组合使用导航模式	165
3.12.1 集合视图介绍	92	5.5.1 应用场景	166
3.12.2 集合视图单元格	94	5.5.2 实现	166
3.12.3 数据源协议与委托协议	97	5.6 小结	173
3.13 小结	97	<b>第 6 章 iOS 常用设计模式</b>	174
<b>第 4 章 表视图</b>	98	6.1 单例模式	174
4.1 概述	98	6.1.1 问题提出	174
4.1.1 表视图的组成	98	6.1.2 实现原理	174
4.1.2 表视图的相关类	99	6.1.3 应用案例	175
4.1.3 表视图分类	100	6.2 委托模式	177
4.1.4 单元格的组成和样式	103	6.2.1 问题提出	177
4.1.5 数据源协议与委托协议	105	6.2.2 实现原理	178
4.2 简单表视图	106	6.2.3 应用案例	181
4.2.1 创建简单表视图	106	6.3 观察者模式	184
4.2.2 自定义单元格	110	6.3.1 问题提出	184
4.2.3 添加搜索栏	114	6.3.2 实现原理	185
4.3 分节表视图	119	6.3.3 通知机制和 KVO 机制	187
4.3.1 添加索引	119	6.4 MVC 模式	193
4.3.2 分组与静态表	122	6.4.1 MVC 模式概述	193
4.4 修改单元格	126	6.4.2 Cocoa Touch 中的 MVC 模式	194
4.4.1 删除和插入单元格	126	6.5 小结	196
4.4.2 移动单元格	132	<b>第 7 章 iPhone 与 iPad 应用开发的差异</b>	197
4.5 表视图 UI 设计模式	134	7.1 概述	197
4.5.1 分页模式	134	7.1.1 应用场景差异	197
4.5.2 下拉刷新模式	134	7.1.2 设计和开发需注意的问题	197
4.5.3 iOS 7 下拉刷新控件	135	7.1.3 构建自适应的 iPhone 和 iPad 工程	200
4.6 小结	137	7.2 iPad 专用 API	202
<b>第 5 章 视图控制器与导航模式</b>	138	7.2.1 UIPopoverController 控制器	202
5.1 概述	138	7.2.2 UISplitViewController 控制器	205
5.1.1 视图控制器的种类	138	7.2.3 模态视图专用属性	211
5.1.2 导航模式	138	7.3 小结	215
5.1.3 模态视图	139	<b>第 8 章 iOS 分层架构设计</b>	216
5.2 平铺导航	145	8.1 低耦合企业级系统架构设计	216
5.2.1 应用场景	145	8.2 iOS 分层架构设计	217

8.2.1 基于同一工程的分层	217	第 12 章 数据持久化	272
8.2.2 基于一个工作空间不同工程的分层	222	12.1 概述	272
8.3 小结	227	12.1.1 沙箱目录	272
第 9 章 iOS 7 中文字排版和渲染引擎——Text Kit	228	12.1.2 持久化方式	273
9.1 Text Kit 基础	228	12.2 属性列表	273
9.1.1 文字的排版和渲染	228	12.3 对象归档	278
9.1.2 Text Kit 架构	228	12.4 使用 SQLite 数据库	282
9.1.3 Text Kit 中的核心类	229	12.4.1 SQLite 数据类型	283
9.1.3 实例：凸版印刷效果	231	12.4.2 创建数据库	283
9.2 文字图片混合排版	233	12.4.3 查询数据	285
9.3 动态字体	236	12.4.4 修改数据	288
9.4 小结	239	12.5 Core Data	290
第 10 章 应用程序设置	240	12.5.1 ORM	290
10.1 概述	240	12.5.2 Core Data 堆栈	290
10.1.1 设置	240	12.5.3 建模和生成实体	294
10.1.2 配置	241	12.5.4 采用 Core Data 分层架构设计	297
10.2 应用程序设置包	243	12.5.5 查询数据	298
10.3 设置项目种类	244	12.5.6 修改数据	300
10.3.1 文本字段	247	12.6 小结	301
10.3.2 开关	249	第 13 章 访问通讯录	302
10.3.3 滑块	250	13.1 概述	303
10.3.4 值列表	252	13.2 读取联系人信息	303
10.3.5 子界面	253	13.2.1 查询联系人记录	304
10.4 读取设置	255	13.2.2 读取单值属性	306
10.5 小结	256	13.2.3 读取多值属性	308
第 11 章 国际化	257	13.2.4 读取图片属性	310
11.1 概述	257	13.3 写入联系人信息	310
11.1.1 需要国际化的内容	257	13.3.1 创建联系人	312
11.1.2 国际化目录结构	259	13.3.2 修改联系人	314
11.2 文本信息国际化	260	13.3.3 删除联系人	315
11.2.1 系统按钮和信息国际化	260	13.4 高级 API	316
11.2.2 应用名称国际化	261	13.4.1 选择联系人	316
11.2.3 程序代码输出的静态文本国际化	263	13.4.2 显示和修改联系人	319
11.2.4 使用 genstring 工具	265	13.4.3 创建联系人	322
11.3 xib 和故事板文件国际化	266	13.5 小结	325
11.3.1 使用 Base 国际化技术	266	第二部分 网络篇	
11.3.2 AutoLayout 与国际化	268	第 14 章 访问 Web Service	328
11.4 资源文件国际化	270	14.1 概述	328
11.4.1 图片资源文件国际化	270	14.2 数据交换格式	328
11.4.2 声音资源文件国际化	271	14.2.1 XML 文档结构	330
11.5 小结	271	14.2.2 解析 XML 文档	331

14.2.3	JSON 文档结构	338	16.1.4	iOS 7 状态栏样式设置	397
14.2.4	JSON 数据解码	339	16.2	从 Xcode 4 到 Xcode 5 的升级	399
14.3	REST Web Service	341	16.2.1	ARC 与 MRC 之争	399
14.3.1	HTTP 和 HTTPS 协议	341	16.2.2	故事板与 xib 之争	402
14.3.2	同步 GET 请求方法	342	16.2.3	找回普通显示屏的模拟器	407
14.3.3	异步 GET 请求方法	346	16.2.4	找回老版本的 Interface Builder	408
14.3.4	POST 请求方式	347	16.2.5	使用资源目录管理图片	409
14.3.5	调用 REST Web Service 的插入、 修改和删除方法	348	16.3	从 iPhone 4 到 iPhone 5 的升级	412
14.4	使用轻量级网络请求框架 MKNetworkKit	352	16.3.1	屏幕适配问题	413
14.4.1	ASIHTTPRequest、AFNetworking 和 MKNetworkKit 比较	353	16.3.2	从 32 位到 64 位的升级	416
14.4.2	安装和配置 MKNetworkKit 框架	353	16.4	小结	417
14.4.3	网络请求	354	<b>第 17 章 iOS 中的商业模式</b>	418	
14.4.4	下载数据	356	17.1	收费策略	418
14.4.5	上传数据	358	17.1.1	iOS 如何赚钱	418
14.5	反馈网络信息改善用户体验	359	17.1.2	避免定价策略误区	419
14.5.1	使用下拉刷新控件改善用户体验	359	17.1.3	免费软件的艺术	419
14.5.2	使用等待指示器控件	362	17.1.4	在适当的时间、适当的地点植入 广告	419
14.5.3	使用网络等待指示器	365	17.1.5	尝试不同的盈利模式	419
14.6	小结	366	17.2	使用苹果 iAd 广告	420
<b>第 15 章 定位服务与地图应用</b>	367		17.2.1	横幅广告	420
15.1	定位服务	367	17.2.2	插页广告	425
15.1.1	定位服务编程	368	17.2.3	查看你的收入	430
15.1.2	地理信息反编码	372	17.3	使用谷歌 AdMob 广告	432
15.1.3	地理信息编码查询	373	17.3.1	注册 AdMob 账号和管理应用	432
15.1.4	关于定位服务的测试	375	17.3.2	下载谷歌 AdMob Ads SDK 和 示例代码	435
15.2	使用 iOS 苹果地图	378	17.3.3	添加 AdMob 横幅广告	437
15.2.1	显示地图	378	17.3.4	添加 AdMob 插页广告	442
15.2.2	添加标注	380	17.3.5	为广告提交用户和位置信息	445
15.2.3	跟踪用户位置变化	383	17.3.6	搜索广告	446
15.3	使用程序外地图	384	17.3.7	查看你的收入	448
15.3.1	调用 iOS 苹果地图	384	17.4	应用内购买	449
15.3.2	调用谷歌 Web 地图	387	17.4.1	概述	449
15.4	小结	388	17.4.2	测试环境搭建	450
			17.4.3	在程序中实现应用内购买	455
			17.4.4	测试应用内购买	461
			17.5	小结	463
			<b>第 18 章 找出程序中的 bug——调试</b>	464	
			18.1	Xcode 调试工具	464
			18.1.1	定位编译错误	464
			18.1.2	查看和显示日志	465
			18.1.3	设置和查看断点	466

### 第三部分 进阶篇

<b>第 16 章 升级?</b>	390	
16.1	从 iOS 6 到 iOS 7 的升级	390
16.1.1	iOS 7 全新的扁平化设计	391
16.1.2	屏幕适配问题	394
16.1.3	iOS 7 状态栏隐藏	396

18.1.4	调试工具栏	470
18.1.5	输出窗口	471
18.1.6	变量查看窗口	473
18.1.7	查看线程	474
18.2	日志与断言输出	475
18.2.1	使用 NSLog 函数	475
18.2.2	使用 NSAssert 宏	476
18.2.3	移除 NSLog 和 NSAssert	477
18.3	LLDB 调试工具	479
18.3.1	断点命令	480
18.3.2	观察点命令	482
18.3.3	查看变量和计算表达式命令	483
18.4	异常堆栈报告分析	485
18.4.1	跟踪异常堆栈	485
18.4.2	分析堆栈报告	488
18.5	在 iOS 设备上调试	488
18.5.1	创建开发者证书	489
18.5.2	设备注册	493
18.5.3	创建 App ID	495
18.5.4	创建配置概要文件	497
18.5.5	设备调试	501
18.6	Xcode 设备管理工具	502
18.6.1	管理设备配置概要文件	503
18.6.2	查看设备上的应用程序	504
18.6.3	设备控制台	505
18.6.4	设备日志	505
18.7	小结	506
<b>第 19 章</b>	<b>测试驱动下的 iOS 应用开发</b>	<b>507</b>
19.1	测试驱动的软件开发概述	507
19.1.1	测试驱动的软件开发流程	507
19.1.2	测试驱动的软件开发案例	508
19.1.3	iOS 7 单元测试框架	513
19.2	使用 XCTest 测试框架	513
19.2.1	添加 XCTest 到工程	514
19.2.2	编写 XCTest 测试方法	515
19.2.3	运行测试程序	521
19.2.4	分析测试报告	521
19.3	iOS 单元测试最佳实践	523
19.3.1	测试数据持久层	523
19.3.2	测试业务逻辑层	527
19.3.3	测试表示层	529
19.4	小结	531
<b>第 20 章</b>	<b>让你的程序“飞”起来——性能优化</b>	<b>532</b>
20.1	内存优化	532
20.1.1	内存泄漏问题的解决	532
20.1.2	查找和解决僵尸对象	540
20.1.3	autorelease 的使用问题	543
20.1.4	响应内存警告	543
20.1.5	选择 xib 还是故事板	545
20.2	优化资源文件	545
20.2.1	图片文件优化	546
20.2.2	音频文件优化	548
20.3	延迟加载	549
20.3.1	资源文件的延迟加载	549
20.3.2	故事板和 xib 文件的延迟加载	553
20.4	数据持久化的优化	555
20.4.1	使用文件	555
20.4.2	使用 SQLite 数据库	559
20.4.3	使用 Core Data	560
20.5	可重用对象的使用	562
20.5.1	表视图中的可重用对象	562
20.5.2	集合视图中的可重用对象	564
20.5.3	地图视图中的可重用对象	565
20.6	并发处理与多核 CPU	565
20.6.1	主线程阻塞问题	566
20.6.2	选择 NSThread 还是 GCD	566
20.7	编译参数	568
20.8	小结	569
<b>第 21 章</b>	<b>管理好你的程序代码——代码版本控制</b>	<b>570</b>
21.1	概述	570
21.1.1	版本控制历史	570
21.1.2	基本概念	571
21.2	Git 代码版本控制	571
21.2.1	服务器搭建	571
21.2.2	Gitolite 服务器管理	573
21.2.3	Git 常用命令	575
21.2.4	Git 分支	577
21.2.5	Git 协同开发	581
21.2.6	Xcode 5 中 Git 的配置与使用	583
21.3	GitHub 代码托管服务	590
21.3.1	创建和配置 GitHub 账号	590
21.3.2	创建代码库	592
21.3.3	派生代码库	595

21.3.4	使用 GitHub 协同开发	597
21.3.5	管理组织	601
21.4	小结	605
<b>第 22 章</b>	<b>把你的应用放到 App Store 上</b>	<b>606</b>
22.1	收官	606
22.1.1	在 Xcode 5 下添加图标	606
22.1.2	Xcode 5 添加启动界面	608
22.1.3	调整 Identity 和 Deployment Info 属性	611
22.1.4	为发布进行编译	612
22.1.5	应用打包	618
22.2	发布流程	619
22.2.1	创建应用及基本信息	620
22.2.2	应用定价信息	621
22.2.3	最后的信息输入	622
22.2.4	上传应用	625
22.3	审核不通过的常见原因	627
22.4	小结	628
<b>第四部分 实战篇</b>		
<b>第 23 章</b>	<b>重构 MyNotes 应用——iOS 网络 通信中的设计模式与架构设计</b>	<b>630</b>
23.1	移动网络通信应用的分层架构设计	630
23.2	基于委托模式实现	631
23.2.1	网络通信与委托模式	631
23.2.2	在异步网络通信中使用委托模式 实现分层架构设计	631
23.2.3	类图	632
23.2.4	时序图	634
23.2.5	数据持久层重构	637
23.2.6	业务逻辑层的代码实现	640
23.2.7	表示层的代码实现	642
23.3	基于观察者模式的通知机制实现	646
23.3.1	观察者模式的通知机制回顾	646
23.3.2	异步网络通信中通知机制的分层 架构设计	647
23.3.3	类图	647
23.3.4	时序图	648
23.3.5	数据持久层的重构	651
23.3.6	业务逻辑层的代码实现	653
23.3.7	表示层的代码实现	653
23.4	小结	658
<b>第 24 章</b>	<b>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</b>	<b>659</b>
24.1	应用分析与设计	659
24.1.1	应用概述	659
24.1.2	需求分析	659
24.1.3	原型设计	660
24.1.4	数据库设计	660
24.1.5	架构设计	662
24.2	iOS 敏捷开发	662
24.2.1	敏捷开发宣言	662
24.2.2	iOS 适合敏捷开发吗	663
24.2.3	iOS 敏捷开发最佳实践	663
24.3	任务 1: 创建应用基本工作空间	665
24.4	任务 2: 信息系统层与持久层开发	666
24.4.1	迭代 2.1: 编写数据库 DDL 脚本	666
24.4.2	迭代 2.2: 插入初始数据到数据 库	667
24.4.3	迭代 2.3: 编写实体类	667
24.4.4	迭代 2.4: DAO 类 XCTest 单元 测试	668
24.4.5	迭代 2.5: 编写 DAO 类	672
24.4.6	迭代 2.6: 发布到 GitHub	676
24.5	任务 3: 业务逻辑层开发	676
24.5.1	迭代 3.1: 比赛项目业务逻辑类 XCTest 单元测试	676
24.5.2	迭代 3.2: 编写比赛项目业务逻 辑类	677
24.5.3	迭代 3.3: 比赛日程业务逻辑类 XCTest 单元测试	678
24.5.4	迭代 3.4: 编写比赛日程业务逻 辑类	679
24.5.5	迭代 3.5: 发布到 GitHub	681
24.6	任务 4: 表示层开发	681
24.6.1	迭代 4.1: 使用资源目录管理图 片和图标资源	682
24.6.2	迭代 4.3: 根据原型设计初步设 计 iPad 故事板	683
24.6.3	迭代 4.3: 根据原型设计初步设 计 iPhone 故事板	684
24.6.4	迭代 4.4: 首页模块	684
24.6.5	迭代 4.5: 比赛项目模块	685
24.6.6	迭代 4.6: 比赛日程模块	690
24.6.7	迭代 4.7: 倒计时模块表示层	692

24.6.8	迭代 4.8: 关于我们模块表示层	694	24.7.4	迭代 5.4: 性能测试与改善	696
24.6.9	迭代 4.9: 发布到 GitHub	694	24.7.5	迭代 5.5: 发布到 GitHub	697
24.7	任务 5: 收工	694	24.7.6	迭代 5.6: 在 App Store 上发布应用	697
24.7.1	迭代 5.1: 添加图标	695	24.8	小结	699
24.7.2	迭代 5.2: 设计和添加启动界面	695			
24.7.3	迭代 5.3: 植入谷歌 AdMob 横幅广告	695			

# 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 与控件
- 第4章 表视图
- 第5章 视图控制器与导航模式
- 第6章 iOS 常用设计模式
- 第7章 iPhone 与 iPad 应用开发的差异
- 第8章 iOS 分层架构设计
- 第9章 iOS 7 中文字排版和渲染引擎——Text Kit
- 第10章 应用程序设置
- 第11章 国际化
- 第12章 数据持久化
- 第13章 访问通讯录



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 7有哪些新特性。

### 1.1.1 iOS介绍

iOS的系统架构分为4层——Cocoa Touch层、Media层、Core Services层和Core OS层，相关内容可参见2.6.1节。下面我们简要介绍一下iOS的一些功能，具体如下所示。

- **多点触摸和手势。**触摸功能在iOS设备之前就被采用，但基本都是单点触摸，即只能用一个手指，而iOS设备能够感应多个手指的触摸。为了配合这种多点触摸，iOS上的触摸分为多种手势：触击、双击、滑动、长时间触击、轻拂、刷屏和手指合拢张开等。
- **统一的屏幕尺寸。**目前，iOS屏幕尺寸有4套：iPhone和iPod touch是3.5英寸，iPhone 5和第5代iPod touch是4英寸，iPad是9.7英寸，iPad mini是7.9英寸。统一的屏幕尺寸给应用软件开发带来很多好处，开发人员可以不用关心屏幕尺寸适配的问题，从而把精力集中在其他方面。
- **高分辨率。**iPhone 4S的屏幕分辨率是960 × 640，iPhone 5和第5代iPod touch的屏幕分辨率是1136 × 640，第1、2代iPad的屏幕分辨率是1024 × 768，第3代iPad的屏幕分辨率是2048 × 1536，而iPad mini的屏幕分辨率是1024 × 768。
- **重力加速计。**iOS内置了重力加速计。有了重力加速计，用户能够玩很多有意思的游戏（如极品飞车，它可以把iPhone作为方向盘，通过重力加速计感应方向的变化）。此外，还有很多与重力加速计有关的应用软件，如水平尺应用等。
- **指南针。**iOS内置了指南针设备。很多应用基于指南针，例如导航软件和地图应用软件。
- **蓝牙和Wi-Fi连接。**iOS内置了蓝牙和Wi-Fi通信模块。iOS设备之间可以采用Wi-Fi互相连接，也可以采用蓝牙进行连接，很多基于局域网的游戏就是通过这个功能实现的。当然，也可以通过Wi-Fi上网，这可以节约用户的上网费用。此外，iOS还可以与电脑连接。

### 1.1.2 iOS 7 新特性

iOS的最新版本为iOS 7.0。苹果公司于北京时间2013年9月19日凌晨1点开放其正式版的下载，它支持iPhone 4S、iPhone 5、iPad 2、iPad 3、iPad mini、iPad mini 2和iPod touch 5等设备。据苹果发布的更新文档显示，iOS 7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 7几个重要的变化。



- UI设计风格。UI设计风格发生了巨大的变化，苹果放弃了以前的拟物化设计，转而采用扁平化设计。拟物化设计风格强调逼真纹理。扁平化设计风格强调简约、抽象，半透明、精致的视觉感受、流畅而逼真的动作带给用户清新、活力的体验。
  - 视图动力学行为。它能够使界面中的元素像在真实的物理世界中一样运动起来，给用户逼真的视觉效果。有点像游戏中的物理引擎。
  - Text Kit。它能够管理视图中的文字和排版。可以为段落、列和页中的文字定义不同的风格。
  - 64位支持。它能够将应用编译成为支持64位设备运行的代码。这些设备有：iPhone 5S、iPad mini 2、iPad Air。
  - 多任务的增强。在iOS 7中，后台多任务增加了两种新模型：提取模式（etch）和远程通知模式（remote-notification）。提取模式适合于应用从网络中定期下载新数据，下载操作可以定期地被唤醒或者启动以在后台进行下载。在远程通知模式下，安装有iOS 7的设备在接收到远程推送通知时，可以启动一个后台下载操作任务，而无需用户干预。
  - Sprite Kit游戏引擎。Sprite Kit是苹果推出的2D游戏引擎，它意味着苹果要进入这一市场，Sprite Kit的使用有点类似于Cocos2d-iphone游戏引擎<sup>①</sup>。
  - 地图增强。苹果放弃了以前使用的谷歌地图，转而使用苹果自己的地图。MapKit框架本身没有变化，只是地图内容不再是谷歌地图。
  - AirDrop。能够与周围的设备进行通信，共享照片、文档、URL和少量数据。
  - 应用间音频（Inter-App Audio）。应用间音频可以实现支持程序间的音频访问，即在同一台设备上不同的两个程序之间发送MIDI命令和音频流。
  - 点对点连接（Peer-to-Peer Connectivity）。可以查找附近的设备，不需要通过互联网就可以直接与这些设备进行通信。该框架具有这样的特点：很容易创建多点会话，支持稳定、可靠和实时的数据传输。
- 此外，iOS 7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 7.0.4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。

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。

打开Xcode 5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，主要是对工作空间中的内容进行导航。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。关于各按钮的具体用法，我们会在以后用到的时候详细介绍。

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。

<sup>①</sup> <http://www.cocos2d-iphone.org/>。