

TURING 图灵原创

- 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
- 精彩手绘原型草图，艺术与科技的结合



iOS 开发指南

从零基础到App Store上架

关东升 编著

 人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

iOS 开发指南

从零基础到App Store上架

关东升 编著

人民邮电出版社
北京

图书在版编目 (CIP) 数据

iOS开发指南：从零基础到App Store上架 / 关东升
编著. — 北京：人民邮电出版社，2013.8
(图灵原创)
ISBN 978-7-115-32444-3

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(2013)第150303号

内 容 提 要

本书共4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。

-
- ◆ 编 著 关东升
责任编辑 王军花
责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 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：850×1168 1/16
印张：44
字数：1 376千字 2013年8月第1版
印数：1-3 500册 2013年8月北京第1次印刷

定价：99.00元

读者服务热线：(010)51095186转604 印装质量热线：(010)67129223

反盗版热线：(010)67171154

广告经营许可证：京崇工商广字第0021号

站在巨人的肩上
Standing on Shoulders of Giants



www.ituring.com.cn

前 言

2012年9月19日，苹果公司开放 iOS 6 下载。2012年12月14日，iPhone 5 在中国发售，此时 iOS 6 正渐入佳境。本书在 iOS 6 第一个 beta 阶段就开始了编写，想在本书中包括 iOS 6 的新增特性，并将其作为第一本国内原创 iOS 6 图书奉献给广大读者。几个月过去了，我们终于在 2013 年元旦之前将书稿提交给出版社。几个月来，我们智捷 iOS 课堂团队夜以继日，我几乎推掉一切社交活动，推掉很多企业邀请我去讲课的机会，每天工作 12 小时，不敢有任何松懈，不敢有任何模棱两可，只做一件事情——编写此书。

内容和组织结构

本书是我们团队编写 iOS 系列丛书中的一本，目的是使一个有 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 11 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 nib 和故事板技术创建了 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 4 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，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 6 中的集合视图。

第 5 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 6 章讨论了如何判断应用是不是需要一个导航功能，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 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，接着使用 nib 和故事板技术分别实现了两个重要的程序模板，最后介绍了 iOS 平台的分层架构设计。

第 8 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 9 章介绍了本地化概念、内容和目录结构，接下来详细阐述了文本信息、nib 及故事板、资源文件的本地化。

第 10 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 11 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 12 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 ASIHTTPRequest 框架。

第 13 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 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 14 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 15 章首先介绍了有哪些调试工具，然后重点介绍了几个常用的调试工具，具体包括日志与断言的输出、LLDB 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 16 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，我们还学习了单元测试框架 OCMock、GUnit 和 OCMock 的用法。

第 17 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 18 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 19 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 20 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 21 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 游戏开发书中介绍，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.com.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，我们会在第

一时间回复你。此外，你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
2012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 iOS 常用设计模式	34
1.1 iOS 概述	2	3.1 单例模式	34
1.1.1 iOS 介绍	2	3.1.1 问题提出	34
1.1.2 iOS 6 新特性	2	3.1.2 实现原理	34
1.2 开发环境及开发工具	3	3.1.3 应用案例	35
1.3 本书中的约定	4	3.2 委托模式	37
1.3.1 案例代码约定	4	3.2.1 问题提出	37
1.3.2 图示的约定	5	3.2.2 实现原理	39
第 2 章 第一个 iOS 应用程序	7	3.2.3 应用案例	41
2.1 创建基于 nib 的 HelloWorld 工程	7	3.3 观察者模式	44
2.1.1 创建工程	7	3.3.1 问题提出	44
2.1.2 Xcode 中的 iOS 工程模板	10	3.3.2 实现原理	45
2.1.3 应用剖析	11	3.3.3 通知机制和 KVO 机制	47
2.2 基于故事板的 HelloWorld 工程	13	3.4 MVC 模式	53
2.2.1 使用故事板重构 HelloWorld	14	3.4.1 MVC 模式概述	53
2.2.2 nib、xib 与故事板	15	3.4.2 Cocoa Touch 中的 MVC 模式	54
2.2.3 故事板中的 Scene 和 Segue	16	3.5 小结	56
2.3 应用生命周期	17	第 4 章 UIView 与控件	57
2.3.1 非运行状态——应用启动场景	18	4.1 视图“始祖”——UIView	57
2.3.2 点击 Home 键——应用退出场景	19	4.1.1 UIView “家族”	57
2.3.3 挂起重新运行场景	20	4.1.2 应用界面的构建层次	59
2.3.4 内存清除——应用终止场景	21	4.1.3 视图分类	60
2.4 视图生命周期	21	4.2 标签控件和按钮控件	60
2.4.1 视图生命周期与视图控制器关系	22	4.2.1 标签控件	61
2.4.2 iOS 6 UI 状态保持和恢复	23	4.2.2 按钮控件	61
2.5 设置产品属性	25	4.2.3 动作和输出口	63
2.5.1 Xcode 中的 Project 和 Target	25	4.3 TextField 控件和 TextView 控件	66
2.5.2 设置常用的产品属性	27	4.3.1 TextField 控件	66
2.6 iOS API 简介	29	4.3.2 TextView 控件	67
2.6.1 API 概述	29	4.3.3 键盘的打开和关闭	67
2.6.2 如何使用 API 帮助	31	4.3.4 关闭和打开键盘的通知	68
2.7 小结	33	4.3.5 键盘的种类	69
		4.4 开关控件、滑块控件和分段控件	70
		4.4.1 开关控件	71

4.4.2 滑块控件	71	5.3 分节表视图	138
4.4.3 分段控件	71	5.3.1 添加索引	138
4.5 网页控件 WebView	72	5.3.2 分组与静态表	141
4.5.1 WebView 介绍	72	5.4 修改单元格	144
4.5.2 使用 WebView 构建 Hybrid 应用	75	5.4.1 删除和插入单元格	144
4.6 屏幕滚动控件 ScrollView	80	5.4.2 移动单元格	150
4.6.1 ScrollView 属性的设置	80	5.5 表视图 UI 设计模式	152
4.6.2 键盘与其他控件的协同	84	5.5.1 分页模式	152
4.7 等待相关的控件与进度条	86	5.5.2 下拉刷新模式	152
4.7.1 活动指示器 ActivityIndicatorView	87	5.5.3 iOS 6 下拉刷新控件	153
4.7.2 进度条 ProgressView	88	5.6 小结	155
4.8 警告框和操作表	88	第 6 章 视图控制器与导航模式	156
4.8.1 警告框 UIAlertView	89	6.1 概述	156
4.8.2 操作表 ActionSheet	90	6.1.1 视图控制器的种类	156
4.9 工具栏和导航栏	92	6.1.2 导航模式	156
4.9.1 工具栏	92	6.1.3 模态视图	157
4.9.2 导航栏	94	6.2 平铺导航	163
4.10 屏幕布局	98	6.2.1 应用场景	163
4.10.1 iPad 与 iPhone 屏幕布局	98	6.2.2 基于分屏导航的实现	165
4.10.2 绝对布局和相对布局	99	6.2.3 基于分页导航的实现	168
4.10.3 使用 AutoLayout 布局	102	6.3 标签导航	173
4.10.4 旋转你的屏幕	103	6.3.1 应用场景	174
4.11 选择器	108	6.3.2 nib 实现	174
4.11.1 日期选择器	108	6.3.3 故事板实现	179
4.11.2 普通选择器	110	6.4 树形结构导航	182
4.11.3 数据源协议与委托协议	112	6.4.1 应用场景	182
4.12 iOS 6 中的集合视图	113	6.4.2 nib 实现	183
4.12.1 集合视图介绍	114	6.4.3 故事板实现	189
4.12.2 集合视图单元格	116	6.5 组合使用导航模式	193
4.12.3 数据源协议与委托协议	118	6.5.1 应用场景	193
4.13 小结	119	6.5.2 故事板实现	194
第 5 章 表视图	120	6.6 小结	200
5.1 概述	120	第 7 章 iPhone 与 iPad 应用开发的差异	201
5.1.1 表视图的组成	120	7.1 概述	201
5.1.2 表视图的相关类	121	7.1.1 应用场景差异	201
5.1.3 表视图分类	122	7.1.2 设计和开发需注意的问题	201
5.1.4 单元格的组成和样式	123	7.1.3 构建自适应的 iPhone 和 iPad 工程	204
5.1.5 数据源协议与委托协议	124	7.2 iPad 专用 API	206
5.2 简单表视图	125	7.2.1 UIPopoverController 控制器	206
5.2.1 创建简单表视图	125	7.2.2 UISplitViewController 控制器	209
5.2.2 自定义单元格	129	7.2.3 模态视图专用属性	215
5.2.3 添加搜索栏	133	7.3 Master-Detail 应用程序模板	219

7.3.1 nib 实现	220	10.2 属性列表	275
7.3.2 故事板实现	225	10.3 对象归档	280
7.4 Utility 应用程序模板	228	10.4 使用 SQLite 数据库	285
7.4.1 nib 实现	229	10.4.1 SQLite 数据类型	285
7.4.2 故事板实现	232	10.4.2 创建数据库	285
7.5 移动平台的分层架构设计	234	10.4.3 查询数据	287
7.5.1 低耦合企业级系统架构设计	234	10.4.4 修改数据	290
7.5.2 移动平台的分层架构设计	235	10.5 Core Data	292
7.5.3 基于同一工程的分层	235	10.5.1 ORM	292
7.5.4 基于一个工作空间不同工程的分层	241	10.5.2 Core Data 堆栈	293
7.6 小结	244	10.5.3 建模和生成实体	296
第 8 章 应用程序设置	245	10.5.4 采用 Core Data 分层架构设计	299
8.1 概述	245	10.5.5 查询数据	301
8.1.1 设置	245	10.5.6 修改数据	302
8.1.2 配置	246	10.6 小结	304
8.2 应用程序设置包	247	第 11 章 访问通讯录	305
8.3 设置项目种类	248	11.1 概述	305
8.3.1 文本字段	251	11.2 读取联系人信息	306
8.3.2 开关	253	11.2.1 查询联系人记录	307
8.3.3 滑块	254	11.2.2 读取单值属性	309
8.3.4 值列表	256	11.2.3 读取多值属性	311
8.3.5 子界面	257	11.2.4 读取图片属性	313
8.4 读取设置	259	11.3 写入联系人信息	313
8.5 小结	260	11.3.1 创建联系人	315
第 9 章 应用程序本地化	261	11.3.2 修改联系人	317
9.1 概述	261	11.3.3 删除联系人	318
9.1.1 本地化内容	261	11.4 高级 API	319
9.1.2 本地化目录结构	263	11.4.1 选择联系人	319
9.2 文本信息本地化	263	11.4.2 显示和修改联系人	322
9.2.1 系统按钮和信息本地化	263	11.4.3 创建联系人	324
9.2.2 应用名称本地化	265	11.5 小结	328
9.2.3 程序代码输出的静态文本本地化	266	第二部分 网络篇	
9.2.4 使用 genstring 工具	268	第 12 章 访问 Web Service	330
9.3 nib 和故事板文件本地化	268	12.1 概述	330
9.3.1 添加本地化	269	12.2 数据交换格式	330
9.3.2 开关使用 ibtool 工具	270	12.2.1 XML 文档结构	332
9.4 资源文件本地化	271	12.2.2 解析 XML 文档	333
9.5 小结	273	12.2.3 JSON 文档结构	340
第 10 章 数据持久化	274	12.2.4 JSON 数据解码	341
10.1 概述	274	12.3 REST Web Service	343
10.1.1 沙箱目录	274	12.3.1 HTTP 和 HTTPS 协议	343
10.1.2 持久化方式	275		

12.3.2	同步 GET 请求方法	344	14.2.1	横幅广告	396
12.3.3	异步 GET 请求方法	348	14.2.2	插页广告	401
12.3.4	POST 请求方式	349	14.2.3	查看你的收入	406
12.3.5	调用 REST Web Service 的插入、 修改和删除方法	350	14.3	使用谷歌 AdMob 广告	408
12.4	使用 ASIHTTPRequest 框架	355	14.3.1	注册 AdMob 账号和管理应用	408
12.4.1	安装和配置 ASIHTTPRequest 框架	355	14.3.2	下载谷歌 AdMob Ads SDK 和示 例代码	412
12.4.2	同步请求	356	14.3.3	添加 AdMob 横幅广告	414
12.4.3	异步请求	358	14.3.4	添加 AdMob 插页广告	418
12.4.4	使用请求队列	359	14.3.5	为广告提交用户和位置信息	421
12.4.5	上传数据	362	14.3.6	搜索广告	422
12.5	反馈网络信息改善用户体验	364	14.3.7	查看你的收入	424
12.5.1	iOS 6 表视图刷新控件的使用	364	14.4	应用内购买	425
12.5.2	使用等待指示器控件	367	14.4.1	概述	425
12.5.3	使用网络等待指示器	369	14.4.2	测试环境搭建	426
12.6	小结	370	14.4.3	在程序中实现应用内购买	431
			14.4.4	测试应用内购买	437
			14.5	小结	439
第 13 章	定位服务与地图应用	371	第 15 章	找出程序中的 bug——调试	440
13.1	定位服务	371	15.1	Xcode 调试工具	440
13.1.1	定位服务编程	371	15.1.1	定位编译错误	440
13.1.2	地理信息反编码	376	15.1.2	查看和显示日志	441
13.1.3	地理信息编码查询	377	15.1.3	设置和查看断点	442
13.1.4	关于定位服务的测试	379	15.1.4	调试工具栏	446
13.2	使用 iOS 6 苹果地图	382	15.1.5	输出窗口	447
13.2.1	显示地图	382	15.1.6	变量查看窗口	448
13.2.2	添加标注	384	15.1.7	查看线程	449
13.2.3	跟踪用户位置变化	387	15.2	日志与断言输出	450
13.3	使用程序外地图	388	15.2.1	使用 NSLog 函数	450
13.3.1	调用 iOS 6 苹果地图	388	15.2.2	使用 NSAssert 宏	451
13.3.2	调用谷歌 Web 地图	391	15.2.3	移除 NSLog 和 NSAssert	452
13.4	小结	392	15.3	LLDB 调试工具	455
			15.3.1	断点命令	455
			15.3.2	观察点命令	457
			15.3.3	查看变量和计算表达式命令	458
			15.4	异常堆栈报告分析	461
			15.4.1	跟踪异常堆栈	461
			15.4.2	分析堆栈报告	463
			15.5	在 iOS 设备上调试	464
			15.5.1	创建开发者证书	465
			15.5.2	设备注册	468
			15.5.3	创建 App ID	470
			15.5.4	创建配置概要文件	471
			15.5.5	设备调试	473

第三部分 进阶篇

第 14 章	iOS 中的商业模式	394
14.1	收费策略	394
14.1.1	iOS 如何赚钱	394
14.1.2	避免定价策略误区	395
14.1.3	免费软件的艺术	395
14.1.4	在适当的时间、适当的地点 植入广告	395
14.1.5	尝试不同的盈利模式	395
14.2	使用苹果 iAd 广告	396

15.6	Xcode 设备管理工具	474	17.4.1	使用文件	546
15.6.1	管理设备配置概要文件	474	17.4.2	使用 SQLite 数据库	549
15.6.2	查看设备上的应用程序	475	17.4.3	使用 Core Data	550
15.6.3	设备控制台	477	17.5	可重用对象的使用	552
15.6.4	设备日志	477	17.5.1	表视图中的可重用对象	553
15.7	小结	479	17.5.2	集合视图中的可重用对象	554
			17.5.3	地图视图中的可重用对象	555
第 16 章	基于测试驱动的 iOS 开发	480	17.6	并发处理与多核 CPU	556
16.1	测试驱动的软件开发概述	480	17.6.1	主线程阻塞问题	556
16.1.1	测试驱动的软件开发流程	480	17.6.2	选择 NSThread 还是 GCD	557
16.1.2	测试驱动的软件开发案例	481	17.7	编译器和编译参数	558
16.1.3	iOS 单元测试框架	486	17.7.1	GCC、LLVM GCC 与 Apple LLVM 比较	558
16.2	使用 OCMock 测试框架	486	17.7.2	ARM 架构	559
16.2.1	添加 OCMock 到工程中	486	17.7.3	Optimization Level	561
16.2.2	应用测试和逻辑测试	489	17.8	小结	562
16.2.3	编写 OCMock 测试方法	489	第 18 章	管理好你的程序代码——代码版本 控制	563
16.2.4	分析测试报告	494	18.1	概述	563
16.3	使用 GHUnit 测试框架	495	18.1.1	版本控制历史	563
16.3.1	添加 GHUnit 到工程	496	18.1.2	基本概念	564
16.3.2	编写 GHUnit 测试用例	498	18.2	Git 代码版本控制	564
16.3.3	分析测试报告	500	18.2.1	服务器搭建	564
16.4	使用伪对象	502	18.2.2	Gitolite 服务器管理	566
16.4.1	添加 OCMock 到工程	503	18.2.3	Git 常用命令	568
16.4.2	使用 OCMock 对象	505	18.2.4	Git 分支	570
16.5	iOS 单元测试最佳实践	507	18.2.5	Git 协同开发	574
16.5.1	iOS 单元测试策略	507	18.2.6	Xcode 中 Git 的配置与使用	576
16.5.2	测试数据持久层	507	18.3	GitHub 代码托管服务	583
16.5.3	测试业务逻辑层	512	18.3.1	创建和配置 GitHub 账号	584
16.5.4	测试表示层	515	18.3.2	创建代码库	586
16.6	小结	522	18.3.3	派生代码库	589
			18.3.4	使用 GitHub 协同开发	591
			18.3.5	管理组织	596
第 17 章	让你的程序“飞”起来——性能 优化	523	18.4	小结	599
17.1	内存优化	523	第 19 章	把你的应用放到 App Store 上	600
17.1.1	内存泄漏问题的解决	523	19.1	收官	600
17.1.2	查找和解决僵尸对象	531	19.1.1	添加图标	600
17.1.3	autorelease 的使用问题	534	19.1.2	添加启动界面	602
17.1.4	响应内存警告	534	19.1.3	调整 Application Target 属性	604
17.1.5	选择 nib 还是故事板	536	19.1.4	为发布进行编译	605
17.2	优化资源文件	537	19.1.5	应用打包	609
17.2.1	图片文件优化	537	19.2	发布流程	610
17.2.2	音频文件优化	538	19.2.1	创建应用及基本信息	611
17.3	延迟加载	539			
17.3.1	资源文件的延迟加载	540			
17.3.2	故事板和 nib 文件的延迟加载	543			
17.4	数据持久化的优化	546			

19.2.2	应用定价信息	612
19.2.3	最后信息输入	613
19.2.4	上传应用	616
19.3	常见审核不通过的原因	618
19.4	小结	619

第四部分 实战篇

第 20 章 重构 MyNotes 应用——iOS 网络通信中的设计模式与架构设计

20.1	移动网络通信应用的分层架构设计	622
20.2	基于委托模式实现	623
20.2.1	网络通信与委托模式	623
20.2.2	在异步网络通信中使用委托模式实现分层架构设计	623
20.2.3	类图	624
20.2.4	时序图	626
20.2.5	数据持久层的代码实现	629
20.2.6	业务逻辑层的代码实现	631
20.2.7	表示层的代码实现	632
20.3	基于观察者模式的通知机制实现	637
20.3.1	观察者模式的通知机制回顾	637
20.3.2	异步网络通信中通知机制的分层架构设计	638
20.3.3	类图	638
20.3.4	时序图	639
20.3.5	数据持久层的代码实现	642
20.3.6	业务逻辑层的代码实现	643
20.3.7	表示层的代码实现	644
20.4	小结	649

第 21 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

21.1	应用分析与设计	650
21.1.1	应用概述	650
21.1.2	需求分析	650
21.1.3	原型设计	651
21.1.4	数据库设计	652
21.1.5	架构设计	652
21.2	iOS 敏捷开发	653
21.2.1	敏捷开发宣言	653
21.2.2	iOS 适合敏捷开发?	654
21.2.3	iOS 敏捷开发最佳实践	654

21.3	任务 1: 创建应用基本工作空间	656
21.4	任务 2: 信息系统层与持久层开发	657
21.4.1	迭代 2.1: 编写数据库 DDL 脚本	657
21.4.2	迭代 2.2: 插入初始数据到数据库	658
21.4.3	迭代 2.3: 编写实体类	658
21.4.4	迭代 2.4: DAO 类 GHUnit 单元测试	659
21.4.5	迭代 2.5: 编写 DAO 类	663
21.4.6	迭代 2.6: 发布到 GitHub	666
21.5	任务 3: 业务逻辑层开发	667
21.5.1	迭代 3.1: 比赛项目业务逻辑类 GHUnit 单元测试	667
21.5.2	迭代 3.2: 编写比赛项目业务逻辑类	668
21.5.3	迭代 3.3: 比赛日程业务逻辑类 GHUnit 单元测试	669
21.5.4	迭代 3.4: 编写比赛日程业务逻辑类	670
21.5.5	迭代 3.5: 发布到 GitHub	672
21.6	任务 4: 表示层开发	672
21.6.1	迭代 4.1: 根据原型设计初步设计 iPad 故事板	673
21.6.2	迭代 4.2: 根据原型设计初步设计 iPhone 故事板	674
21.6.3	迭代 4.3: 首页模块	674
21.6.4	迭代 4.4: 比赛项目模块	675
21.6.5	迭代 4.5: 比赛日程模块	679
21.6.6	迭代 4.6: 倒计时模块表示层	681
21.6.7	迭代 4.7: 关于我们模块表示层	683
21.6.8	迭代 4.8: 发布到 GitHub	683
21.7	任务 5: 收工	684
21.7.1	迭代 5.1: 添加图标	684
21.7.2	迭代 5.2: 设计和添加启动界面	684
21.7.3	迭代 5.3: 植入谷歌 AdMob 横幅广告	685
21.7.4	迭代 5.4: 性能测试与改善	686
21.7.5	迭代 5.5: 发布到 GitHub	687
21.7.6	迭代 5.6: 在 App Store 上发布应用	687
21.8	小结	690

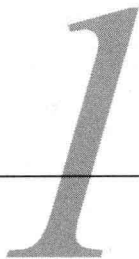
Part 1

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 iOS 常用设计模式
- 第4章 UIView 与控件
- 第5章 表视图
- 第6章 视图控制器与导航模式
- 第7章 iPhone 与 iPad 应用开发的差异
- 第8章 应用程序设置
- 第9章 应用程序本地化
- 第10章 数据持久化
- 第11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 6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、长期间触击、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 6 新特性

iOS的最新版本为iOS 6.0。苹果公司于2012年9月20日凌晨1点开放其正式版的下载，它支持iPhone 3GS、iPhone 4、iPhone 4S、iPhone 5、iPad 2、iPad 3、iPod touch 4和iPod touch 5等设备。据苹果发布的更新文档显示，iOS 6新增了200多项功能，更好地支持了中国市场，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 6几个重要的新性能。

- ❑ **地图**。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。
 - ❑ **社交网络 (social network)**。iOS 5集成了Twitter，iOS 6新集成了Facebook和新浪微博，还提供了一个新的视图控制器UIActivityViewController。使用这个视图控制器，可以非常方便地发送短消息和邮件，而且将内容复制到剪贴板之后，还可以利用视图控制器发消息到Twitter、Facebook以及新浪微博。
 - ❑ **Pass Kit**。它是开发Passbook的API。Passbook可以集中管理各种优惠券、打折卡、登机牌，这样以后就不必在钱包中放一堆卡了。个人认为该新功能在未来电子商务的发展中会有不容小视的贡献。
 - ❑ **游戏中心 (Game Center)**。Game Center是苹果在发布iOS 4时引入的API，目的是开发基于苹果Game Center的游戏。iOS 6对Game Center的API进行了升级。
 - ❑ **提醒 (Reminder)**。在iOS 6中，苹果开放了访问设备中的提醒API，开发者可以通过Event Kit读写提醒信息。
 - ❑ **应用内购买 (In-App Purchase, 缩写为IAP)**。使用这个功能，可以在应用中购买付费道具，增加新功能，订阅杂志，也可以购买和下载iTunes Store上的音乐、电影和图书。
 - ❑ **集合视图 (Collection View)**。它是一种增强网格视图，虽然这种网格视图在开源社区中已经有开源代码，但是使用起来不是很方便，而有了iOS 6的集合视图API后，使用起来变得非常方便。
 - ❑ **界面状态保持 (UI State Preservation)**。iOS 6之前，在应用退出，进入后台并且被终止的情况下，如果需要保持界面中UI控件的状态，则需要通过自己编写代码读写数据来实现，而在iOS 6之后这些工作就变得非常简单了。
 - ❑ **自动布局 (Auto Layout)**。这个功能首先应用于Mac OS X 10.7下的开发，现在也可以在iOS 6中使用，它为视图布局定义一套约束，而约束定义了两个界面中视图之间的关系。
 - ❑ **数据隐私 (Data Privacy)**。为了防止隐私泄漏，在iOS 6中访问联系人、日历、提醒和照片库时，需要经过用户允许。以前只有访问位置信息时，才需要经过用户允许。
- 此外，iOS 6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 6.1.2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与iOS SDK 6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 4.5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。