

# iOS 7

## iPhone/iPad

### 应用开发技术详解

刘一道 著

---

iOS 7  
iPhone & iPad Application Development

---

- 技术新颖，基于iOS 7撰写，详细讲解开发iPhone和iPad应用所需掌握的各项基础知识、开发流程和高级技巧，是系统学习iOS应用开发的经典著作
- 资深Mac/iOS工程师撰写，内容循序渐进，实战性强，每个知识点都准备了辅助理解的小案例，最后还提供两个综合性案例



机械工业出版社  
China Machine Press

# iOS 7

## iPhone/iPad

### 应用开发技术详解

---

iOS 7  
iPhone & iPad Application Development

---

刘一道 著



机械工业出版社  
China Machine Press

## 图书在版编目 ( CIP ) 数据

iOS 7: iPhone/iPad 应用开发技术详解 / 刘一道著. —北京: 机械工业出版社, 2013.11

ISBN 978-7-111-44051-2

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

中国版本图书馆 CIP 数据核字 (2013) 第 218499 号

**版权所有 · 侵权必究**

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书由资深 Mac/iOS 开发工程师基于 iOS 7 撰写。内容全面,从 Objective-C 语法知识、iOS 功能特性,到高级开发方法和技巧,几乎涵盖了中初级 iOS 开发工程师需要掌握的所有技术和知识;实战性强,每个知识点都有辅助理解的小案例,最后还提供了两个综合性应用开发案例。内容循序渐进,是系统学习 iOS 应用开发的经典著作。

全书共 20 章,分 4 部分。准备篇(第 1 ~ 2 章)介绍了 iOS、Objective-C 和 Xcode 的入门知识,详细描述了 iOS 7 新特性、Objective-C 应用开发环境的搭建,以及 iPhone 应用开发的完整过程;语法篇(第 3 ~ 8 章)详细讲解 Objective-C 的语法知识,其中包括类、对象、消息和协议,以及内存管理,重点讲解了 Foundation 框架常用类的使用方法;基础篇(第 9 ~ 18 章)讲解了 iOS 应用开发的框架,包含多状态和多任务、视图和视图控制器、事件和通知、音频和视频等,这些内容是本书的重点;实战篇(第 19 ~ 20 章)详细讲解了两个实战案例的完整开发过程,分别是基于 iPad 的应用程序“精灵小书柜”和基于 iPhone 的“弹球游戏”,目标是通过两个案例把前面学到知识应用到实践中,帮助读者真正掌握 iOS 应用开发的过程。

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:白宇

北京市荣盛彩色印刷有限公司印刷

2013 年 11 月第 1 版第 1 次印刷

186mm × 240mm · 33 印张

标准书号:ISBN 978-7-111-44051-2

定 价:79.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

客服热线:(010) 88378991 88361066

投稿热线:(010) 88379604

购书热线:(010) 68326294 88379649 68995259

读者信箱:hzsjj@hzbook.com

# 前 言

## 如何成为一名软件开发工程师

从大学毕业到如今，不知不觉敲代码已有十余载。从初期的“乱石穿空，惊涛拍岸，卷起千堆雪”，到如今的“道可道也，非恒道也。名可名也，非恒名也”，真可谓“谈笑间，檣櫓灰飞烟灭”。少了份“浮躁”，多了份“平静与思考”。懂得了敲代码正如雕刻家雕刻作品一样，需要一点一点精雕细刻，才可能创造出“艺术品”；懂得了“道”存在于代码之间，正所谓“为学者日益，为道者日损，损之又损，以至于无为”。

对于软件开发这个殿堂，估计很多人会长久徘徊于门外，特别是那些想迈进这个大门的初学者或者爱好者。实际上，当初的我也是带着很多的困惑和很大的压力步入这个行业的。

大学期间留给我们的多是“一无所有”的四年。而吾等众生，面对毕业之后父母“断奶”之迫，需要自己靠双手来“自力更生”；而面对校园外的“招聘”，自己总是那么惶惶不安缺乏“自信”，特别是参加过几场人才招聘会之后，真可谓“欲哭无泪”呀！

引用普希金的诗《假如生活欺骗了你》：“假如生活欺骗了你，不要伤心，不要哭泣；这样的日子里，需要勇气；要相信快乐的日子总会来临。”只要你有梦想有激情，有强烈的学习欲望，有较强的逻辑思维能力和一定的英语基础，热爱软件开发工作，同时能做到持之以恒，相信你不但能成为一名职业的软件开发工程师，而且还能成为一名优秀的软件开发工程师。

这一行是靠能力说话的，同时，这又是一个充满激情的行业。不管你是否毕业，不管你是否是计算机专业，甚至不管你是否上过大学，只要肯努力，有上进心，都可以加入软件开发团队中。也许你就是中国未来的比尔·盖茨或者乔布斯，世界会因“你”而精彩（李开复很喜欢这个座右铭）！

这里我还想说两点，希望对即将毕业或者刚步入大学的你有用：

- 对于高校在读的学生，不管将来要从事什么行业，建议早有规划，早做一些准备。以免毕业做“啃老族”。
- 对于即将大学毕业或者已经毕业的学生，建议最好不要参加社会上的“软件开发培训”。一是需要一笔可观的培训费（如果经济条件好，参加也不错）；二是不利于自学能力的培养。因为从事软件开发行业，需要保持一颗不断学习的心，否则很容易被淘汰。

## 为什么学习 iPhone/iPad 应用开发技术

### 1) 基于 iPhone/iPad 的应用开发已成为当今移动智能设备两大主流应用开发之一。

最近几年国内外从事基于 iPhone/iPad 开发的人员增速一直位于前列，已经形成了一个庞大的开发队伍。无论是传统互联网还是移动互联网，iPhone/iPad 应用设计理念一直引导着众多人模仿、推崇和跟随。iPhone/iPad 已成为当今移动智能设备应用发展的里程碑。

### 2) 支持 iPhone/iPad 的操作系统 iOS 是当今移动智能设备运行最稳定的操作系统。

当今市场上应用最广的两大主流智能移动设备操作系统分别是 Android 和 iOS。前者系统的稳定性、开销性和安全性成为“三座大山”，难以“移动”。很多用过 Android 手机的用户都遇到过手机“死机”、“莫名重启”等问题，而 iOS 就很少发生这类情况。

### 3) iPhone/iPad 框架的原生态语言是 Objective-C。

选择 Objective-C 作为 iPhone/iPad 框架的原生态语言有许多方面的原因。

- 首先，最主要的原因是，Objective-C 是面向对象的语言。iPhone/iPad 框架中的很多功能只能通过面向对象技术来呈现。iPhone/iPad SDK 用的就是 Objective-C。
- 其次，Objective-C 是标准 C 语言的一个超集，现存的 C 程序无须重新开发就能够使用 iPhone/iPad 软件框架，并且在 Objective-C 中可以使用 C 的所有特性。可以选择什么时候采用面向对象的编程方式（例如定义一个新的类），什么时候使用传统的面向过程的编程方式（定义数据结构和函数）。
- 此外，Objective-C 是一个简洁的语言，它的语法简单，没有歧义，易于学习。因为术语易于混淆以及抽象设计的重要性，对于初学者来说，面向对象编程的学习曲线比较陡峭。像 Objective-C 这种结构良好的语言使得成为一个面向对象程序员更为容易。介绍 Objective-C 的章节也如同其语言本身一样简洁。
- 和其他基于标准 C 语言的面向对象语言相比，Objective-C 对动态机制支持得更为彻底。编译器为运行环境保留了很多对象本身的数据信息，因此某些在编译时需要做出的选择就可以推迟到运行时来决定。这种特性使得基于 Objective-C 的程序非常灵活和强大。

### 4) 基于 iPhone/iPad 的应用市场已经成为“成熟有价市场”。

苹果应用商店（App Store）是苹果公司为其 iPhone、iPod touch、iPad、Mac 等产品创建和维护的数字化应用发布平台，允许用户从 iTunes Store 浏览和下载一些由 iOS SDK 或者 Mac SDK 开发的应用程序。根据应用发布的不同情况，用户可以付费或者免费下载。

如果你有不错的优秀产品，可以发布到苹果商店，标上价格。如果有人下载你的应用就会很有信用地“付款”，达到一定的下载数量，也是一笔可观收入。国内有很多个人和小公司，通过发布的应用已经获得了很大的成功。

## 本书适合哪些读者

- 对软件开发，特别是对 iPhone/iPad 开发有兴趣的人。
- 想成为一名专职的软件开发人员。
- 在校的学生可将本书作为学习计算机软件开发的教程。
- 想了解有关 iPhone/iPad 软件开发方面知识的从业人员，本书能帮你很快“过渡”。
- 开设相关专业课程的大专院校。

## 你将学到什么

本书是 iPhone/iPad 开发的入门级书，也是系统介绍支持 iPhone/iPad 应用开发框架的书。本书最大的作用就是引导你入门，只要你有梦想，有激情，通过本书你将会学到 iPhone/iPad 开发入门所需要掌握的基本知识。掌握本书知识，便可以迈进 iPhone/iPad 开发的门槛。

但是想成为一名优秀的 iPhone/iPad 软件开发人员，还需要很多方面的知识，所以希望阅读完本书的你，继续深造学习，使自己“更上一层楼”！

## 你该如何阅读本书

本书采取循序渐进的方式。对于有一定的软件开发经验，或者缺乏耐心的读者，可以根据“需求”选择不同的章节来阅读，这也是最有效的读书方式，也是我推崇的读书方式。

本书共 20 章，从内容上可以分四部分。

准备篇（第 1 ~ 2 章）：首先介绍一些 iOS、Objective-C 和 Xcode 的入门知识，使读者了解这三者之间的关系，粗略了解它们的发展史；并且关注了 iOS 7 的新特性。然后介绍了大家熟悉的 Objective-C 应用开发环境的搭建，使读者初次感受 iPhone 开发过程。经过这部分的学习使读者对有关 iPhone/iPad 开发有一个大致的了解，为以后应用开发做准备。

语法篇（第 3 ~ 8 章）：主要介绍 Objective-C 语法知识，这部分内容有些枯燥，但却是日后开发过程中必须掌握的基础知识。其中包括类、对象、消息和协议，以及内存管理。重点介绍了 Foundation 框架常用类的使用方法。

基础篇（第 9 ~ 18 章）：重点介绍 iOS 应用开发的框架，使读者掌握实际应用开发所需要的基础知识，这一部分是本书的重点。通过本章的学习，读者应该掌握多状态和多任务、视图和视图控制器、事件和通知、音频和视频。

实战篇（第 19 ~ 20 章）：由于篇幅有限，实战环节只列出两个案例，分别是基于 iPad 的应用程序和基于 iPhone 的小游戏。通过两个案例，把前面学到的知识应用到实践中，使读

者真正体验和了解软件开发的全过程。

## 勘误和支持

除封面署名外，参加本书编写工作的还有：孙振江、陈连增、边伟、郭合苍、郑军、吴景峰、杨珍民、王文朝、崔少闯、韦闪雷、刘红娇、王洁、于雪龙、孔琴。由于作者的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，特意在此公布我的博客地址 <http://blog.sina.com.cn/Beijingwolf>。可以将书中的错误发布在 Bug 勘误表页面中，如果你遇到任何问题可以留言，我将尽量在线上提供最满意的解答。同时，我也会在博客上不定期发布一些实战示例，供读者下载学习，来不断强化所学知识。

该书定稿的时候，苹果陆续公布了大量有关 iOS 7 的技术文档和范例，且一直在进行着。我会通过博客（<http://blog.sina.com.cn/Beijingwolf>）不定期发表有关 iOS 7 的最新材料，紧跟 iOS 7 最新的技术进展，为各位读者提供完善的技术材料和指导。书中实例的源文件可以从华章网站<sup>⊖</sup>下载，如果你有更多的宝贵意见，欢迎发送邮件至邮箱 [yifeilang@aliyun.com](mailto:yifeilang@aliyun.com)，期待能够得到你们的真挚反馈。

## 致谢

我爱读书，往往不求甚解。中学时虽翻阅过《道德经》、《奇门遁甲》等书，但不得要领，转瞬间，烟消云散。2012 年，到张瑞海先生的公司做事，有幸重温《道德经》。张先生是一位令我“醍醐灌顶”的人，“慧根不深”但“后天上进，悟性很高”，不但创造了庞大事业，还引导公司员工“禅悟”《道德经》。正是这个机会，使我悟到乔布斯在 iPhone/iPad 设计的“禅道”，在冥冥中明白了“0”和“1”，与“道生一，一生二，二生三，三生万物。万物负阴而抱阳，冲气以为和”相通。明白写代码“天下之至柔”才是代码“重构”之道，程序的设计最高境界在于“无为而无不为”。因有机会聆听先生之“布道”，使我掌握了 iPhone/iPad 设计之要领并写作本书。如今能与诸位同仁共享本书，真心感激张先生这种独特的“布道、禅道和悟道”之术。

高山之巅，白云之上，羽翔之光。我甚幸遇到机械工业出版社华章公司杨福川先生，他为我安排白宇女士作为本书的编辑。在编程行业，程序员都有一个过分的溺爱——容不得别人“批评”自己写的东西，正如母亲爱护自己的孩子一样。之前，与其他出版社合作出版过一本书，由于对方的编辑欠些“火候”，一本书将近一百个批注点，竟然只有三四个点“适得要领”，虽然该书已出版，但我总感觉缺些什么。而白宇女士，俨然一位“武林”高手，熟谙“点穴之法”，笔笔正中穴位之处，其批注使人心悦诚服，油然而敬佩，为本书大增“光色”。

最后，感谢的就是你，我亲爱的读者，感谢你拿起这本书，你们的认可，就是我的最大的快乐。

刘一道

---

⊖ 华章网站 [www.hzbook.com](http://www.hzbook.com)。

# 目 录

## 前言

## 第一部分 准备篇

### 第 1 章 初识 iOS、Objective-C 和 Xcode / 2

#### 1.1 认识 iOS / 2

- 1.1.1 iOS 的发展历程 / 2
- 1.1.2 iOS 的设计和功能特性 / 3
- 1.1.3 iOS 7 的新特性 / 5
- 1.1.4 iOS 架构 / 6
- 1.1.5 iOS 框架 / 8
- 1.1.6 iOS 系统框架的变迁 / 11
- 1.1.7 Mac OS X 和 iOS 平台  
不同框架的差异性 / 13
- 1.1.8 初步了解 iOS 开发者  
工具 / 16

#### 1.2 认识 Objective-C / 20

- 1.2.1 发展历程及版本变化 / 20
- 1.2.2 语言性能与分析 / 21

#### 1.2.3 框架和代码的关系 / 22

### 1.3 基于非苹果机平台搭建开发环境 / 23

- 1.3.1 前期准备 / 23
- 1.3.2 创建用于安装 Mac OS X  
的 VMWare 虚拟机 / 24
- 1.3.3 安装 Mac OS X / 26
- 1.3.4 安装 Xcode SDK / 30

#### 1.4 小结 / 33

### 第 2 章 创建你的第一个 iOS 应用程序 / 34

#### 2.1 应用程序的实现目标 / 34

#### 2.2 入门的开始 / 35

- 2.2.1 新建一个 Xcode 项目 / 35
- 2.2.2 在模拟器中查看应用程序  
的效果 / 37

#### 2.3 启动一个应用程序 / 39

- 2.3.1 探究 main.m 源文件 / 40
- 2.3.2 分析属性列表文件 / 41
- 2.3.3 查看串联图 / 41

- 2.4 检查视图控制器及其视图 / 43
  - 2.4.1 如何使用检查器 / 43
  - 2.4.2 更改视图的背景颜色 / 45
- 2.5 对视图进行配置和管理 / 47
  - 2.5.1 新增用户界面元素 / 47
  - 2.5.2 为按钮增添一个动作 / 51
  - 2.5.3 为文本框和标签创建插座 / 53
  - 2.5.4 打开 Connections 检查器验证连接 / 56
  - 2.5.5 对文本框进行委托处理 / 57
  - 2.5.6 让应用程序具有辅助功能 / 58
- 2.6 使用视图控制器完成应用程序 / 59
  - 2.6.1 给用户的名称添加属性 / 59
  - 2.6.2 实现 changeGreeting: 方法 / 60
  - 2.6.3 把视图控制器作为输入文本框的委托 / 61
- 2.7 测试应用程序 / 62
  - 2.7.1 排查和检测代码 / 62
  - 2.7.2 程序代码清单 / 63
- 2.8 小结 / 64
- 3.1.4 属性和存取方法 / 71
- 3.1.5 块 / 72
- 3.1.6 协议和类别 / 73
- 3.1.7 类型和编码策略 / 74
- 3.1.8 import 语句 / 76
- 3.2 Objective-C 2.0 新增特性 / 78
  - 3.2.1 关联引用 / 78
  - 3.2.2 快速枚举 / 80
  - 3.2.3 选择器 / 81
  - 3.2.4 静态类型的使用 / 84
- 3.3 进一步认识块 / 87
  - 3.3.1 块可以带参数和返回值 / 87
  - 3.3.2 块可以捕获封闭范围内的值 / 87
  - 3.3.3 在块内捕获变量值的变化 / 88
  - 3.3.4 块可以作为函数或者方法的参数 / 89
  - 3.3.5 用类型定义可以简化块语法 / 90
  - 3.3.6 使用属性可跟踪块 / 90
- 3.4 小结 / 91

## 第 4 章 类——构建应用程序的类型对象原型 / 92

## 第二部分 语法篇

### 第 3 章 Objective-C——构建 iOS 应用程序的基石 / 66

- 3.1 探窥 Objective-C 语言 / 66
  - 3.1.1 面向对象语言 Objective-C 是 C 语言的超集 / 67
  - 3.1.2 类和对象 / 67
  - 3.1.3 方法和发消息 / 69

- 4.1 认识根类 / 92
  - 4.1.1 NSObject 简介 / 92
  - 4.1.2 根类和协议 / 93
  - 4.1.3 根类方法 / 93
  - 4.1.4 根类接口规范 / 95
  - 4.1.5 根类实例方法和类方法 / 95
- 4.2 如何设计类 / 95
  - 4.2.1 设计接口 / 96
  - 4.2.2 设计实现 / 99

- 4.2.3 如何使用类名 / 100
  - 4.2.4 如何比较类 / 101
  - 4.3 类的类型 / 101
    - 4.3.1 如何指定静态类型 / 101
    - 4.3.2 类型的自查处理机制 / 102
  - 4.4 变量 / 102
    - 4.4.1 局部变量 / 102
    - 4.4.2 全局变量 / 103
    - 4.4.3 实例变量 / 103
    - 4.4.4 静态变量 / 106
    - 4.4.5 变量的存储类别 / 108
  - 4.5 属性 / 108
    - 4.5.1 属性的声明 / 109
    - 4.5.2 属性的实现 / 110
    - 4.5.3 属性类型和相关函数 / 111
    - 4.5.4 属性的类型编码 / 112
    - 4.5.5 属性重声明 / 112
    - 4.5.6 修改父类的属性 / 113
    - 4.5.7 新旧版本属性运行时的区别 / 113
  - 4.6 方法 / 114
    - 4.6.1 方法通用格式 / 114
    - 4.6.2 方法的调用 / 114
  - 4.7 继承 / 115
    - 4.7.1 类的继承关系 / 115
    - 4.7.2 继承父类的实例变量 / 116
    - 4.7.3 继承父类的方法 / 116
    - 4.7.4 哪些类需要被继承 / 117
    - 4.7.5 对象的合成 / 118
  - 4.8 方法重写 / 119
    - 4.8.1 方法重写的规则 / 119
    - 4.8.2 方法重写的使用 / 120
  - 4.9 方法重载 / 121
    - 4.9.1 方法重载的规则 / 121
    - 4.9.2 方法重载的使用 / 121
    - 4.9.3 调用还是重载 / 123
    - 4.9.4 重载的类型 / 123
  - 4.10 类的扩展 / 125
    - 4.10.1 类别的用法 / 125
    - 4.10.2 延伸的用法 / 127
  - 4.11 异常处理 / 128
    - 4.11.1 如何启用异常处理 / 128
    - 4.11.2 如何捕捉不同类型的异常 / 128
    - 4.11.3 如何抛出异常 / 129
  - 4.12 小结 / 129
- 第 5 章 对象——构建应用程序的重要“活体” / 130**
- 5.1 理解对象 / 130
    - 5.1.1 对象和根类的关系 / 130
    - 5.1.2 对象如何构建程序 / 131
    - 5.1.3 对象动态类型 / 132
    - 5.1.4 对象的生命周期 / 133
  - 5.2 创建对象 / 137
    - 5.2.1 对象分配处理的机制 / 137
    - 5.2.2 对象初始化方法的原型 init / 137
    - 5.2.3 初始化方法的返回值 / 139
    - 5.2.4 init 方法的实现 / 140
    - 5.2.5 多个初始化方法和指定初始化方法 / 141
    - 5.2.6 使用 dealloc 方法 / 143
    - 5.2.7 类工厂方法 / 144
  - 5.3 对象的所有权 / 144
    - 5.3.1 对象所有权策略有哪些 / 145

- 5.3.2 保留计数的处理机制 / 145
- 5.3.3 自动释放所有权 / 146
- 5.3.4 共享对象的有效性 / 147
- 5.3.5 如何获取所有权 / 148
- 5.4 回收对象 / 149
  - 5.4.1 dealloc 方法的实现 / 149
  - 5.4.2 通过引用返回的对象 / 149
  - 5.4.3 保留循环的处理机制 / 150
  - 5.4.4 对象的弱引用 / 150
  - 5.4.5 资源的有效管理 / 151
- 5.5 应用对象 / 152
  - 5.5.1 验证对象的功能 / 152
  - 5.5.2 比较对象 / 153
  - 5.5.3 复制对象 / 153
- 5.6 小结 / 154
- 第 6 章 消息和协议——对象之间的通信方式 / 155**
  - 6.1 认识消息 / 155
    - 6.1.1 消息的基本语法 / 155
    - 6.1.2 消息处理机制 / 157
    - 6.1.3 获得方法地址 / 158
    - 6.1.4 使用隐藏的参数 / 159
  - 6.2 消息发送 / 159
    - 6.2.1 向 nil 发送消息 / 160
    - 6.2.2 向自己发送消息 / 160
    - 6.2.3 通过发送消息调用父类方法 / 161
  - 6.3 消息转发 / 163
    - 6.3.1 消息转发处理机制 / 164
    - 6.3.2 与多重继承关系 / 165
    - 6.3.3 与类继承关系 / 166
    - 6.3.4 与消息代理对象关系 / 167
  - 6.4 认识协议 / 167
    - 6.4.1 协议的分类 / 167
    - 6.4.2 协议对象 / 169
    - 6.4.3 预定义接口 / 169
    - 6.4.4 如何使用预定义方法 / 170
    - 6.4.5 为匿名对象声明方法 / 171
  - 6.5 应用协议 / 172
    - 6.5.1 如何采用一个协议 / 172
    - 6.5.2 如何服从一个协议 / 172
    - 6.5.3 协议类型校验处理机制 / 173
    - 6.5.4 如何实现协议嵌套协议 / 173
    - 6.5.5 如何引用其他协议 / 174
  - 6.6 小结 / 175
- 第 7 章 Foundation 框架——提供基本的系统服务 / 176**
  - 7.1 认识 Foundation 框架 / 176
    - 7.1.1 Foundation 类层次结构 / 176
    - 7.1.2 与 Core Foundation 框架的区别 / 180
    - 7.1.3 如何引用及查询 Foundation 框架信息 / 181
    - 7.1.4 对象的可变性和不变性 / 183
  - 7.2 创建和使用值对象 / 184
    - 7.2.1 创建值对象 / 184
    - 7.2.2 字符串和 NSString 字面常量 / 185
    - 7.2.3 NSNumber 字面常量 / 185
    - 7.2.4 日期和时间 / 186
  - 7.3 创建和使用集 / 186

- 7.3.1 将对象按某种顺序储存在数组中 / 187
- 7.3.2 将键值对储存在字典中 / 189
- 7.3.3 将无序对象储存在集合中 / 190
- 7.4 Foundation 框架常用类的使用方法详解 / 190
  - 7.4.1 数字类型 / 191
  - 7.4.2 字符串类型 / 193
  - 7.4.3 数组类型 / 197
  - 7.4.4 字典类型 / 205
  - 7.4.5 日期类型 / 207
- 7.5 小结 / 209
- 第 8 章 内存管理——应用程序高效运行的基础 / 210**
  - 8.1 内存管理基础知识 / 210
    - 8.1.1 Cocoa 的引用计数机制 / 210
    - 8.1.2 内存管理规则 / 211
    - 8.1.3 内存管理应用实例 / 211
    - 8.1.4 内存管理混乱的原因及解决方法 / 212
  - 8.2 存取方法 / 214
    - 8.2.1 声明存取方法 / 214
    - 8.2.2 实现存取方法 / 214
    - 8.2.3 存取方法的使用 / 216
    - 8.2.4 实现重置方法 / 217
    - 8.2.5 应用存取方法常见错误 / 218
  - 8.3 自动释放池 / 218
    - 8.3.1 自动释放池工作原理 / 219
    - 8.3.2 非 Application Kit 程序中的自动释放池 / 219
    - 8.3.3 自动释放池和线程 / 220
    - 8.3.4 作用域和嵌套自动释放池的关系 / 220
    - 8.3.5 所有权策略 / 221
    - 8.3.6 如何实现垃圾回收 / 221
  - 8.4 内存中复制的处理机制 / 222
    - 8.4.1 深复制和浅复制 / 222
    - 8.4.2 独立副本 / 223
    - 8.4.3 使用 alloc 和 init 方式复制 / 223
    - 8.4.4 使用 NSCopyObject 函数复制 / 224
    - 8.4.5 可变对象和不可变对象的复制 / 225
    - 8.4.6 值对象和复制 / 226
  - 8.5 Nib 对象的内存管理 / 226
    - 8.5.1 插座对象实现的内存管理机制 / 227
    - 8.5.2 Nib 文件实现的内存管理机制 / 227
  - 8.6 小结 / 228
- 第三部分 基础篇**
- 第 9 章 探究 iOS 应用程序的核心 / 230**
  - 9.1 iOS 应用程序核心架构 / 230
    - 9.1.1 iOS 应用程序的生命周期 / 230
    - 9.1.2 iOS 应用程序的主函数 / 230
    - 9.1.3 应用程序的委托 / 232
    - 9.1.4 主 Nib 文件 / 232
    - 9.1.5 事件处理周期 / 232
    - 9.1.6 应用程序的基本设置模式 / 234

- 9.2 iOS 应用程序的核心对象 / 235
    - 9.2.1 iOS 应用程序的常见对象 / 235
    - 9.2.2 数据模型的定义 / 237
    - 9.2.3 构建用户界面 / 239
  - 9.3 iOS 应用程序包 / 241
    - 9.3.1 典型的 iOS 应用程序捆绑包 / 242
    - 9.3.2 信息属性列表 / 243
    - 9.3.3 程序图标和启动图像 / 246
    - 9.3.4 Nib 文件 / 246
  - 9.4 iOS 应用程序关键任务的处理机制 / 247
    - 9.4.1 初始化和终止 / 247
    - 9.4.2 响应中断 / 248
    - 9.4.3 低内存警告 / 249
  - 9.5 iOS 应用程序的行为定制 / 250
    - 9.5.1 以景观模式启动 / 250
    - 9.5.2 和其他应用程序的通信 / 251
    - 9.5.3 定制的 URL 模式 / 251
    - 9.5.4 应用程序的偏好设置 / 253
    - 9.5.5 关闭屏幕锁定 / 254
  - 9.6 小结 / 254
- 第 10 章 多状态和多任务——iOS 功能日趋增强的表现 / 255**
- 10.1 应用程序的状态 / 255
    - 10.1.1 状态切换遵循的原则 / 255
    - 10.1.2 应用程序的状态及切换路径 / 256
  - 10.2 应用程序启动周期 / 257
    - 10.2.1 加载进入前台 / 257
    - 10.2.2 加载进入后台 / 258
    - 10.2.3 主函数 / 260
    - 10.2.4 启动处理的机制 / 260
  - 10.3 响应中断 / 261
    - 10.3.1 基于警告的中断处理机制 / 261
    - 10.3.2 中断应激处理 / 262
    - 10.3.3 通话时用户界面的调整处理 / 263
  - 10.4 前台与后台之间的切换机制 / 263
    - 10.4.1 从前台切换到后台的流程 / 263
    - 10.4.2 转换后台时应激处理 / 264
    - 10.4.3 后台运行时的内存情况 / 265
    - 10.4.4 从后台回转到前台的流程 / 265
    - 10.4.5 在唤醒时处理排队的通知 / 266
    - 10.4.6 应用程序的终止条件 / 266
  - 10.5 主运行循环 / 267
    - 10.5.1 主运行循环的运行机制 / 267
    - 10.5.2 iOS 应用程序中的事件 / 268
  - 10.6 后台执行和多任务处理 / 268
    - 10.6.1 判断设备系统多任务是否可用 / 269
    - 10.6.2 在后台执行有限长度的任务 / 269
    - 10.6.3 调度本地通知的递送 / 270
    - 10.6.4 允许在后台运行的任务 / 271

- 10.6.5 选择退出后台的执行 / 271
- 10.7 小结 / 271
- 第 11 章 视图——iOS 应用程序交互的基础 / 272
  - 11.1 窗口、视图、视图控制器 / 272
    - 11.1.1 三者之间的关系 / 272
    - 11.1.2 窗口 / 273
    - 11.1.3 视图 / 273
    - 11.1.4 视图控制器 / 274
  - 11.2 视图的几何属性特征 / 275
    - 11.2.1 视图坐标系 / 275
    - 11.2.2 边框、边界和中心的关系 / 275
    - 11.2.3 坐标系变换 / 277
  - 11.3 视图架构处理 / 277
    - 11.3.1 视图交互模型 / 278
    - 11.3.2 视图渲染架构 / 279
    - 11.3.3 改变视图的层 / 280
    - 11.3.4 内容模式与比例缩放 / 280
    - 11.3.5 自动尺寸调整行为 / 281
  - 11.4 视图的创建和管理 / 283
    - 11.4.1 创建视图对象 / 284
    - 11.4.2 视图的标识和命名 / 284
    - 11.4.3 子视图的添加和移除 / 285
    - 11.4.4 视图层次中的坐标转换 / 287
    - 11.4.5 视图的查询 / 288
  - 11.5 在运行时修改视图 / 289
    - 11.5.1 实现视图动画 / 290
    - 11.5.2 配置动画的参数 / 290
    - 11.5.3 配置动画的委托 / 291
    - 11.5.4 响应布局的变化 / 291
    - 11.5.5 重画视图的内容 / 292
    - 11.5.6 隐藏视图 / 293
  - 11.6 定制视图对象 / 293
    - 11.6.1 初始化定制视图 / 293
    - 11.6.2 绘制视图内容 / 294
    - 11.6.3 响应事件 / 295
    - 11.6.4 视图对象的清理 / 295
  - 11.7 小结 / 296
- 第 12 章 UIKit 框架——创建基于触摸的用户界面 / 297
  - 12.1 UIKit 标准视图的分类 / 297
  - 12.2 显示视图 / 298
    - 12.2.1 标签 / 298
    - 12.2.2 图片视图 / 299
    - 12.2.3 进度条视图 / 300
    - 12.2.4 等待视图 / 301
  - 12.3 控件 / 302
    - 12.3.1 按钮 / 302
    - 12.3.2 文本框 / 304
    - 12.3.3 滑块 / 305
    - 12.3.4 切换开关 / 306
  - 12.4 导航视图 / 307
    - 12.4.1 导航栏 / 307
    - 12.4.2 标签栏 / 309
  - 12.5 警告视图和动作表单 / 310
    - 12.5.1 警告视图 / 310
    - 12.5.2 动作表单 / 311
  - 12.6 文本和 Web 视图 / 313
    - 12.6.1 文本视图 / 313
    - 12.6.2 Web 视图 / 314
  - 12.7 容器视图 / 315
    - 12.7.1 表视图 / 315

- 12.7.2 滚动视图 / 319
- 12.7.3 工具栏 / 321
- 12.8 其他常用类型 / 322
  - 12.8.1 分页控件 / 322
  - 12.8.2 搜索栏 / 324
- 12.9 小结 / 328
- 第 13 章 视图控制器——视图的幕后操纵者 / 329**
  - 13.1 视图控制器基础知识 / 329
    - 13.1.1 视图控制器的功能 / 329
    - 13.1.2 视图控制器的管理机制 / 330
    - 13.1.3 视图控制器的分类 / 331
    - 13.1.4 视图控制器的内容多种展示方式 / 333
    - 13.1.5 多种视图控制器混合应用 / 334
  - 13.2 视图控制器的生命周期 / 335
    - 13.2.1 视图控制器的初始化 / 336
    - 13.2.2 视图的加载和卸载 / 336
  - 13.3 标准视图控制器 / 338
    - 13.3.1 标准视图控制器的功能及使用方法 / 339
    - 13.3.2 标准视图控制器的应用 / 340
  - 13.4 分割视图控制器 / 342
    - 13.4.1 分割视图控制器构建原理及功能 / 342
    - 13.4.2 分割视图控制器的应用 / 343
  - 13.5 导航视图控制器 / 344
    - 13.5.1 导航视图控制器构建原理及功能 / 344
    - 13.5.2 导航视图控制器的应用 / 346
  - 13.6 选项卡视图控制器 / 347
    - 13.6.1 选项卡视图控制器构建原理及功能 / 347
    - 13.6.2 选项卡视图控制器的应用 / 349
  - 13.7 翻页视图控制器 / 349
    - 13.7.1 翻页视图控制器构建原理及功能 / 349
    - 13.7.2 翻页视图控制器的应用 / 351
  - 13.8 小结 / 351
- 第 14 章 事件——应用程序的驱动力 / 352**
  - 14.1 iOS 系统中的事件 / 352
    - 14.1.1 事件是如何驱动应用程序的 / 352
    - 14.1.2 什么是触摸事件 / 352
    - 14.1.3 什么是运动事件 / 353
    - 14.1.4 事件和触摸 / 354
    - 14.1.5 事件的传递 / 356
    - 14.1.6 响应者对象和响应者链 / 356
    - 14.1.7 调整事件的传递 / 357
  - 14.2 常见手势的处理实例 / 358
    - 14.2.1 触摸事件处理方法 / 358
    - 14.2.2 单个和多个触碰手势处理方法 / 359
    - 14.2.3 检测碰擦手势 / 360
    - 14.2.4 处理复杂的多点触摸序列 / 361

- 14.2.5 触摸事件处理技巧 / 362
- 14.3 键盘管理 / 363
  - 14.3.1 接收键盘通告 / 363
  - 14.3.2 显示键盘 / 365
  - 14.3.3 取消键盘 / 365
  - 14.3.4 移动键盘下面的内容 / 365
- 14.4 小结 / 368
- 第 15 章 通知——消息的多样化展示 / 369
  - 15.1 通知的实现原理 / 369
    - 15.1.1 本地通知 / 369
    - 15.1.2 推送通知 / 370
    - 15.1.3 通知的应用场景 / 371
  - 15.2 通知的相关事务 / 373
    - 15.2.1 自定义警告声音 / 373
    - 15.2.2 创建和调度本地通知 / 373
    - 15.2.3 接收远程通知 / 375
    - 15.2.4 如何处理通知 / 377
  - 15.3 推送通知服务 / 380
    - 15.3.1 推送通知和路径 / 380
    - 15.3.2 服务的反馈和质量 / 381
    - 15.3.3 推送通知的安全处理机制 / 381
    - 15.3.4 通知的负载处理 / 385
  - 15.4 推送通知的配置和开发 / 386
    - 15.4.1 沙箱环境和产品环境 / 386
    - 15.4.2 配置处理流程 / 387
  - 15.5 实现推送通知服务的通信功能 / 389
    - 15.5.1 普通提供的通信功能 / 389
    - 15.5.2 二进制接口和通知的格式 / 390
    - 15.5.3 服务的反馈 / 393
  - 15.6 小结 / 393
- 第 16 章 Core Data 框架——管理应用程序的数据模型 / 394
  - 16.1 认识 Core Data / 394
    - 16.1.1 初窥 Core Data 特性 / 394
    - 16.1.2 Core Data 数据管理机制 / 395
    - 16.1.3 探究 Core Data 的本质 / 396
  - 16.2 Core Data 堆栈配置 / 396
    - 16.2.1 托管对象和上下文 / 398
    - 16.2.2 读取请求 / 399
    - 16.2.3 持久化存储协调者 / 400
    - 16.2.4 持久化存储 / 401
    - 16.2.5 持久化文档 / 402
    - 16.2.6 托管对象模型 / 402
  - 16.3 Core Data 模块的运作机制及基础类 / 402
    - 16.3.1 NSManagedObjectContext 类 / 403
    - 16.3.2 NSManagedObject 类 / 404
    - 16.3.3 NSManagedObjectContext 类 / 405
    - 16.3.4 NSPersistentStoreCoordinator 类 / 405
    - 16.3.5 NSPersistentDocument 类 / 406
    - 16.3.6 NSFetchedRequest 类 / 406

- 16.4 Core Data 堆栈访问技术 / 407
    - 16.4.1 创建一个新的托管对象上下文 / 407
    - 16.4.2 读取托管对象模型和实体 / 407
    - 16.4.3 添加持久化存储 / 407
  - 16.5 托管对象的管理 / 408
    - 16.5.1 读取托管对象 / 408
    - 16.5.2 读取特定的属性值 / 411
    - 16.5.3 创建托管对象 / 412
    - 16.5.4 保存托管对象 / 413
    - 16.5.5 删除托管对象 / 413
  - 16.6 小结 / 413
- 第 17 章 音频和视频——强大的多媒体功能支持 / 414**
- 17.1 初识 iOS 多媒体框架 / 414
    - 17.1.1 iOS 声音处理工具 / 414
    - 17.1.2 Core Audio 框架 / 414
    - 17.1.3 音频硬件编解码 / 415
    - 17.1.4 iOS 支持的音频回放和录制格式 / 415
    - 17.1.5 Core Audio 音频会话接口 / 416
    - 17.1.6 iOS 系统支持的音频单元 / 417
  - 17.2 录制音频 / 417
    - 17.2.1 使用 AVAudioRecorder 类进行录制 / 417
    - 17.2.2 用音频队列服务进行录制 / 418
    - 17.2.3 音频中断处理 / 419
  - 17.3 播放音频 / 419
    - 17.3.1 通过 iPod 媒体库访问接口播放媒体项 / 419
    - 17.3.2 使用系统声音服务播放短声音及触发振动 / 420
    - 17.3.3 通过 AVAudioPlayer 类轻松播放声音 / 421
    - 17.3.4 用音频队列服务播放和控制声音 / 423
    - 17.3.5 使用 OpenAL 播放和定位声音 / 426
    - 17.3.6 解析音频流 / 426
  - 17.4 iPhone/iPad 音频的最佳实践 / 426
    - 17.4.1 操作音频的贴士 / 426
    - 17.4.2 iOS 中偏好的音频格式 / 427
  - 17.5 iOS 中的视频 / 428
    - 17.5.1 录制视频 / 428
    - 17.5.2 播放视频 / 428
  - 17.6 小结 / 429
- 第 18 章 设备特性——支持多种功能的应用 / 430**
- 18.1 识别可用的硬件特性 / 430
  - 18.2 External Accessory 框架的管道机制 / 430
    - 18.2.1 声明应用程序支持的协议 / 431
    - 18.2.2 在运行时连接配件 / 431
    - 18.2.3 处理和流相关的数据 / 432
    - 18.2.4 监控与配件有关的事件 / 433
  - 18.3 访问加速计事件 / 433