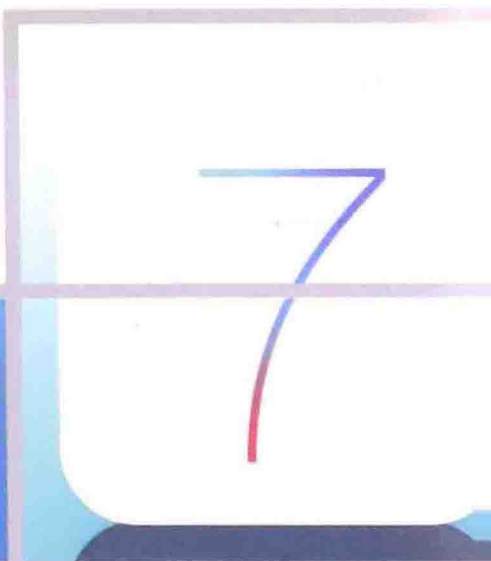


[美] Rob Napier Mugunth Kumar 著
美团移动 译

iOS编程实战

- iOS开发进阶首选
- 深入挖掘iOS高级特性与开发技巧
- 挑战编程极限，打造非凡应用

```
in PBXFrameworksBuildPhase section 7F  
0ECCF15F5F66004A5BFF77 Frameworks 7F  
# PBXFrameworksBuildPhase  
IDActionMask = 2147483647  
v = 7  
131D0615FF5B66004A5BFF77 UIKit.framework in Frameworks 7F  
131D0615FF5B66004A5BFF77 UIKit.framework in Frameworks 7F  
131D0615FF5B66004A5BFF77 UIKit.framework in Frameworks 7F  
Only the default build phase is shown.
```



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

iOS编程实战

[美] Rob Napier Mugunth Kumar 著

美团移动 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

iOS编程实战 / (美) 纳皮尔 (Napier, R.), (美) 库玛 (Kumar, M.) 著; 美团移动译. -- 北京: 人民邮电出版社, 2014.9

(图灵程序设计丛书)

ISBN 978-7-115-36803-4

I. ①i… II. ①纳… ②库… ③美… 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(2014)第191743号

内 容 提 要

本书是最受开发者喜爱的 iOS 进阶图书。它包含大量代码示例, 主线是围绕如何设计、编写和维护优秀的 iOS 应用。开发者可从本书学到大量关于设计模式、编写可重用代码以及语法与新框架的知识。

相对上一版, 新版进行了大幅修订, 新增 6 章阐述 iOS 7 新特性, 并对大部分内容进行了更新, 涵盖了 iOS 7 大部分新增特性, 包括新的后台操作 (第 11 章)、Core Bluetooth (第 13 章)、UIKit 动力学 (第 19 章) 以及 TextKit (第 21 章)。我们提供了如何处理新的扁平化 UI 的指南 (第 2 章), 还新增了一章开发者不太常见但相当实用的“小技巧”(第 3 章)。

本书适合 iOS 移动开发人员。

-
- ◆ 著 [美] Rob Napier Mugunth Kumar
 - 译 美团移动
 - 责任编辑 朱 巍
 - 责任印制 焦志伟
 - ◆ 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: 800×1000 1/16
 - 印张: 25.5
 - 字数: 603千字 2014年9月第1版
 - 印数: 1-3 500册 2014年9月河北第1次印刷
 - 著作权合同登记号 图字: 01-2014-1133号
-

定价: 79.00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

从某种程度上说，iOS 7 是从 SDK 随着 iPhone OS 2 发布到现在 iOS 发生过的最大的变化。人们在新闻和博客中讨论新的扁平化用户界面的各个方面，及其对应用开发者和用户的意义。可以说，从没有一次 iOS 的升级会使得这么多的开发者重新设计 UI。

但是从另外的角度看，iOS 7 几乎可以从 iOS 6 无缝升级。比起 iOS 4 在多任务上的变化，iOS 7 只需要对应用做很小的改动，尤其是开发者使用标准 UI 或者完全自定义 UI 的情况。对于这两种极端情况，UI 的变化要么是自动完成的，要么压根儿跟开发者没关系。

不过，对所有的开发者来说，iOS 还是带来了变化。有很多管理后台操作的方法，但是后台运行的规则甚至比以前更严格了。UIKit 动力学意味着更灵活的动画，不过实现起来不简单。TextKit 为文本布局带来了令人难以置信的特性，也伴随着令人发疯的限制和 bug。iOS 7 是大杂烩，既有美好也有挫败。不过你得学习 iOS 7，因为用户很快就会升级。

如果你准备好了去探索最新的苹果系统，准备好了挑战应用的极限，那么本书会助你一臂之力。

读者对象

这并不是一本入门书。其他一些书会教你 Objective-C 并一步步指导你学习 Interface Builder。不过本书假定你已经拥有一些 iOS 开发经验。可能是自学的，或者上过培训班，没准已经有一个应用即将完工只是没有上架而已。对于此类读者，如果你打算学习更深入的内容、最佳实践以及作者源自真实工程的开发经验，那你就找对书了。

这本书并不是示例的简单堆砌，它包含大量代码，不过主线还是围绕如何设计、编写和维护优秀的 iOS 应用。本书不仅会教你怎么做，并且会剖析这样做的原因。你会学到很多关于设计模式、编写可重用代码以及语法与新框架的知识。

本书内容

iOS 平台总是向前发展，本书也一样。书中大部分示例需要至少 iOS 6 才能运行，有些需要 iOS 7。所有示例都启用了自动引用计数（ARC）、自动属性合成和对象字面量。除了很少几处外，本书不会讨论向后兼容。如果你的代码过于庞大，必须要向后兼容，你可能知道如何处理。本书主旨是通过最好的特性来创造最佳应用。

本书专注于 iPhone 5、iPad 3 和更新的型号。大部分主题对其他 iOS 设备也适用。第 15 章讲了如何处理平台间的差异。

新版内容

本版涵盖了 iOS 7 大部分的新增特性，包括新的后台操作（第 11 章）、Core Bluetooth（第 13 章）、UIKit 动力学（第 19 章）以及 TextKit（第 21 章）。我们提供了如何处理新的扁平化 UI 的指南（第 2 章），还新增了一章你可能不知道的“小技巧”（第 3 章）。

本书专注于 iOS 7 中最有价值的信息。前几版的有些章节被移到了网站上（iosptl.com）。读者可以在那里找到关于常见的 Objective-C 实践、定位服务、错误处理等内容的章节。

本书结构

iOS 提供了非常丰富的工具，既有 UIKit 这样的高层框架，也有 Core Text 这样的低层工具。有时候，同一个目标可以通过多种方式来达成。作为开发人员，如何找到最合适的工具呢？

本书既考虑了日常开发需求，也考虑了特定的用途，能够帮你作出正确的选择。学完本书，你会明白每个框架存在的价值、框架之间的相互关系，以及什么时候选用哪一个框架。最终，你会知道哪个框架最适合解决哪一类问题。

本书分四部分，从最常用的工具一直讲到最强大的工具。这一版新增的章会在前面用“新增”字样标识，而经过大范围更新的章会用“更新”字样标识。

第一部分：全新功能

如果你熟悉 iOS 6，这部分可以让你快速了解 iOS 7 的新特性。

- （新增）第 1 章“全新的系统”——iOS 7 增加了大量新特性，本章将带你快速概览有哪些内容。
- （新增）第 2 章“世界是平的：新的 UI 范式”——iOS 7 对 iOS 应用的外观和行为做了巨大的改变。本章将介绍迁移所需的新模式和设计语言。

第二部分：充分利用日常工具

作为一名 iOS 开发人员，你应该掌握很多常用工具，比如通知、表视图和动画图层。不过要想发挥它们的全部潜力，就要熟悉它们。在这一部分，我们将学到 Cocoa 开发的最佳实践。

- （新增）第 3 章“你可能不知道的”——即使你是一位有经验的开发者，你可能并不熟悉 Cocoa 的一些小特性和技巧。本章介绍作者根据多年 iOS 开发经验总结的最佳实践，以及 Cocoa 一些不那么常见的方面。
- （更新）第 4 章“故事板及自定义切换效果”——故事板仍然会使一些熟悉 nib 文件的开发人员感到费解。你在这里将会学到如何使用故事板来提升应用。
- （更新）第 5 章“掌握集合视图”——集合视图正在逐步替代表视图，成为开发人员偏爱的布局控件。即使对于表格类布局，集合视图也提供了极大的灵活性，要想开发出迷人的应用，你应该理解这一点。本章会教你如何掌握这一重要工具。
- （新增）第 6 章“使用自动布局”——如果 WWDC 2013 有什么核心的信息，那么一定是：使用自动布局。会议期间几乎所有的 UIKit 会话都在反复强调这一点。由于 Xcode 4 中诸多 Interface

Builder 问题，开发者可能尽量避免使用自动布局。Xcode 5 极大地改进了对自动布局的支持。不管你是喜欢使用约束，还是渴望回归 springs & struts，你都不该错过最新的自动布局。

- ❑ 第 7 章“更完善的自定义绘图”——很多新开发者都对自定义绘图退避三舍，但它却是快速创建美观用户界面的关键。这一章将探究 UIKit 和 Core Graphics 中有关绘图的功能，告诉大家怎么才能做到既快又美。
- ❑ 第 8 章“Core Animation”——iOS 设备对动画的支持是无与伦比的。借助强大的 GPU 和高度优化的 Core Animation，你可以创建直观又吸引人的界面。在这一章中，我们会介绍一些基础知识以及动画的原理。
- ❑ （更新）第 9 章“多任务”——多任务是许多应用程序的重要部分，这一章将介绍如何同时使用操作和 GCD 执行多任务。

第三部分：选择工具

- ❑ 第 10 章“创建（Core）Foundation 框架”——说到 iOS 中最强大的框架，你能想到的 Core 框架可能会有 Core Graphics、Core Animation、Core Text，但它们都是基于 Core Foundation 框架的。在这一章中，我们学习如何使用 Core Foundation 数据类型，以便充分利用 iOS 提供的功能。
- ❑ （更新）第 11 章“幕后制作：后台处理”——iOS 7 的后台处理又灵活了很多，但是，要想充分利用这些新变化，你得遵循一些新规则。本章带你深入学习新的 NSURLSession，以及如何最好地实现状态恢复。
- ❑ 第 12 章“使用 REST 服务”——基于 REST 的服务是现代应用程序的核心，这一章将教会你在 iOS 中最好地实现它们。
- ❑ （新增）第 13 章“充分利用蓝牙设备”——苹果一直在加强 iOS 与其他设备创建 ad hoc 网络的能力。这使得开发全新的应用成为可能：从更好的游戏到微定位服务，再到更方便的文件共享。加快创新，投入这个全新的市场吧！
- ❑ 第 14 章“通过安全服务巩固系统安全”——用户安全和保护隐私永远是第一位的。这一章会介绍如何通过钥匙串、证书和密码保护应用和用户数据不会被盗用。
- ❑ （更新）第 15 章“在多个苹果平台和设备及 64 位体系结构上运行应用”——iOS 家族人丁兴旺，不仅有了 iPod touch、iPhone、iPad、Apple TV，而且新机型仍会不断涌现。目前还无法一次编写随处运行。为了保证应用在任何平台上都表现卓越，本章将讨论如何基于硬件和平台调整应用。
- ❑ 第 16 章“国际化和本地化”——虽然你现在可能只想关注某个国家的市场，但让应用明天能够顺利走向世界也只需做一点点工作。本章会告诉你如何不影响当前开发，又能减少未来的麻烦和成本。
- ❑ 第 17 章“调试”——要是每个应用第一次就能完美运行该有多好。幸好，Xcode 和 LLDB 提供了很多能帮助你抓住狡猾 bug 的工具。你会学到很多高级的内容，了解实际开发中如何处理错误。
- ❑ （更新）第 18 章“性能调优”——高性能可以让应用脱颖而出。优化 CPU 和内存性能非常重要，不过你也需要优化电池以及网络使用。苹果公司提供了 Instruments 这个强大的工具来解决这些问题。你会学到如何使用 Instruments 来找到瓶颈，以及如何在找到问题后改善性能。

第四部分：超越极限

这一部分是全书最精彩的内容。你已经学到了基础知识，掌握了基本技能。现在该使用高级工具来超越极限了。这一部分将带你深入地了解 iOS。

- ❑（新增）第 19 章“近乎物理效果：UIKit 动力学”——苹果一直致力于让动态的动画界面更容易实现。UIKit 动力学是其最新杰作，为 UIKit 带来了“类物理效果”的引擎。这个工具很强大，同时要用好也很难。本章学习如何使用。
- ❑（新增）第 20 章“魔幻的自定义过渡”——WWDC 2013 最绚丽的演示程序就是关于自定义过渡效果的。忘了“推入”，来学习如何创建动态和交互式的过渡效果吧。
- ❑（更新）第 21 章“精妙的文本布局”——iOS 7 以文本为中心的 UI 需要在字体处理和文本布局的细节上投入大量精力。TextKit 带来了很多新特性，从动态字体到排除路径，还带来了 bug 和令人抓狂的限制。无论怎么处理文本，首先你得掌握属性化字符串。本章介绍这些强大的数据结构的前方，以及如何用 TextKit 充分利用这些数据结构。
- ❑ 第 22 章“Cocoa 的大招：键值编码和观察”——苹果的许多强大框架都是依靠 KVO (Key-Value Observing) 来维护性能和灵活性的。你会学到如何利用灵活性和 KVO 的速度，以及让它如此透明的诀窍。
- ❑（新增）第 23 章“超越队列：GCD 高级功能”——分派队列是非常强大的工具，已经成了很多应用的重要组成部分。但是除了队列以外，GCD 还有别的东西。本章介绍信号量、分派组还有非常强大的分派数据和分配 IO 这些工具。
- ❑ 第 24 章“深度解析 Objective-C”——这一章致力于揭开 Objective-C 背后的秘密，包括如何使用 Objective-C 运行时直接动态地修改类和方法、如何通过 Objective-C 函数调用 C 方法，以及如何通过系统来扩展程序。

以上各章可以跳读，除了需要 Core Foundation 数据对象（特别是 Core Graphics、Core Animation）的几章，其他章都是相互独立的。关于 Core Foundation 的内容，最终会归总到第 10 章“创建 Core Foundation 框架”。

阅读条件

本书所有示例都是用 Mac OS X 10.8 上的 Xcode 5 以及 iOS 7 开发的。你需要一个苹果开发人员账户来访问大部分工具和文档，并且需要一个开发人员许可证来运行 iOS 设备上的应用程序。对此，请参考<http://developer.apple.com/programs/ios>并注册账号。

本书中大部分示例可以在 Xcode 5 的 iOS 模拟器中运行。使用 iOS 模拟器就不需要苹果开发人员许可证了。

苹果文档

苹果公司在自己的网站上和 Xcode 中提供了大量文档。这些文档的 URL 地址变动很频繁而且非常长。本书会使用标题而不是 URL 来引用这些文档。如果想在 Xcode 中寻找文档，请按下 Cmd-Option-? 快捷键或点击 Help→Documentation and API Reference。并在 Documentation Organizer 窗口中点击搜索

图标, 输入文档的标题, 并从搜索结果中选择文档。可以参考图 0-1 中搜索 Coding Guidelines for Cocoa 的示例。

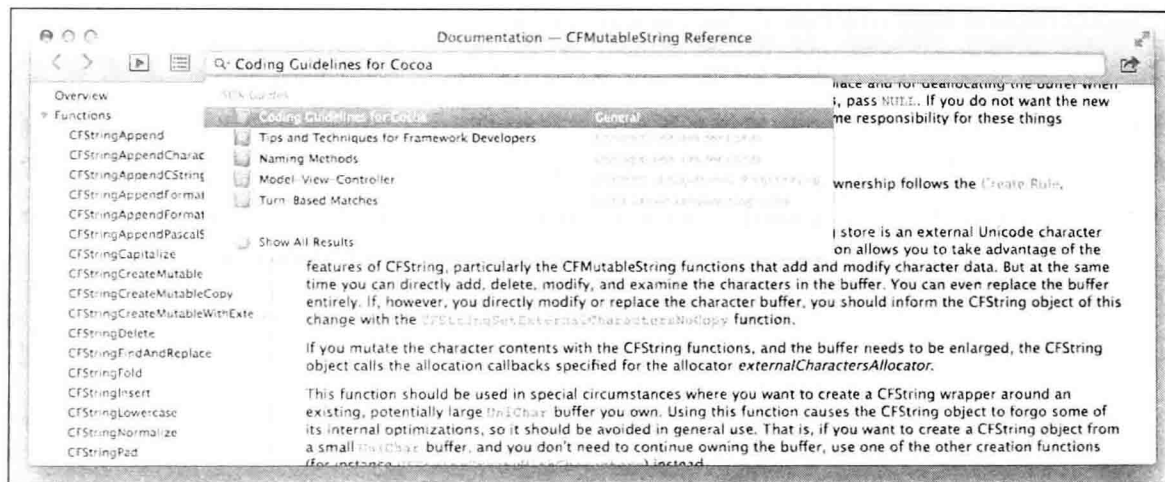


图 0-1 搜索 Coding Guidelines for Cocoa

如果想在苹果官方网站查找文档, 可以访问<http://developer.apple.com>, 点击 Member Center, 并登录。选择 iOS Dev Center, 并在搜索框中输入文档的标题。

在线文档与 Xcode 文档是相同的。你可能会接收到 iOS 和 Mac 两个平台的结果, 请阅读 iOS 版。很多 iOS 文档是 Mac 版的副本, 偶尔会包含 iOS 不支持的函数调用或常量。本书会告诉你哪些功能在 iOS 上能用。

源代码

在学习本书示例的时候, 可以手工输入代码, 也可以使用本书附带的源代码文件。本书所有的源代码可以在<https://github.com/iosptl/ios7ptl>或www.wiley.com/go/ptl/ios7programming上下载得到^①。举个例子, 下载之后, 在第 21 章文件夹 SimpleLayout 工程的 CoreTextLabel.m 文件中可以看到如下代码:

CoreTextLabel.m (SimpleLayout)

```
- (id)initWithFrame:(CGRect)frame {
    if ((self = [super initWithFrame:frame])) {
        CGAffineTransform
        transform = CGAffineTransformMakeScale(1, -1);
        CGAffineTransformTranslate(transform,
                                   0, -self.bounds.size.height);
        self.transform = transform;
        self.backgroundColor = [UIColor whiteColor];
    }
    return self;
}
```

^① 也可以访问图灵社区本书页面 (<http://www.ituring.com.cn/book/1340>) 下载。——编者注

本书中有些代码片段并不完整，其目的只是为了辅助上下文说明问题。要想查看完整代码，可在本书同步网站下载（www.wiley.com/go/ptl/ios7programming）。

勘误

虽然我们已经尽了最大努力，但错误在所难免。有些可能是因为内容有变化，有些可能是输入错误，有些可能是我们理解上有偏差。为了确保代码与时俱进，请参考<https://github.com/iosptl/ios7ptl>上的最新版本，以及博客中的相关文章。任何问题都可以发送给 robnapier@gmail.com 或 contact@mk.sg。

目 录

第一部分 全新功能

第 1 章 全新的系统.....2

1.1 新的 UI	2
1.2 UIKit Dynamics 和 Motion Effects	3
1.3 自定义过渡效果	4
1.4 新的多任务模式	4
1.5 Text Kit	5
1.6 动态字体	5
1.7 MapKit 集成	5
1.8 SpriteKit	5
1.9 LLVM 5	5
1.10 Xcode 5	7
1.10.1 nib 文件格式的变化	7
1.10.2 源代码控制集成	7
1.10.3 自动配置	7
1.10.4 对调试导航面板的改进	8
1.10.5 文档浏览器	8
1.10.6 Asset Catalog	9
1.10.7 测试导航面板	9
1.10.8 持续集成	10
1.10.9 Auto Layout 改进	10
1.10.10 iOS 模拟器	10
1.11 其他	10
1.12 小结	11
1.13 扩展阅读	11

第 2 章 世界是平的：新的 UI 范式.....12

2.1 清晰、依从和层次	12
2.2 动画、动画、动画	12
2.2.1 UIKit Dynamics	13

2.2.2 UIMotionEvent	13
2.3 着色	14
2.4 用半透明实现层次和上下文	14
2.5 动态字体	15
2.6 自定义过渡效果	16
2.7 把应用过渡（迁移）到 iOS 7	17
2.7.1 UIKit 变化	17
2.7.2 自定义设计	17
2.7.3 支持 iOS 6	17
2.8 小结	20
2.9 扩展阅读	20

第二部分 充分利用日常工具

第 3 章 你可能不知道的.....22

3.1 命名最佳实践	22
3.1.1 自动变量	22
3.1.2 方法	22
3.2 属性和实例变量最佳实践	23
3.3 分类	24
3.4 关联引用	26
3.5 弱引用容器	27
3.6 NSCache	28
3.7 NSURLComponents	28
3.8 CFStringTransform	29
3.9instancetype	30
3.10 Base64 和百分号编码	31
3.11 -[NSArray firstObject]	31
3.12 小结	31
3.13 扩展阅读	32

第4章 故事板及自定义切换效果	33
4.1 初识故事板	33
4.1.1 实例化故事板	34
4.1.2 加载故事板中的视图控制器	34
4.1.3 连线	34
4.1.4 使用故事板来实现表视图	36
4.2 自定义切换效果	37
4.2.1 优点	39
4.2.2 白璧微瑕——合并冲突	39
4.3 小结	39
4.4 扩展阅读	39
第5章 掌握集合视图	41
5.1 集合视图	41
5.1.1 类与协议	41
5.1.2 示例	42
5.2 用集合视图自定义布局实现高级定制	49
5.2.1 石工布局	50
5.2.2 封面浏览布局	55
5.3 小结	56
5.4 扩展阅读	56
第6章 使用自动布局	58
6.1 Xcode 4 的自动布局	58
6.2 了解自动布局	59
6.3 Xcode 5 中自动布局的新特性	59
6.3.1 在 Xcode 5 中使用自动布局	61
6.3.2 固有尺寸	62
6.3.3 固有尺寸和本地化	63
6.3.4 设计时和运行时布局	63
6.3.5 自动更新边框	64
6.3.6 顶部和底部布局引导	65
6.3.7 辅助编辑器中的布局预览	65
6.3.8 在设计时调试自动布局	65
6.3.9 在自动布局中使用滚动视图	66
6.3.10 使用自动布局和边框	66
6.3.11 可视格式化语言	66
6.3.12 可视格式化语言的缺点	67
6.3.13 调试布局错误	68
6.4 小结	69
6.5 扩展阅读	69
第7章 更完善的自定义绘图	71
7.1 iOS 的不同绘图系统	71
7.2 UIKit 和视图绘图周期	72
7.3 视图绘制与视图布局	73
7.4 自定义视图绘制	74
7.4.1 通过 UIKit 绘图	74
7.4.2 路径	75
7.4.3 理解坐标系	77
7.4.4 重新调整大小以及内容模式	79
7.4.5 变形	80
7.4.6 通过 Core Graphics 进行绘制	82
7.4.7 混用 UIKit 与 Core Graphics	85
7.4.8 管理图形上下文	85
7.5 优化 UIView 绘制	87
7.5.1 避免绘图	87
7.5.2 缓存与后台绘制	88
7.5.3 自定义绘图与预渲染	88
7.5.4 像素对齐与模糊文本	89
7.5.5 透明、不透明与隐藏	90
7.6 小结	90
7.7 扩展阅读	91
第8章 Core Animation	92
8.1 视图动画	92
8.2 管理用户交互	94
8.3 图层绘制	94
8.3.1 直接设置内容	96
8.3.2 实现 display 方法	97
8.3.3 自定义绘图	97
8.3.4 在自己的上下文中绘图	99
8.4 移动对象	99
8.4.1 隐式动画	100
8.4.2 显式动画	101
8.4.3 模型与表示	101
8.4.4 关于定时	103
8.5 三维动画	105
8.6 美化图层	108
8.7 用动作实现自动动画	108

8.8 为自定义属性添加动画	110
8.9 Core Animation 与线程	111
8.10 小结	111
8.11 扩展阅读	111
第 9 章 多任务	113
9.1 多任务和运行循环简介	113
9.2 以操作为中心的多任务开发	114
9.3 用 GCD 实现多任务	118
9.3.1 分派队列简介	119
9.3.2 用分派屏障创建同步点	120
9.3.3 分派组	121
9.4 小结	121
9.5 扩展阅读	121

第三部分 选择工具

第 10 章 创建 (Core) Foundation 框架	124
10.1 Core Foundation 类型	124
10.2 命名和内存管理	125
10.3 分配器	126
10.4 内省	126
10.5 字符串和数据	127
10.5.1 常量字符串	127
10.5.2 创建字符串	128
10.5.3 转换为 C 字符串	129
10.5.4 其他字符串操作符	131
10.5.5 字符串的支持存储	131
10.5.6 CFData	132
10.6 容器类型	132
10.6.1 CFArray	133
10.6.2 CFDictionary	133
10.6.3 CFSet 和 CFBag	134
10.6.4 其他容器类型	134
10.6.5 回调函数	134
10.7 自由桥接	136
10.8 小结	139
10.9 扩展阅读	139

第 11 章 幕后制作：后台处理	140
11.1 后台运行最佳实践：能力越大责任越大	140
11.2 iOS 7 中后台运行的重要变化	142
11.3 用 NSURLSession 访问网络	142
11.3.1 会话配置	143
11.3.2 任务	143
11.3.3 后台传输	144
11.4 周期性拉取和自适应多任务	146
11.5 后台唤醒	146
11.6 状态恢复系统	147
11.6.1 测试状态恢复系统	147
11.6.2 选择性加入	148
11.6.3 应用启动过程的变化	149
11.6.4 状态恢复标识符	149
11.6.5 状态编码器与状态解码器	149
11.6.6 表视图和集合视图	153
11.7 小结	154
11.8 扩展阅读	154
第 12 章 使用 REST 服务	156
12.1 REST 简介	157
12.2 选择数据交换格式	157
12.2.1 在 iOS 中解析 XML	157
12.2.2 在 iOS 中解析 JSON	158
12.2.3 XML 与 JSON	159
12.2.4 模型版本化	160
12.3 假想的 Web 服务	160
12.4 重要提醒	161
12.5 RESTfulEngine 架构 (iHotelApp 示例代码)	161
12.5.1 NSURLConnection 与第三方框架	161
12.5.2 创建 RESTfulEngine	162
12.5.3 使用访问令牌对 API 调用进行认证	165
12.5.4 在 RESTfulEngine.m 中覆盖相关方法以添加自定义认证头部	165

12.5.5	取消请求	166	13.5	使用蓝牙设备	192
12.5.6	请求响应	166	13.5.1	通过扫描寻找服务	192
12.5.7	对 JSON 数据进行键值编码	167	13.5.2	连接设备	194
12.5.8	列表页面的 JSON 对象与详 细页面的 JSON 对象	169	13.5.3	直接获取外围设备	194
12.5.9	嵌套 JSON 对象	169	13.5.4	发现服务	194
12.5.10	少即是多	171	13.5.5	发现特性	195
12.5.11	错误处理	171	13.6	创建自己的外围设备	197
12.5.12	本地化	173	13.6.1	广播服务	197
12.5.13	使用分类处理其他格式	173	13.6.2	常见场景	200
12.5.14	在 iOS 中提升性能的小技 巧	174	13.7	在后台运行	200
12.6	缓存	174	13.7.1	后台模式	200
12.7	需要离线支持的原因	174	13.7.2	电量考虑	200
12.8	缓存策略	175	13.7.3	状态保存和恢复	200
12.8.1	存储缓存	175	13.8	小结	201
12.8.2	缓存版本和失效	178	13.9	扩展阅读	201
12.9	数据模型缓存	178	第 14 章	通过安全服务巩固系统安全	203
12.10	缓存版本控制	182	14.1	理解 iOS 沙盒	203
12.11	创建内存缓存	183	14.2	保证网络通信的安全	204
12.11.1	为 AppCache 设计内存 缓存	184	14.2.1	证书工作原理	205
12.11.2	处理内存警告	185	14.2.2	检验证书的有效性	207
12.11.3	处理结束和进入后台通 知	186	14.2.3	判断证书的可信度	210
12.12	创建 URL 缓存	186	14.3	使用文件保护	211
12.12.1	过期模型	187	14.4	使用钥匙串	213
12.12.2	验证模型	187	14.5	使用加密	216
12.12.3	示例	187	14.5.1	AES 概要	217
12.12.4	用 URL 缓存来缓存图片	188	14.5.2	使用 PBKDF2 将密码转换 成密钥	217
12.13	小结	188	14.5.3	AES 模式和填充	219
12.14	扩展阅读	188	14.5.4	初始化向量	220
第 13 章	充分利用蓝牙设备	190	14.5.5	使用 HMAC 进行认证	221
13.1	蓝牙历史	190	14.5.6	错误的密码	222
13.2	为什么选择低功耗蓝牙	191	14.5.7	组合使用加密和压缩	222
13.3	蓝牙 SDK	191	14.6	小结	222
13.3.1	服务器	191	14.7	扩展阅读	222
13.3.2	客户端	191	第 15 章	在多个苹果平台和设备及 64 位体系结构上运行应用	224
13.4	类和协议	191	15.1	开发多平台应用	225
			15.1.1	可配置的目标设置: Base SDK 和 Deployment Target	225

15.1.2	支持多个 SDK 时的注意事项: 框架、类和方法	225
15.1.3	检查框架、类和方法的可用性	227
15.2	检测设备的功能	228
15.2.1	检测设备及判断功能	228
15.2.2	检测硬件和传感器	229
15.3	应用内发送 Email 和短信	233
15.4	支持新的 4 英寸设备族系	233
15.4.1	Cocoa 自动布局	235
15.4.2	代码中固化屏幕尺寸	235
15.4.3	iPhone 5s 和新的 64 位指令集	235
15.5	向 iOS 7 迁移	236
15.5.1	自动布局	236
15.5.2	支持 iOS 6	236
15.5.3	应用图标	237
15.5.4	无边界按钮	237
15.5.5	着色	237
15.5.6	图片更新	237
15.6	向 64 位体系结构迁移	238
15.6.1	数据溢出	238
15.6.2	序列化数据	239
15.6.3	针对 64 位体系结构的条件编译	239
15.7	UIRequiredDeviceCapabilities	239
15.8	小结	240
15.9	扩展阅读	240

第 16 章 国际化和本地化 242

16.1	什么是本地化	242
16.2	本地化字符串	243
16.3	对未本地化的字符串进行审查	244
16.4	格式化数字和日期	245
16.5	nib 文件和 Base Internationalization	248
16.6	本地化复杂字符串	248
16.7	小结	250
16.8	扩展阅读	251

第 17 章 调试 252

17.1	LLDB	252
17.2	使用 LLDB 进行调试	252
17.2.1	dSYM 文件	253
17.2.2	符号化	254
17.3	断点	255
17.4	观察点	258
17.5	LLDB 控制台	259
17.6	NSZombieEnabled 标志	262
17.7	不同的崩溃类型	263
17.7.1	EXC_BAD_ACCESS	263
17.7.2	SIGSEGV	263
17.7.3	SIGBUS	264
17.7.4	SIGTRAP	264
17.7.5	EXC_ARITHMETIC	264
17.7.6	SIGILL	264
17.7.7	SIGABRT	264
17.7.8	看门狗超时	265
17.7.9	自定义错误信号处理程序	265
17.8	断言	265
17.9	异常	267
17.10	收集崩溃报告	268
17.11	第三方崩溃报告服务	269
17.12	小结	270
17.13	扩展阅读	270

第 18 章 性能调优 272

18.1	性能思维模式	272
18.1.1	指导方针一: 产品是为了取悦用户才存在的	272
18.1.2	指导方针二: 设备是为了方便用户而存在的	272
18.1.3	指导方针三: 做到极致	272
18.1.4	指导方针四: 用户的感知才是实际的	273
18.1.5	指导方针五: 关注能带来大收益的方面	273
18.2	欢迎走入 Instruments 的世界	273
18.3	查找内存问题	275

18.4	查找 CPU 问题	279
18.4.1	Accelerate 框架	282
18.4.2	GLKit	283
18.4.3	编译器优化	283
18.4.4	链接器优化	284
18.5	绘图性能	284
18.6	优化磁盘访问和网络访问	286
18.7	小结	286
18.8	扩展阅读	286

第四部分 超越极限

第 19 章 近乎物理效果：UIKit 动力学

19.1	动画类、行为和动力项	290
19.2	UIKit “物理”	291
19.3	内置行为	292
19.3.1	迅速移动	292
19.3.2	附着	292
19.3.3	推力	293
19.3.4	重力	294
19.3.5	碰撞	294
19.3.6	动力项	295
19.4	行为层次结构	295
19.5	自定义操作	295
19.6	实战：一个“撕开”视图	296
19.6.1	拖拽视图	296
19.6.2	撕开该视图	297
19.6.3	添加额外效果	300
19.7	多个动力学动画类	301
19.8	与 UICollectionView 交互	302
19.9	小结	305
19.10	扩展阅读	305

第 20 章 魔幻的自定义过渡

20.1	iOS 7 中的自定义过渡	306
20.2	过渡协调器	307
20.3	集合视图和布局过渡	308
20.4	使用故事板和自定义联线的自定义 视图控制器过渡	308
20.5	自定义视图控制器过渡：iOS 7 风格	308

20.6	使用 iOS 7 SDK 的交互式自定义 过渡	310
20.7	小结	312
20.8	扩展阅读	313

第 21 章 精妙的文本布局

21.1	理解富文本	314
21.1.1	字符与字形	314
21.1.2	理解字体	316
21.1.3	段落样式	316
21.2	属性化字符串	317
21.2.1	用字体描述符选择字体	318
21.2.2	设置段落样式	319
21.2.3	HTML	319
21.2.4	简化属性化字符串的使用	320
21.3	动态字体	321
21.4	Text Kit	322
21.4.1	Text Kit 的组件	323
21.4.2	多容器布局	324
21.4.3	排除路径	325
21.4.4	继承文本容器	326
21.4.5	继承文本存储	327
21.4.6	继承布局管理器	331
21.4.7	针对字形的布局	334
21.5	Core Text	337
21.5.1	用 CTFramesetter 进行 简单的布局	337
21.5.2	为非连续路径创建框架	338
21.5.3	排版器、文本行、连续文本 和字形	340
21.6	小结	340
21.7	扩展阅读	341

第 22 章 Cocoa 的大招：键值编码

	和观察	342
22.1	键值编码	342
22.1.1	用 KVC 赋值	344
22.1.2	用键路径遍历属性	345
22.1.3	KVC 和容器类	345
22.1.4	KVC 和字典	347
22.1.5	KVC 和非对象	347

22.1.6 用 KVC 实现高阶消息传递	347	23.8 扩展阅读	367
22.1.7 容器操作符	347	第 24 章 深度解析 Objective-C	368
22.2 键值观察	348	24.1 理解类和对象	368
22.2.1 KVO 和容器类	350	24.2 使用方法和属性	370
22.2.2 KVO 是如何实现的	351	24.3 使用方法签名和调用	372
22.3 KVO 的权衡	351	24.4 消息传递如何工作	378
22.4 小结	352	24.4.1 动态实现	378
22.5 扩展阅读	353	24.4.2 快速转发	380
第 23 章 超越队列：GCD 高级功能	354	24.4.3 普通转发	383
23.1 信号量	354	24.4.4 转发失败	384
23.2 分派源	356	24.4.5 各种版本的 objc_msgSend	384
23.3 定时器源	358	24.5 方法混写	385
23.4 单次分派	358	24.6 ISA 混写	387
23.5 队列关联数据	359	24.7 方法混写与 ISA 混写	388
23.6 分派数据和分派 I/O	362	24.8 小结	389
23.7 小结	367	24.9 扩展阅读	389

Part 1

第一部分

全新功能

本 部 分 内 容

- 第 1 章 全新的系统
- 第 2 章 世界是平的：新的 UI 范式