

iOS PROGRAMMING 4TH EDITION

iOS编程 (第4版)

THE BIG NERD RANCH GUIDE



荣获Jolt生产力大奖

[美] Christian Keur Aaron Hillegass Joe Conway 著
丁道骏 译 张召 吴春燕 审校



华中科技大学出版社

<http://www.hustp.com>

iOS 编程(第 4 版)

Christian Keur
[美] Aaron Hillegass 著
Joe Conway

丁道骏 译
张召 吴春燕 审校

华中科技大学出版社
中国·武汉

内 容 简 介

《iOS 编程(第4版)》获21届Jolt生产力大奖。第4版更新了iOS 7和Xcode 5的内容,涵盖了开发iOS应用的方方面面。从Objective-C基础知识到新增加的语言特性;从AppKit库到常见的Cocoa设计模式;从Xcode技巧到Instruments等。如果读者刚接触iOS编程,阅读本书前两章可以快速入门。如果读者已经有iOS编程经验,阅读本书可以拾遗补缺。

Authorized translation from the English language edition, entitled IOS PROGRAMMING: THE BIG NERD RANCH GUIDE, 4th Edition, 9780321942050 by CONWAY, JOE; HILLEGASS, AARON, published by Pearson Education, Inc, publishing as Big Nerd Ranch Guides, Copyright ? 2014 Big Nerd Ranch, LLC.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS, Copyright ? 2015.

湖北省版权局著作权合同登记 图字:17-2014-383号

图书在版编目(CIP)数据

iOS 编程/(美)科尔(Keur,C.), (美)赫乐嘎斯(Hillegass,A.), (美)康威(Conway,J.) 著;丁道骏译. —4版. —武汉:华中科技大学出版社,2014.11

ISBN 978-7-5609-9790-2

I. ①i… II. ①科… ②赫… ③康… ④丁… 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269667号

iOS 编程(第4版) [美]Christian Keur, Aaron Hillegass, Joe Conway 著
丁道骏 译 张召 吴春燕 审校

策划编辑:徐定翔

责任校对:刘 竣

责任编辑:陈元玉

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321915

录 排:华中科技大学惠友文印中心

印 刷:湖北新华印务有限公司

开 本:787mm×960mm 1/16

印 张:35

字 数:940千字

版 次:2015年1月第4版第1次印刷

定 价:99.90元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前言

Introduction

要成为一名优秀的 iOS 程序员，需要攻克以下三道难题。

- 必须学会 Objective-C 语言。Objective-C 是 C 语言的扩展，小巧简单。读完本书的前 3 章，读者就能掌握 Objective-C 的基础知识。
- 必须掌握 Cocoa 的常用技术。其中包括内存管理、委托机制（delegation）、固化机制（archiving），以及如何正确使用视图控制器（view controller）。理解这些技术需要花些时间。本书前半部分会介绍这些内容。
- 必须掌握框架（framework）。读者的最终目标是了解 iOS 的所有框架，学会如何使用框架中的每一个类和方法。但这几乎是不可能完成的任务：iOS 有 3000 多个方法，200 多个类。而且随着 iOS 的版本升级，Apple 还会不断地加入新的类和新的方法。本书将会介绍 iOS SDK 中的各个组成部分，但是不会太过深入。作者的目标是带领读者入门，使读者能够自行阅读并理解 Apple 的参考文档。

Big Nerd Ranch 公司采用本书作为“iOS 新手培训课程”的教材。这些内容经过了长期的检验，并且帮助很多人成为 iOS 程序员。真心希望本书也能给你带来帮助。

本书适合哪些读者

本书假设读者已经跃跃欲试，准备开发 iOS 应用，所以不会花费笔墨去证明 iPhone、iPad 和 iPod touch 是很棒的产品。本书假设读者了解 C 语言并对“面向对象编程”有一定的了解。如果不是，建议读者先阅读 Big Nerd Ranch 出版的《Objective-C 编程》（华中科技大学出版社）。

第 4 版有哪些更新

第 4 版使用的是 Xcode 5，并且所有的应用都运行于安装有 iOS 7 的设备或模拟器。

作者在第 4 版采用了更加现代化的语法编写 Objective-C 代码，使用属性、点语法、实例变量自动合成和字面量，并更多地使用 Block 对象。

第 4 版还加入了众多新章节，分别介绍第 3 版以来 Apple 为 iOS 新加入的特性，包括自动布局、动态字体、状态恢复、Block 动画和 **NSURLSession** 等。

除了以上这些明显的变化，作者还根据读者和学生提出的问题对本书进行了大量修订。可

以说与前一版本相比，新版页页有改进。

教学理念

本书将向读者传授 iOS 编程的基本概念。在阅读的同时，读者还要输入大量的代码，并创建一组应用。这样，完成本书的学习后，读者得到的将不仅是知识，还有经验。相信你我多少都有过被灌输知识的痛苦经历，所以本书转而使用“边做边学”的教学方法——概念与代码并重。

多年的 iOS 编程教学工作让作者了解到：

- 某些概念是学习 iOS 编程必须知道的。本书会集中介绍这部分内容。
- 学习能立即派上用处的概念，效果最佳。
- 知识与经验并重时，学习效果最佳。
- 实际操作很重要。本书会要求读者先输入代码，再理解含义。读者可能会觉得这种不明就里的模仿意义不大，但是“找出错误并修正代码”是学习编程的好方法。这种最基础的调试过程不仅不是累赘，反而能帮助读者彻底理解代码。这也是作者鼓励读者自己输入代码的原因。虽然可以直接下载例子代码，但是拷贝粘贴不是编程。本书对读者及读者的编程技能有更高的期望。

这种模仿对读者意味着什么？意味着读者要信任本书作者，而且要有耐心。本书会尽可能地将问题讲透，但读者有时只能相信作者的意见（如果读者对此有异议，请往下看——作者列出了若干解决方案，也许能有帮助）。遇到暂时不能理解的概念时不要气馁，因为本书不会将涉及某个概念的所有知识一次全部介绍完，这是有意为之。如果某个概念没有解释清楚，那么很可能会在需要时再提供更详细的介绍。有些初看无法理解的概念，可能会在读者第一次（或第十二次）实际应用时突然变得清晰易懂。

每个人的学习方法不同。读者可能会喜欢本书这种“按需分步介绍概念”的方法，也可能不喜欢。如果是后者，这里提供若干建议。

- 先不要着急，等待在后续的章节中将问题讲透。
- 查阅索引，先阅读相关概念的详细介绍。
- 查阅 Apple 提供的在线文档。这些文档是非常重要的开发工具，需要多多练习使用。读者应该尽早地，也应尽可能多地使用在线文档。
- 如果在学习 Objective-C 或面向对象的编程概念时遇到困难（或是预感会有困难），那么建议读者先阅读 Big Nerd Ranch 的《Objective-C 编程》（华中科技大学出版社出版）。

如何使用本书

本书内容基于 Big Nerd Ranch 的培训课程，所以有其特定的阅读方法。

读者可以先设定一个合理的目标，例如“每天阅读一章”。然后在阅读时为自己找一个安静的场所，至少一个小时内不会被打断。关掉 Email、Twitter 客户端和聊天工具——读书无法多任务并行，必须集中精力。

在阅读的同时，读者还需要编写代码。根据读者的喜好，可以先通读整章。但是真正的学习从编写代码开始。要真正理解某个概念，需要编写程序并进行实际操作，尤其是调试 (debug) 程序的过程，会特别有帮助。

书中的部分项目需要使用额外的文件，例如第 1 章的 Quiz 应用需要一个图标文件。本书已经为读者准备好了这些文件，通过以下网址可以下载本书的项目代码和资源文件：
<https://github.com/dingdaojun/iOSProgramming4ed>。

学习分两类。学习历史时，要做的只是在已经理解的知识架构上添加更多细节。我们将这类学习称为简单学习 (easy learning)。学习历史的确需要花费很长的时间，但是难度不大。学习 iOS 编程则是困难学习 (hard learning)，会经常卡壳，尤其是在刚开始的时候。作者编写本书的目的是帮助读者越过难度陡增的学习曲线。下面提供两则建议，希望帮助读者轻松越过障碍。

- 找位懂 iOS 开发并且愿意回答读者提问的程序员。第一次尝试将应用安装至 iOS 设备时，如果缺少有经验的程序员的协助，就可能会遇到困难。
- 保证足够的睡眠。缺少睡眠将无法记住所学的知识。

本书是如何组织的

本书各章都会先介绍一个或多个 iOS 开发概念，然后给出具体的示例代码。每章末尾有练习，为读者提供编写代码的机会。建议读者至少完成部分的练习，以巩固所学，同时也提升自信。此外，多数章节最后会有一到两个“深入学习”部分，对之前介绍的内容做一些补充。

第 1 章介绍如何创建并安装一个很小的应用，以此带领读者入门 iOS 开发。读者将开始学习如何使用 Xcode 和 iOS 模拟器，以及创建项目和文件所需的全部步骤。第 1 章还介绍了 Model-View-Controller 及其在 iOS 开发中的作用。

第 2、3 章简单介绍 Objective-C 和内存管理。这两章没有创建 iOS 应用，而是构建并调试了一个名为 RandomItems 的工具，以帮助读者理解这些概念。

第 4、5 章将创建一个名为 Hypnosister 的应用，介绍视图和视图层次结构，并使用它们构建用户界面。

第 6、7 章通过 HypnoNerd 应用介绍如何使用视图控制器管理用户界面。读者将有机会练习如何使用视图和视图控制对象，并通过 UITabBar 对象切换屏幕。另外，读者还将获得大量与委托机制有关的经验（委托机制是一种重要的设计模式），并学习使用协议 (protocol)、调

试器和本地通知。

第 8 章介绍本书最庞大的应用 **Homepwner**（顺便提一句，**Homepwner** 并没有拼错。读者可以通过 www.urbandictionary.com 找到 **pwn** 的定义）。**Homepwner** 的功能是保留一份财产清单，以供发生灾难后核对。本书将花 14 章的篇幅来介绍 **Homepwner**。

第 8、9 和 19 章介绍如何使用并显示表格。读者将学到 **UITableView**、**UITableViewController** 和数据源机制；学会如何在表格视图中显示数据、如何让用户编辑表格及如何改善相关的界面。

第 10 章以第 6 章的导航机制为基础，介绍 **UINavigationController** 的用法，并为 **Homepwner** 增加一个垂直（drill-down）界面和一个导航条。

第 11 章介绍如何通过相机拍照，以及如何为 **Homepwner** 实现显示图片和保存图片的功能。读者将有机会使用 **NSDictionary** 和 **UIImagePickerController**。

第 12、13 章将暂时跳出 **Homepwner** 应用，创建一个名为 **TouchTracker** 的绘图应用，以介绍触摸事件。读者将学到如何为应用增加多点触摸功能，以及如何使用 **UIGestureRecognizer** 来响应特定的手势。此外，读者还将了解第一响应对象（first responder）和响应对象链（responder chain），并针对 **NSDictionary** 做更多练习。

第 14 章将向读者介绍如何使用调试仪表、Instruments 和静态分析器（static analyzer），并借此优化 **TouchTracker** 应用的性能。

第 15、16 章会将 **Homepwner** 改为通用应用（universal application），能在 iPhone 和 iPad 上全屏运行。同时，读者还会学习自动布局（auto layout），使 **Homepwner** 的界面可以适配不同大小的屏幕。

第 17 章介绍如何实现自动转屏（autorotation），以及如何使用模态视图控制器和 iPad 特有的 **UIPopoverController**。

第 18 章将介绍多种保存/读取数据的机制。值得注意的是，**Homepwner** 应用会通过 **NSCoding** 协议固化（archive）数据。本章还介绍了切换应用状态（例如激活状态、后台运行状态和悬停状态）的各个过程。

第 20 章将升级 **Homepwner** 应用，支持动态字体（dynamic type），根据用户的偏好显示不同大小的字体。

第 21 章会再次跳出 **Homepwner** 应用，通过创建一个名为 **Nerdfeed** 的应用，介绍如何实现 Web 服务通信。**Nerdfeed** 会通过 **NSURLSession** 和 **NSJSONSerialization** 从指定的服务器获取并解析 JSON 数据。JSON 是目前最流行的数据传输格式。此外，**Nerdfeed** 还会用 **UIWebView** 对象来显示相应的 Web 页面。

第 22 章将介绍 **UISplitViewController**，并为 **Nerdfeed** 添加 **UISplitViewController** 对象，以适应 iPad 更大的屏幕尺寸。

第 23 章介绍 Core Data。**Homepwner** 应用将改用 **NSManagedObjectContext** 对象来保存

/读取数据。

第 24 章将为 **Homepwner** 启用状态恢复，无论用户何时重新回到应用，都会看到自己最后一次看到的界面。

第 25 章将介绍与国际化、本地化有关的概念和技术。读者将学习如何使用 **NSLocale**、字符串对照表（strings table）和 **NSBundle**，完成 **Homepwner** 应用的本地化。

第 26 章将介绍如何通过 **NSUserDefaults** 来永久保存用户偏好设置（user preferences）。本章将彻底完成 **Homepwner** 应用的开发。

第 27 章将介绍 Block 动画，并为 **HypnoNerd** 应用添加动画效果。读者将学习几种简单又有趣的动画效果，包括 iOS 7 中新引入的弹簧动画。

第 28 章将介绍 storyboard，一项能够帮助读者构建应用的 iOS 新特性。读者将学习如何使用 storyboard 文件来构建应用，以及 storyboard 的优缺点。

代码风格

本书包含大量代码，作者希望这些代码及其背后的设计思路能具备参考价值。作者在编写这些代码时，已经尽可能地符合 Cocoa 编程习惯。尽管如此，其中的某些部分还是会和 Apple 的示例代码，或者其他书籍中的代码有一定的差别。虽然读者目前可能还无法理解这些差别的具体含义，但是在正文开始前，还是有必要将这些问题先列出来。

- 本书采用编写代码或 XIB 文件的方式来创建视图控制器。还有一种方式是 storyboard 文件，本书第 28 章会介绍 storyboard 文件及其用法。但是作者自己在 Big Nerd Ranch 开发 iOS 应用时，几乎没有用过 storyboard 文件。
- 本书的绝大部分项目都创建自最简单的 Xcode 模板：empty application（空应用）。这里没有采用其他模板，是因为这些模板会自动生成额外代码，不利于初学者理解应用的工作原理。

版式说明

为了方便读者阅读，本书会对某些特定的内容使用专门的字体。其中，类名、方法名和函数名会以粗体、等宽的字体显示。类名的首字母会用大写，方法名的首字母会用小写。正文中出现的类名、方法名和函数名将采用加粗的 Bitstream Vera Sar 字体表示，例如，“在 **BNRRexViewController** 类的 **loadView** 方法里，使用 **NSLog** 函数输出结果。”

正文中出现的变量、常量和类型将采用 Bitstream Vera Sar 字体表示，但是不加粗。例如，“将变量 **fido** 定义为 **float** 类型，并赋予初始值 **M_PI**。”

应用程序和系统菜单选项将采用 Helvetica 字体表示，例如，“打开 Xcode，从 File 菜单选择 New Project...”。

书中的所有示例代码都将采用 Bitstream Vera Sar 字体表示。需要读者输入的代码会加粗显示。例如下面这段代码，除了第一行和最后一行外（这两行是已经存在的代码，列出来是为了让读者知道应该在哪里加入新的代码），其余部分都需要读者自行输入。

```
@interface BNRQuizViewController ()  
  
@property (nonatomic, weak) IBOutlet UILabel *questionLabel;  
@property (nonatomic, weak) IBOutlet UILabel *answerLabel;  
  
@end
```

开发所需的硬件与软件

开发 iOS 应用要使用（也只能使用）基于 Intel 芯片的 Mac 计算机。读者需要下载由 Apple 提供的 iOS SDK，其中包括 Xcode（Apple 的集成开发环境）、iOS 模拟器，以及其他开发工具。

读者还需要注册加入 Apple 的 iOS 开发者计划，费用为每年 99 美元。原因有以下三个。

- 注册后可以免费下载最新的开发工具。
- 任何应用必需先经过数字签名，然后才能在设备上运行。因为只有注册会员才能签名应用，所以，如果读者需要在设备上测试应用，就需要注册。
- 只有会员才能将应用提交至 App Store。

如果读者打算花时间读完本书的全部内容，那么注册加入 iOS 开发者计划是值得的。注册网站：<http://developer.apple.com/programs/ios/>。

进行 iOS 开发需要准备哪些设备？本书前半部分所涉及的多数应用都是针对 iPhone 的，但是也能在 iPad 上运行。在 iPad 上运行 iPhone 应用时，系统只会以 iPhone 的屏幕尺寸显示。这样虽然无法充分利用 iPad 的大尺寸屏幕，但是在没有 iPhone 的情况下也是可以的。本书的前几章会集中介绍 iOS SDK 的基础部分，这些内容和设备无关。稍后的章节会介绍若干只针对 iPad 的内容，以及如何让应用能够在所有的 iOS 设备上全屏运行。

读者是否已经准备好了呢？下面开始正文。

目录

Table of Contents

前言.....	xiii
本书适合哪些读者	xiii
第 4 版有哪些更新	xiii
教学理念	xiv
如何使用本书	xv
本书是如何组织的	xv
代码风格	xvii
版式说明	xvii
开发所需的硬件与软件	xviii
第 1 章 第一个简单的 iOS 应用.....	1
1.1 创建 Xcode 项目	2
1.2 模型-视图-控制器	4
1.3 设计 Quiz	5
1.4 创建视图控制器	6
1.5 创建界面	8
1.6 创建关联	14
1.7 创建模型对象	18
1.8 大功告成	21
1.9 在模拟器上运行应用	22
1.10 安装应用	23
1.11 应用图标	25
1.12 启动图片	27
第 2 章 Objective-C	29
2.1 对象	29
2.2 使用对象	30
2.3 编写命令行工具 RandomItems	33
2.4 创建 Objective-C 类的子类	38
2.5 深入学习 NSArray 与 NSMutableArray	58
2.6 异常与未知选择器	60

2.7 练习	62
2.8 初级练习：查找问题	62
2.9 中级练习：另一个初始化方法	62
2.10 高级练习：另一个类	63
2.11 关于深入学习部分	63
2.12 深入学习：如何为类命名	63
2.13 深入学习：#import 和 @import	64
第3章 通过 ARC 管理内存	65
3.1 栈	65
3.2 堆	66
3.3 指针变量与对象所有权	66
3.4 强引用与弱引用	70
3.5 属性	74
3.6 深入学习：属性合成	81
3.7 深入学习：Autorelease 池与 ARC 历史	83
第4章 视图与视图层次结构	85
4.1 视图基础	86
4.2 视图层次结构	86
4.3 创建 UIView 子类	88
4.4 在 drawRect:方法中自定义绘图	94
4.5 关于开发者文档	105
4.6 初级练习：绘制图像	106
4.7 深入学习：Core Graphics	106
4.8 高级练习：阴影和渐变	108
第5章 视图：重绘与 UIScrollView	111
5.1 运行循环和重绘视图	112
5.2 类扩展	114
5.3 使用 UIScrollView	115
第6章 视图控制器	119
6.1 创建 UIViewController 子类	120
6.2 另一个视图控制器	123
6.3 UITabBarController	130
6.4 视图控制器的初始化方法	134
6.5 添加本地通知	135
6.6 加载和显示视图	136

6.7 与视图控制器及其视图进行交互	138
6.8 初级练习：增加一个标签项	139
6.9 中级练习：控制逻辑	139
6.10 深入学习：键值编码	139
6.11 深入学习：Retina 显示屏	140
第 7 章 委托与文本输入	143
7.1 文本框（UITextField）	143
7.2 委托	146
7.3 协议	148
7.4 向屏幕中添加 UILabel 对象	150
7.5 运动效果	152
7.6 使用调试器	153
7.7 深入学习：main()与 UIApplication	157
7.8 中级练习：捏合-缩放	157
第 8 章 UITableView 与 UITableViewController	159
8.1 编写 Homepwner 应用	159
8.2 UITableViewController	160
8.3 UITableView 数据源	164
8.4 UITableViewCell 对象	170
8.5 代码片段库	175
8.6 初级练习：表格段	178
8.7 中级练习：固定行	178
8.8 高级练习：修改 UITableView 对象的外观	178
第 9 章 编辑 UITableView	179
9.1 编辑模式	179
9.2 增加行	185
9.3 删除行	187
9.4 移动行	188
9.5 初级练习：更改“删除”按钮的标题	190
9.6 中级练习：禁止移动某个表格行	190
9.7 高级练习：彻底禁止移动某个表格行	190
第 10 章 UINavigationController	191
10.1 UINavigationController 对象	192
10.2 额外的视图控制器	196
10.3 UINavigationController 的导航功能	201

10.4	UINavigationController	205
10.5	初级练习：显示数字键盘	210
10.6	中级练习：关闭数字键盘	210
10.7	高级练习：压入更多视图控制器	210
第 11 章	相机	211
11.1	通过 UIImageView 对象显示照片	212
11.2	通过 UIImagePickerController 拍摄照片	216
11.3	创建 BNRImageStore	220
11.4	NSDictionary	222
11.5	创建并使用键	225
11.6	使用 BNRImageStore	227
11.7	关闭键盘	228
11.8	初级练习：编辑照片	230
11.9	中级练习：删除照片	230
11.10	高级练习：Camera Overlay	230
11.11	深入学习：导航实现文件	230
11.12	深入学习：摄像	233
第 12 章	触摸事件与 UIResponder	235
12.1	触摸事件	235
12.2	创建 TouchTracker 应用	237
12.3	实现 BNRDrawView，完成绘图功能	239
12.4	处理触摸事件并创建线条对象	240
12.5	初级练习：保存与读取	245
12.6	中级练习：颜色	245
12.7	高级练习：圆圈	246
12.8	深入学习：响应对象链	246
12.9	深入学习：UIControl	247
第 13 章	UIGestureRecognizer 与 UIMenuController	249
13.1	UIGestureRecognizer 子类	250
13.2	用 UITapGestureRecognizer 对象识别“按下”手势	250
13.3	同时添加多种触摸手势	252
13.4	UIMenuController	254
13.5	UILongPressGestureRecognizer	256
13.6	UIPanGestureRecognizer 以及同时识别多个手势	257
13.7	深入学习：UIMenuController 与 UIResponderStandardEditActions	260

13.8	深入学习：再谈 UITapGestureRecognizer	261
13.9	中级练习：修正错误	262
13.10	高级练习：速度与宽度	262
13.11	高级练习：颜色	262
第 14 章	调试工具	263
14.1	仪表	263
14.2	Instruments	265
14.3	静态分析器	275
14.4	项目、目标和构建设置	277
第 15 章	自动布局入门	283
15.1	通用化 Homepwner	283
15.2	自动布局系统	285
15.3	在 Interface Builder 中添加约束	289
15.4	调试约束问题	298
15.5	初级练习：打造完美界面	306
15.6	中级练习：通用化 Quiz	307
15.7	深入学习：使用 _autolayoutTrace 方法调试约束问题	307
15.8	深入学习：使用多个 XIB 文件	308
第 16 章	在代码中使用自动布局	309
16.1	视觉化格式语言	310
16.2	创建约束	311
16.3	添加约束	312
16.4	固有内容大小	315
16.5	另一种添加方式	316
16.6	深入学习：NSAutosizingMaskLayoutConstraint	318
第 17 章	自动转屏，UIPopoverController 与模态视图控制器	321
17.1	自动转屏	321
17.2	自动转屏通告机制	324
17.3	UIPopoverController	326
17.4	更多的模态视图控制器	329
17.5	线程安全的单例	337
17.6	初级练习：为另一个类添加线程安全的单例	339
17.7	高级练习：UIPopoverController 对象的外观	339
17.8	深入学习：位掩码	339
17.9	深入学习：视图控制器之间的关系	340

第 18 章 保存、读取与应用状态	345
18.1 固化	345
18.2 应用沙盒	348
18.3 NSKeyedArchiver 与 NSKeyedUnarchiver	350
18.4 应用状态与状态切换	353
18.5 通过 NSData 将数据写入文件	356
18.6 NotificationCenter 和内存过低警告	358
18.7 模型-视图-控制器-存储设计模式	361
18.8 初级练习: PNG	361
18.9 深入学习: 应用的状态切换	362
18.10 深入学习: 文件系统的读取和写入	363
18.11 深入学习: 应用程序包	366
第 19 章 创建 UITableViewCell 子类	369
19.1 创建 BNRItemCell	369
19.2 处理图片	377
19.3 由 UITableViewCell 对象转发动作消息	380
19.4 捕获变量	385
19.5 初级练习: 设置颜色	386
19.6 高级练习: 缩放	387
19.7 深入练习: UICollectionView	387
第 20 章 动态字体	389
20.1 使用用户首选字体	390
20.2 响应用户首选字体的改变	392
20.3 修改自动布局约束	393
20.4 确定用户首选字体大小	395
20.5 修改 BNRItemCell	397
第 21 章 Web 服务与 UIWebView	403
21.1 Web 服务	404
21.2 UIWebView	414
21.3 认证信息	416
25.4 中级练习: 加强 UIWebView	418
21.5 高级练习: 课程预告	418
21.6 深入学习: HTTP 请求主体	419
第 23 章 Core Data	431
23.1 对象-关系映射	431

23.2	用 Core Data 重写 BNRItemStore 的数据保存功能	432
23.3	再谈 SQL	450
23.4	Faults	451
23.5	各种存取机制的优缺点	453
23.6	初级练习: Asset 的 iPad 界面	454
23.7	中级练习: 增加 BNRAssetType 对象	454
23.8	高级练习: 显示某种类型的 BNRItem 对象	454
第 24 章	状态恢复	455
24.1	状态恢复的工作原理	455
24.2	启用状态恢复	456
24.3	恢复标识和恢复类	457
24.4	状态恢复与应用生命周期	459
24.5	恢复视图控制器	461
24.6	编码状态数据	464
24.7	保存视图状态	465
24.8	中级练习: 为另一个应用启用状态恢复	467
24.9	深入学习: 设置快照	467
第 25 章	本地化	469
25.1	通过 NSNumberFormatter 实施国际化	470
25.2	资源的本地化	473
25.3	NSLocalizedString()与字符串对照表	477
25.4	初级练习: 再添加一套本地化资源	480
25.5	深入学习: NSBundle 在国际化过程中的作用	480
25.6	深入学习: 不通过基础国际化对 XIB 文件实施本地化	481
第 26 章	NSUserDefaults	483
26.1	NSUserDefaults	483
26.2	设置束	487
第 27 章	控制动画	491
27.1	基础动画	491
27.2	关键帧动画	494
27.3	在动画完成后执行特定的代码	496
27.4	弹簧动画	497
27.5	中级练习: 提升 Quiz 的用户体验	498
第 28 章	UIStoryboard	499
28.1	创建 Storyboard 文件	499

28.2	Storyboard 文件中的 UINavigationController.....	503
28.3	Segue	506
28.4	改变颜色	512
28.5	传递数据	513
28.6	Storyboards 的优缺点	520
28.7	深入学习：状态恢复	521
第 29 章	后记	523
29.1	接下来做什么	523
29.2	结束语	524
第 22 章	UISplitViewController	421
22.1	在 Nerdfeed 中使用 UISplitViewController.....	422
22.2	在竖排模式下显示主视图控制器	425
22.3	将 Nerdfeed 改为通用应用	428
索引		525