

iOS PROGRAMMING 2ND EDITION

iOS编程 (第2版)

THE BIG NERD RANCH GUIDE

[美] Joe Conway & Aaron Hillegass 著

夏伟频 译 李骏 审校



华中科技大学出版社

<http://www.hustp.com>

iOS 编程（第 2 版）

[美] Joe Conway 著
Aaron Hillegass

夏伟频 译

李 骏 审校

华中科技大学出版社

中国·武汉

内 容 简 介

本书涵盖了开发 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。本书的另一个特色是教读者以正确的方法解决问题。Objective-C 的习惯约定有哪些(例如命名约定、内存管理约定),创建子类时如何处理初始化方法,Cocoa 的常见设计模式有哪些,如何选择数据保存方法等。这些问题,如果读者自己摸索,难免费时费力。本书已经将这些知识整理、归纳并清楚地呈现在读者面前。

Original English language edition copyright © 2011 by Joe Conway and Aaron Hillegass The Chinese Translation edition Copyright © 2012 by HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS in arrangement with BIG NERD RANCH, INC.

湖北省版权局著作权合同登记 图字:17-2011-180 号

图书在版编目(CIP)数据

iOS 编程(第 2 版)/(美)Joe Conway,(美)Aaron Hillegass 著;夏伟频 译;李骏 审校. — 武汉:华中科技大学出版社,2012.4

ISBN 978-7-5609-7697-6

I. i… II. ①J… ②A… ③夏… ④李… 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(2012)第 011204 号

iOS 编程(第 2 版)

[美]Joe Conway Aaron Hillegass 著
夏伟频 译 李 骏 审校

策划编辑:徐定翔

责任校对:张 琳

责任编辑:陈元玉 林 航

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)87557437

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:34

字 数:886 千字

版 次:2012 年 4 月第 1 版第 1 次印刷

定 价:88.00 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

翻译说明

本书的术语、词汇翻译优先使用 Apple 公司的正式译法，我们参考了 Apple 公司中国官方网站、iOS 设备中文版使用手册、中文开发文档、中文软件系统界面的译法。有些翻译无固定译法可循，特作如下说明。

- 在 Objective-C 中，“发送消息”和“调用方法”的概念虽略有差别，但是可以认为实际效果相同。例如，“向某个对象发送 release 消息”和“调用某个对象的 release 方法”意思相同。
- 原书不区分“实例”（instance）和“对象”（object），中文版根据原文翻译。两者意思相同。
- 本书将 iOS 4.0 加入的语言特性 Blocks 翻译为“Block 对象”，因为 Blocks 是简称，其正式的名称为 Block Object。
- Category 翻译为“范畴”。“范畴”是 Apple 公司的官方开发文档选用的译法。
- Override 有多种译法，本书译为“覆盖”。

我们列出了详细的中英文术语对照表，并尽可能提供译法的出处。欢迎访问：
<http://www.iosprogrammingbook.com>。

Apple 公司不断推出新版本的 iOS SDK，加入新的特新，淘汰陈旧的功能。为了方便读者学习书中的示例，译者在此增加两点针对新版 SDK 的补充说明：如何添加 Window-Based Application 模板，以及解决 Xcode 调试器（debugger）无法定位异常抛出点的问题。限于篇幅，更多补充信息会发布在本书网站上，欢迎访问：
<http://www.iosprogrammingbook.com>。

添加 Window-Based Application 模板

本书的绝大部分例子都创建自 Window-Based Application 模板。但是在最新的 iOS SDK 中，Apple 公司去掉了该模板，并且会默认选择 Storyboard 和 Automatic Reference Counting（iOS 5 新加入的特性）。如果读者需要参照本书中的例子，从头开始创建项目，那么请参照以下步骤，加入本书原作者自己制作的 Window-Based Application 模板（该模板仅供学习使用）。此外，还要取消使用 Storyboard 和 Automatic Reference Counting。

步骤如下。

- 下载模板：<http://www.bignerdranch.com/solutions/Templates.zip>。

- 在 Finder 中，打开目录~/Library/Developer/Xcode/。
在 Mac OS X 10.7 中，~/Library/是隐藏目录。可用以下方法进入：在 Finder 中，选择“前往”菜单中的“前往文件夹...”（也可用快捷键 Command-Shift-G），然后输入上述的文件夹路径，并打开。
- 解压 Templates.zip，将解开的 Templates 目录放入~/Library/Developer/Xcode/。
- 重新运行 Xcode。

Xcode 调试器（debugger）无法定位异常抛出点

本书有多处提到：“当应用抛出异常时，调试器（debugger）会将执行代码停在异常的抛出点”。在当前版本的 iOS SDK 中，调试应用时，Xcode 的 debugger 可能不会将执行代码停在异常的抛出点。如果读者遇到了此类问题，可参考以下解决方法。

- 选中断点导航面板（Command+6）。
- 单击面板左下角的+按钮。
- 在新出现的弹出式菜单中，选择第一项“Add Exception Breakpoint...”。
- 在新出现的窗口中，不做任何改动，单击 Done 按钮。

致谢

Acknowledgments

虽然封面只列出了作者的名字，但是本书的面世离不开幕后很多人的帮助。作者借此表达真诚的谢意。

- iOS 新手培训课程的其他教员不断地提供建议和修改意见。他们是 Scott Ritchie、Brian Hardy、Mikey Ward、Christian Keur、Alex Silverman 和 Alex von Below。
 - 本书编辑 Susan Loper 将原本晦涩难懂的文字润色为通顺流畅的好文。
 - 几位技术审校者帮助我们找到并修正了多处瑕疵。他们是 Bill Monk、Mark Miller 和 Jonathan Saggau。
 - Ellie Volckhausen 设计了本书封面（照片拍摄的是自行车架的一个底部支撑脚）。
 - 来自 IntelligentEnglish.com 的 Chris Loper 设计并制作了本书的 EPUB 版本和 Kindle 版本。
 - 来自 Pearson Technology Group 的团队耐心地指导我们走完整个出版流程。
- 最后还要感谢我们的学生，他们激发我们编写了本书。

前言

Introduction

要成为一名优秀的 iOS 程序员，需要攻克以下三项难题。

- 必须学会 Objective-C 语言。Objective-C 是 C 语言的小巧且简单的扩展。读完本书的前 4 章，读者就能基本掌握 Objective-C 的相关知识。
- 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 **UIViewController** 的正确使用。理解这些技术需要花些时间。本书前半部分会覆盖这些内容。
- 必须掌握框架（framework）。我们的终极目标是学会 iOS 的所有框架，以及全部的类和方法。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 iOS 升级的时候会加入新的类和方法。本书会全面介绍 iOS SDK，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 采用本书作为“iOS 新手培训课程”的教材。这些课程经过了长期检验，帮助很多人成为了 iOS 程序员。真心希望这本书也能给你带来帮助。

本书适合哪些读者

Prerequisites

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。

此外，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 C 和 Objective-C 的入门书籍，例如《Objective-C 编程》。

第 2 版有哪些更新

What's Changed in the Second Edition?

首先，书名从《iPhone 编程》改为了《iOS 编程》。这是因为第 2 版加入了更多针对 iPad 的内容，例如新增加的关于 **UIPopoverController** 和 **UISplitViewController** 的章节，这些内容只对 iPad 有效。

Core Data 技术已经相当成熟，所以第 2 版删除了 SQLite 章节，不再介绍如何直接使用 SQLite。Core Data 这章的位置也提前了，以示其重要性。

第 2 版新增加了两章，分别介绍推送通知（push notification）和 Block 对象（blocks）。此外，还增加了一章专门介绍静态分析器（static analyzer）和性能分析工具 Instruments。

第 2 版假设读者使用的是 Xcode 4。获取 Xcode 4 的详细方法请见前言的最后一节。

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，对本书做了大量修订。可以说和前一版本相比，新版页页有改进。

教学理念

Our Teaching Philosophy

本书将向读者传授 iOS 编程的基本概念，同时会列出大量代码并创建一组应用。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。
- 对读者有用的概念，读者掌握得最牢。
- 知识与经验并重时效果最佳。
- 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，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，因为本书不会一次介绍完所有知识。如果某个概念没有解释清楚，那么很可能在下次出现时再提供更详细的介绍。有些初看无法理解的概念，可能会在你第一次（或第二十次）应用时突然变得清晰易懂。

每个人的学习方法不同。也许读者会喜欢本书介绍概念的方法，也可能不会喜欢。如果是后者，这里提供若干选项。

- 深呼吸，克服心理障碍。作者能做到，读者也可以。
- 查阅索引。先阅读相关内容的详细介绍。
- 查阅 Apple 的在线文档。这些文档是非常重要的“工具”，需要多多练习使用。读者应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》。

如何使用本书

How To Use This Book

本书内容乃是基于作者在 Big Nerd Ranch 教授的课程，故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。

然后要编写代码。如果读者愿意，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，需要编写程序并实际操作，特别是对程序的调试过程。

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：<http://www.bignerdranch.com/solutions/iOSProgramming.zip>。

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陡增的学习曲线。下面提供两则建议，以期帮助读者越过障碍。

- 找个懂 iOS 开发，并且愿意回答读者提问的程序员。第一次将应用安装至 iOS 设备时，缺少有经验的程序员的协助可能会特别困难。
- 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

How This Book Is Organized

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。每章末尾会有额外的练习，为读者提供更多编写代码的机会。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。另外，多数章节最后有一两个“扩展阅读”部分，对之前介绍的内容做一些扩展。

第 1 章向读者展示了创建并安装一个小应用的全过程，带领读者入门 iOS 开发。读者将学习 Xcode 和 iOS 模拟器，以及创建项目和文件所需的全部步骤。本章还介绍了 Model-View-Controller 及其在 iOS 开发中的作用。

第 2、3 章对 Objective-C 和内存管理做了简单的介绍。这两章没有创建带界面的 iOS 应用，而是创建并调试了一个名为 RandomPossessions 的工具，帮助读者理解这些概念（第 10 章介绍的 Homepwner 应用也会用到该工具以及相应的类文件）。

第 4、5 章介绍 Core Location 框架和 MapKit 框架并创建一个名为 Whereami 的地图应用。读者将获得大量和委托机制有关的经验（委托机制是一种重要的设计模式），并且有

机会使用协议（protocol）、框架（framework）、对象图和 Apple 开发文档。

第 6、7 章通过 Hypnosister 应用和 HypnoTime 应用，集中介绍 iOS 用户界面。读者将有大量的机会练习如何使用视图和视图控制对象，实现滚动、缩放并通过 **UITabBar** 切换屏幕。

第 8 章介绍方向感应器（accelerometer）。读者将学到如何从方向感应器获取数据，并通过过滤和使用这些数据来处理运动事件（包括摇动）。此外，还会用方向感应器数据为 HypnoTime 应用增加一项新功能。

第 9 章介绍如何创建一个规模稍小的应用 HeavyRotation。同时还会介绍通告机制（notifications）以及如何实现自动转屏（autorotation）。此外，还会介绍如何通过自动缩放（autoresizing）让 HeavyRotation 适应 iPad 的屏幕尺寸。

第 10 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 9 章的篇幅来完成 Homepwner。

第 10、11 章和第 16 章介绍如何以单列表格的形式显示数据。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以及如何改善界面。

第 12 章以第 7 章介绍的导航机制为基础，向读者介绍如何使用 **UINavigationController**，并为 Homepwner 增加一个垂直（drill-down）界面和一个导航条。

第 13 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。读者有机会在本章使用 **NSDictionary** 和 **UIImagePickerController**。

第 14 章介绍模态视图控制对象以及 iPad 特有的 **UIPopoverController**。此外，还会将 Homepwner 改为通用应用——一种既支持 iPhone 又支持 iPad 的原生应用。

第 15 章会介绍多种保存/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。本章还展示了如何使用多任务机制，以及切换应用状态（例如激活状态、后台运行状态和悬停状态）的各个过程。

第 17 章介绍 Core Data。Homepwner 应用将改用 **NSManagedObjectContext** 对象来保存/读取数据。

第 18 章介绍与国际化、本地化有关的概念和技术。通过 Homepwner 的本地化过程，读者将学会如何使用 **NSLocale**、字符串对照表（strings table）和 **NSBundle**。本章将完成 Homepwner 应用。

第 19 章介绍如何通过 **NSUserDefaults** 永久保存用户预置（user preferences）。

第 20 章会创建一个名为 TouchTracker 的绘图应用。读者将学到如何为应用增加多点触摸功能，并了解更多和触摸事件有关的内容。此外，还会介绍第一响应对象（first responder）和响应对象链（responder chain）概念，并针对 **NSDictionary** 做更多练习。

第 21 章通过调试 TouchTracker 的性能和内存问题，向读者介绍如何使用 Instruments。此外，还会介绍 Xcode 的方案（scheme）和静态分析器（static analyzer）。

第 22、23 章介绍层（layer）和 Core Animation 框架，并为 HypnoTime 应用实现动画效果。读者将学到隐式动画和动画对象，例如 **CABasicAnimation** 和 **CAKeyframeAnimation**。

第 24 章介绍 Objective-C 的两个重要特性：Block 对象（blocks）与范畴（Category）。

第 25 章通过创建 Nerdfeed 应用，介绍如何实现 Web 服务通信。Nerdfeed 通过 `NSURLConnection` 和 `NSXMLParser`，从指定的服务器获取并解析 RSS 文件。Nerdfeed 还会使用 `UIWebView` 对象显示相应的 Web 页面。

第 26 章介绍 `UISplitViewController`，并为 Nerdfeed 增加新的 `UISplitViewController` 对象，以适应 iPad 更大的屏幕尺寸。

第 27 章通过创建名为 `MediaPlayer` 的应用，介绍如何播放音频和视频。读者将学到如何播放音频和视频，如何保存这些资源，流媒体的限制以及底层音频 API。此外，读者还会学到允许 `MediaPlayer` 在后台运行状态下继续播放音乐的方法，使用后台执行机制需要遵守的规则，后台执行机制的其他用途。

第 28 章会创建一个 Mac 桌面应用。读者需要编写另一个 iOS 应用并通过 Bonjour 搜寻同一网段中的桌面应用，找到桌面应用后，你将通过 HTTP 协议与其进行通信。

第 29 章会扩充前一章的桌面应用，使之能够向 iOS 设备发送推送通知。

代码风格

Style Choices

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。这些代码会尽可能地符合 Cocoa 编程习惯，但是有些地方可能会和 Apple 的示例代码或是其他书籍中的代码有差别。读者可能会对此表示不解，所以这里先行列出。

- 存取方法（accessor method）有一种被称为 dot-notation 的调用语法。本书会解释 dot-notation，但不会使用。对初学者而言，dot-notation 容易产生混淆。
- 创建 `UIViewController` 子类时，本书会将指定初始化方法（designated initializer）统一改为 `init`。这样创建实例时无需知道 `UIViewController` 子类是否使用 XIB 文件，更不用知道文件名。
- 本书采用编写代码的方式创建视图控制对象。虽然可以通过 Xcode 直接在 XIB 文件中创建 `UIViewController` 实例，但这样做会导致项目难以理解和调试。
- 本书的绝大部分项目都创建自最简单的 Xcode 模板：window-based application（基于窗口的应用）。其他模板生成的代码不符合本书的代码风格，所以没有采用，以保证代码的一致性。

作者相信，以上规则能让本书的代码更易读，也更容易维护。当读者（在以上规则的限制下）完成本书的学习后，可以尝试打破这些规则，看看情况是会变好还是变糟。

版式说明

Typographical Conventions

为了方便读者阅读，本书使用了特定的字体和书写规范，其中包括：类的首字母大

写，方法首字母小写。应用程序和系统菜单选项采用 Helvetica 字体表示，例如，“打开 Xcode，选择 New Project...”。

正文中出现的代码关键字（如变量、常量、类型等）采用 Bitstream Vera Sar 字体表示，例如，“将变量 fido 定义为 float 类型，并赋予初始值 M_PI。”

正文中出现的类、方法、函数采用加粗的 Bitstream Vera Sar 字体表示，例如，“在 **RexViewController** 类的 **loadView** 方法里，使用 **NSLog** 函数输出结果。”

书中示例代码同样使用 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，除了第一行和最后两行外，其余部分加粗表示需要读者自行输入。

```
@interface QuizAppDelegate : NSObject <UIApplicationDelegate> {
    int currentQuestionIndex;

    // model 对象
    NSMutableArray *questions;
    NSMutableArray *answers;

    // view 对象
    IBOutlet UILabel *questionField;
    IBOutlet UILabel *answerField;
    UIWindow *window;
}
```

开发所需的硬件与软件

Necessary Hardware and Software

开发 iOS 应用要使用（也只能使用）基于 Intel 芯片的 Mac 电脑。读者要下载 Apple 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。原因有三。

- 注册后可以免费下载最新的开发工具。
- 只有签名后的应用才能在设备上运行，并且只有注册会员才能签名应用。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/>。

需要哪些 iOS 设备？本书前半部分涉及的多数应用都是针对 iPhone 的，但也能在 iPad 上运行，只是显示大小仍然是 iPhone 的屏幕尺寸。虽然浪费了 iPad 的屏幕，但仍可用。本书前几章会集中介绍 iOS SDK 的基础部分，而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，以及如何开发通用应用（Universal Application）——既能在 iPhone 上运行，也能在 iPad 上运行的原生应用。

目录

Table of Contents

前言	xiii
本书适合哪些读者	xiii
第 2 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 创建界面	5
1.3 模型-视图-控制器	10
1.4 声明	13
声明实例变量	13
声明方法	14
1.5 创建关联	15
设置指针	15
设置目标 (target) 和动作 (action)	17
关联小结	19
1.6 实现方法	20
1.7 用模拟器构建并运行应用	21
1.8 安装应用	23
1.9 应用图标	24
1.10 启动图片	27
第 2 章 Objective-C	31
2.1 对象	31
2.2 使用实例	32
创建对象	32
发送消息	33
释放对象	34
2.3 编写命令行工具 RandomPossessions	35
NSString	38
NSArray 与 NSMutableArray	39

2.4	子类化 Objective-C 类.....	40
	实例变量	44
	存取方法	45
	实例方法	48
	初始化方法	49
	其他初始化方法与初始化方法链	52
	类方法	53
	测试子类	55
2.5	异常与控制台窗口	56
2.6	快速枚举	57
2.7	练习	58
第 3 章	内存管理	59
3.1	内存管理基础	59
	C 语言的内存管理	59
	通过对象管理内存	61
3.2	引用计数	61
	使用 retain 计数	61
	使用 autorelease	64
	存取方法和内存管理	67
	实现 dealloc	69
	通过属性机制简化存取方法	70
	copy 和 mutableCopy	72
	Retain 计数规则	73
3.3	深入学习：再谈内存管理	74
第 4 章	委托机制与 Core Location	77
4.1	项目、目标和框架	77
4.2	Core Location	79
	通过 CLLocationManager 获取更新数据	81
4.3	委托机制	82
	协议	83
	委托机制、控制器对象和内存管理	85
4.4	使用调试器	86
4.5	练习：方位	91
4.6	深入学习：构建阶段、编译器错误和连接器错误	91
	预处理（Preprocessing）	92
	编译	93
	连接	94
第 5 章	MapKit 与文本输入	97
5.1	对象图	97
5.2	MapKit 框架	99
5.3	界面属性	99

5.4	设置 MapView 对象的委托对象	102
	使用文档	103
	创建 MKAnnotation 子类	108
	标记位置	112
	完成 Whereami	113
5.5	练习：扩充标注	114
5.6	练习：根据地理位置获取城市信息	114
第 6 章	创建 UIView 子类	115
6.1	创建自定义视图	116
	drawRect: 方法	116
	创建 UIView 实例	118
6.2	绘制文字和阴影	120
6.3	使用 UIScrollView	121
6.4	实现缩放功能	122
6.5	隐藏状态条	123
6.6	练习：颜色	125
6.7	深入学习：Retain 循环	125
6.8	深入学习：重画视图	125
第 7 章	视图控制对象	127
7.1	视图控制对象与 XIB 文件	127
7.2	使用视图控制对象	129
	创建 UITabBarController 对象	130
	创建视图控制对象和标签项	131
	为视图控制对象创建视图	135
7.3	视图的显示和消失	142
7.4	视图控制对象的生命周期与内存过低警告	142
7.5	UIViewController 子类与模板	146
7.6	练习：地图标签	147
第 8 章	方向感应器	149
8.1	设置方向感应器	149
8.2	获取方向感应器数据	151
8.3	加速方向与加速数值	151
8.4	使用加速感应器数据	151
8.5	使加速感应器数据更“平滑”	152
8.6	侦测摇动	153
8.7	练习：改变颜色	155
8.8	深入学习：滤波与频率	155
8.9	深入学习：Retina 显示屏	156
第 9 章	通告机制与设备转动	159
9.1	通告中心	159
9.2	UIDevice 通告	160

9.3	自动转屏	161
	用代码设置自动缩放掩码与按位运算符	167
9.4	强制使用横排模式	169
9.5	练习：设备近身通告	170
9.6	深入学习：覆盖自动转屏	170
第 10 章	UITableView 与 UITableViewController	171
10.1	编写 Homepwner 应用	171
10.2	UITableViewController	172
	创建 UITableViewController 子类	173
10.3	UITableView 数据源	176
	PossessionStore：单实例对象	177
	实现数据源方法	180
10.4	UITableViewCell 对象	182
	创建并获取 UITableViewCell 对象	183
	重用 UITableViewCell 对象	185
10.5	代码片段库	186
10.6	练习：表格段	189
第 11 章	编辑 UITableView	191
11.1	编辑模式	191
11.2	增加行	197
11.3	删除行	198
11.4	移动行	199
第 12 章	UINavigationController	203
12.1	UINavigationController	204
	UINavigationController	207
12.2	加入额外的 UIViewController	211
12.3	UINavigationController 的导航功能	217
	将视图控制对象压入栈	217
	视图控制对象之间的数据传递	219
	视图的显示和消失	221
12.4	练习：数字键盘（Number Pad）	221
第 13 章	相机	223
13.1	通过 UIImageView 显示图片	223
	通过 UIImagePickerController 拍摄照片	226
	ImageStore	231
	创建并使用键	234
	隐藏键盘	237
13.2	练习：删除图片	238
13.3	深入学习：摄像	238
第 14 章	UIPopoverController 与模态视图控制对象	241
14.1	将 Homepwner 改写为通用应用	242

14.2	UIPopoverController	244
14.3	模态视图控制对象	246
	关闭模态视图控制对象	249
	模态视图控制对象的样式	250
	编写视图控制对象的委托协议	252
	模态视图控制对象的过渡动画	254
第 15 章	保存、读取与多任务处理	257
15.1	应用沙盒	257
	获取文件路径	258
15.2	固化	260
	对象的固化	260
	对象的解固	262
15.3	应用状态、状态切换与多任务处理	265
15.4	通过 NSData 保存数据	269
15.5	再谈内存过低警告	271
15.6	模型-视图-控制器-存储设计模式	272
15.7	练习：为 Whereami 增加固化功能	272
15.8	深入学习：应用状态的切换	272
15.9	深入学习：文件系统的读取和写入	273
15.10	深入学习：程序包	276
第 16 章	创建 UITableViewCell 子类	279
16.1	创建 HomeownerItemCell	280
	创建子视图	281
	子视图的布局	282
	使用 UITableViewCell 子类对象	283
16.2	处理图片	284
16.3	练习：辅助指示视图	289
16.4	练习：缩小原图	289
第 17 章	Core Data	291
17.1	对象-关系映射	291
17.2	用 Core Data 重写 PossessionStore 的数据保存功能	291
	模型文件	292
	NSManagedObject 与 NSManagedObject 子类	297
	更新 PossessionStore	301
	为 Homeowner 增加 AssetType 功能	307
17.3	再谈 SQL	311
17.4	各种保存机制的优缺点	314
17.5	练习：增加 AssetType 对象	315
17.6	练习：AssetType 对象的 iPad 界面	315
第 18 章	本地化	317
18.1	通过 NSLocale 实现国际化	318