

需要整本电子书，联系我QQ: [2667271557](#);
此处是样章，取的完整版的前面几页，和最后
面几页；完整版是带书签的，样章没带书签；
另外需要其他书，也可以找我。

O'REILLY®



涵盖Xcode 6



Swift 编程实战

iOS应用开发实例及完整解决方案

iOS 8 Swift Programming Cookbook

机械工业出版社
China Machine Press

Vandad Nahavandipoor 著
梁士兴 陈作君 赵萌 等译

Swift编程实战： iOS应用开发实例及完整解决方案

Vandad Nahavandipoor 著
梁士兴 陈作君 赵萌 等译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权机械工业出版社出版

机械工业出版社

(Vandad Nahavandipoor.) 著; 宋上六等译. —北京: 机械工业出版社, 2016.5

(O'Reilly精品图书系列)

书名原文: iOS 8 Swift Programming Cookbook: Solutions & Examples for iOS Apps
ISBN 978-7-111-53927-8

I. S… II. ①万… ②梁… III. ①程序语言—程序设计 ②移动终端—应用程序—
程序设计 IV. ①TP312 ②TN929.53

中国版本图书馆CIP数据核字 (2016) 第112573号

北京市版权局著作权合同登记

图字: 01-2015-0261号

©2015 Vandad Nahavandipoor. All rights reserved.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and China Machine Press, 2016. Authorized translation of the English edition, 2014 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由O'Reilly Media, Inc. 出版2014。

简体中文版由机械工业出版社出版 2016。英文原版的翻译得到O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

封底无防伪标均为盗版

本书法律顾问

北京大成律师事务所 韩光/邹晓东

书 名/ Swift编程实战: iOS应用开发实例及完整解决方案

书 号/ ISBN 978-7-111-53927-8

责任编辑/ 陈佳媛

封面设计/ Karen Montgomery, 张健

出版发行/ 机械工业出版社

地 址/ 北京市西城区百万庄大街22号 (邮政编码 100037)

印 刷/ 北京诚信伟业印刷有限公司

开 本/ 178毫米×233毫米 16开本 49.5印张

版 次/ 2016年6月第1版 2016年6月第1次印刷

定 价/ 139.00元 (册)

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)88379426; 88361066

购书热线: (010)68326294; 88379649; 68995259

投稿热线: (010)88379604

读者信箱: hzit@hzbook.com

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

译者序

移动应用开发作为当今最热门的应用开发方向，已经受到全世界开发者的极大关注。一方面，在主流的移动平台iOS中，Swift语言作为这一领域的新贵，无疑会在未来几年赚足眼球。对于原有的iOS开发人员，有喜也有悲。欢喜的是，开发人员终于有了一门更加优秀、更加现代化、更加高性能的编程语言。将Swift作为武器，可以高效地开发出高质量的移动应用。悲伤的是，如果开发人员现在还不开始学习Swift，恐怕用不了多久就会被Swift带来的技术革新无情淘汰。另一方面，iOS本身也在快速地进行技术演进。不同以往，iOS 8引入和开放了许多别出心裁的新技术，如Extensions、HealthKit、HomeKit、TouchID，等等。如果开发人员能够充分利用这些新技术，就可以显著改善应用带给用户的使用体验，进而在移动应用时代取得制高点。在这个时候，我深深地体会到拥有一本优秀教材的重要性。

第一次拿到本书的原版作品时，既兴奋又倍感压力。一方面，本书既介绍了iOS，又介绍了如何用Swift进行实战，与当今的市场需求非常契合。另一方面，随着深入阅读，我深深地被原作者的睿智和才华所打动。本书的内容完整详实，书中例子尽可能追求简单，并切中要害。既可以作为教材供在校师生使用，也可以作为工具书，供一线工程师备查。如何将这样一部优秀的作品尽可能快并且尽可能高质量地呈现给国内的读者，对我们来说是一个前所未有的巨大挑战。

作为美团移动团队的成员，我们都是奋战在一线的iOS开发工程师，但是翻译图书对我们来说仍然是个巨大的挑战。首先是时间，我们需要利用业余时间和尽可能多的碎片时间进行本书的翻译工作，深夜还经常看到小伙伴们仍然在奋笔疾书。不仅如此，我还有些许忧虑，担心自己把握不好原著恰到好处的笔锋，不能有效地将这样一部优秀的作品

资料，力求做到专业词汇准确权威，将原书的精华呈现给每一位读者。

现在，我怀着期盼和忐忑的心情，将这本译著呈现给大家，渴望得到您的认可，更渴望与您成为朋友，如果您有任何问题和建议，请与我联系（liangshixing@gmail.com），让我们一起探讨，共同进步！

感谢机械工业出版社华章公司的陈佳媛老师对我们的信任与支持。感谢（美团）酒店旅游事业群负责人陈亮建立了这支充满技术热情的移动团队。感谢另外两位主要译者——我的多年好友陈作君以及团队主管赵萌——几个月以来的共同努力。感谢北京航空航天大学的宋友老师和杨思宇同学，他们为本书的定稿和审读提出了大量有建设性的宝贵意见，并做出了许多有创新性的指导工作。感谢团队其他小伙伴，郝田田、王禹华、吴卓和张杰，对本书翻译审读工作的鼎力相助。最后，还要特别感谢家人的理解与包容。

梁士兴

2016年6月于北京

前言	1
第1章 基础知识	9
1.0 介绍	9
1.1 为视图添加模糊效果	17
1.2 使用popover展示临时信息	20
1.3 使用UIImageView显示图片	24
1.4 使用UILabel显示静态文本	28
1.5 使用UIButton向UI中添加按钮	34
1.6 显示警告视图或者操作列表	38
1.7 使用UISwitch创建、使用、自定义开关	43
1.8 使用UIPickerView选取值	47
1.9 使用UIDatePicker选择日期或时间	51
1.10 使用UISlider实现区间滑块	55
1.11 使用UISegmentedControl实现分组的紧凑选项	59
1.12 使用UIActivityViewController显示分享选项	62
1.13 使用UIActivityViewController展示自定义分享选项	67
1.14 在导航栏上显示图片	72
1.15 使用UIBarButtonItem向导航栏添加按钮	73
1.16 使用UITextField接受用户文本输入	78
1.17 使用UITextView显示多行文本	85
1.18 使用UIScrollView创建可滚动的内容	89
1.19 使用WebKit载入网页	92

1.21 使用UIProgressView显示进度	96
1.22 创建配置文件	100
第2章 扩展	107
2.0 介绍	107
2.1 为照片应用添加新的照片编辑能力	110
2.2 向iOS提供自定义分享扩展	117
2.3 构建自定义键盘	126
2.4 通过操作扩展，在应用内部提供服务	132
2.5 向通知中心添加小工具	139
第3章 使用HealthKit管理健康数据	149
3.0 介绍	149
3.1 设置应用以使用HealthKit	149
3.2 获取和修改用户体重信息	152
3.3 访问和修改用户身高信息	158
3.4 获取用户特征	165
3.5 观察用户健康信息变化	169
3.6 读写用户燃烧的总卡路里值	174
3.7 单位转换	186
第4章 使用HomeKit管理家用电器	189
4.0 介绍	189
4.1 模拟HomeKit配件	191
4.2 在HomeKit中管理用户住宅	197
4.3 向用户的住宅添加房间	204
4.4 在用户住宅中指定区域	208
4.5 发现和管理支持HomeKit的配件	213
4.6 与HomeKit配件进行交互	218
4.7 对HomeKit配件进行分组	225

5.0	介绍	228
5.1	向UI组件添加重力效果	229
5.2	在UI组件之间检测并响应碰撞	231
5.3	使用推动 (push) 实现UI组件动画	237
5.4	对动力项目进行关联	241
5.5	向UI组件添加“跳动”效果	245
5.6	为动力特效设定特征	248
第6章	表格视图和集合视图	252
6.0	介绍	252
6.1	填充表格视图数据	253
6.2	为表格视图的单元格添加滑动删除功能	257
6.3	在表格视图中创建页眉和页脚	258
6.4	在表格视图中显示刷新控件	265
6.5	为集合视图提供基本内容	268
6.6	使用.xib 文件填充集合视图的自定义单元格	271
6.7	处理集合视图中的事件	278
6.8	为集合视图提供页眉和页脚	281
6.9	为集合视图添加自定义交互	285
第7章	并发和多任务	289
7.0	介绍	289
7.1	执行UI相关任务	292
7.2	执行非UI相关任务	294
7.3	在特定延时之后执行任务	301
7.4	对某个任务仅仅执行一次	303
7.5	将任务进行分组	305
7.6	使用操作 (NSOperation) 创建简单并发	307
7.7	在操作间创建依赖关系	312
7.8	触发周期性任务	314
7.9	在后台执行长时间运行的任务	317

7.12	在后台处理位置变化	331
7.13	在后台处理网络连接	333
第8章	安全	336
8.0	介绍	336
8.1	使用Touch ID验证用户	342
8.2	在应用中启用安全和保护功能	344
8.3	在钥匙串中存储数据	348
8.4	在钥匙串中查找值	350
8.5	更新钥匙串的已有值	353
8.6	在钥匙串中删除已有值	356
8.7	在多个应用中共享钥匙串数据	358
8.8	对iCloud中的钥匙串数据进行读写	362
8.9	在应用沙盒中安全存储文件	365
8.10	用户界面安全	368
第9章	Core Location、iBeacon和地图	370
9.0	介绍	370
9.1	检测用户在建筑中所在楼层	370
9.2	iBeacon介绍和处理	371
9.3	定位设备位置	377
9.4	在地图上显示大头针	382
9.5	在地图上添加自定义大头针	385
9.6	在地图上进行搜索	389
9.7	在地图上进行导航	394
9.8	使用摄像头自定义地图视图	400
第10章	手势识别	403
10.0	介绍	403
10.1	检测轻扫手势	405
10.2	检测旋转手势	407

10.4	检测长按手势.....	411
10.5	检测轻击手势.....	414
10.6	检测捏合手势.....	416
10.7	检测屏幕边缘拖曳手势.....	418
第11章	网络与分享	420
11.0	介绍.....	420
11.1	使用NSURLSession下载数据.....	420
11.2	通过NSURLSession在后台下载数据	427
11.3	通过NSURLSession上传数据.....	430
11.4	通过NSURLConnection进行异步下载	432
11.5	处理异步连接中的超时.....	436
11.6	通过NSURLConnection同步下载.....	437
11.7	自定义URL请求.....	439
11.8	通过NSURLConnection发送HTTP请求	440
11.9	序列化和反序列化JSON对象.....	444
11.10	将社交分享整合到应用中	447
第12章	多媒体.....	451
12.0	介绍	451
12.1	播放音频文件.....	451
12.2	录制音频	453
12.3	播放视频文件.....	460
12.4	从视频文件中获取缩略图.....	464
12.5	访问音乐库	466
第13章	通讯录.....	474
13.0	介绍	474
13.1	在系统界面上获得联系人记录	476
13.2	在系统界面上获得联系人信息	479
13.3	请求访问通讯录数据库.....	482
13.4	获取通讯录中全部联系人	484

13.7	在通讯录中插入群组	491
13.8	添加联系人到群组	493
13.9	查找通讯录	496
13.10	获取和设置通讯录联系人的图片	498
第14章 文件与文件夹管理		500
14.0	介绍	500
14.1	获得磁盘上最常用的文件夹路径	502
14.2	对文件进行读写操作	504
14.3	在磁盘中创建文件夹	508
14.4	枚举文件和文件夹	510
14.5	删除文件和文件夹	515
14.6	将对象保存到文件中	518
第15章 摄像头和照片库		521
15.0	介绍	521
15.1	探测和探查摄像头	522
15.2	用摄像头拍摄照片	526
15.3	用摄像头拍摄视频	530
15.4	在照片库中存储照片	533
15.5	在照片库中存储视频	538
15.6	检索并获取图像和视频	540
15.7	响应图像和视频中的变化	544
15.8	编辑设备中的图像和视频	549
第16章 通知		556
16.0	介绍	556
16.1	发送通知	557
16.2	监听并响应通知	559
16.3	监听并响应键盘通知	562
16.4	安排本地通知	570

16.6	处理本地系统通知	577
16.7	为应用设置推送通知	580
16.8	向应用发送推送通知	585
16.9	响应推送通知	592
第17章 Core Data		594
17.0	介绍	594
17.1	在Core Data中执行批量更新	595
17.2	向Core Data写入数据	598
17.3	从Core Data中读取数据	600
17.4	从Core Data中删除数据	602
17.5	在Core Data中对数据排序	605
17.6	在表格视图中提升数据访问	606
17.7	在Core Data中实现关系	614
17.8	在后台获取数据	619
17.9	在Core Data模型中使用自定义数据类型	623
第18章 日期、日历和事件		628
18.0	介绍	628
18.1	构建日期对象	629
18.2	获取日期组件	630
18.3	请求日历访问权限	631
18.4	从iOS设备上获取日历组	636
18.5	向日历中添加事件	637
18.6	访问日历内容	642
18.7	从日历中删除事件	644
18.8	向日历中添加重复事件	647
18.9	获取事件的出席者	652
18.10	向日历中添加提醒	654
第19章 图形和动画		657
19.0	介绍	657

19.2	绘制图像	665
19.3	构造可变尺寸图像	667
19.4	画线	671
19.5	构造路径	677
19.6	绘制矩形	681
19.7	为形状添加阴影	684
19.8	绘制渐变	689
19.9	变换视图	694
19.10	进行视图动画	699
第20章	Core Motion	708
20.0	介绍	708
20.1	获取高度数据	709
20.2	获取计步器数据	710
20.3	侦测加速计是否可用	714
20.4	侦测陀螺仪的可用性	716
20.5	获取加速计数据	717
20.6	侦测iOS设备的摇动	720
20.7	获取陀螺仪数据	721
第21章	Cloud	724
21.0	介绍	724
21.1	在您的应用中使用CloudKit	726
21.2	使用CloudKit储存数据	730
21.3	使用CloudKit获取数据	737
21.4	使用CloudKit查询Cloud内容	743
21.5	观察CloudKit中记录的变化	748
21.6	从CloudKit中获取用户信息	756
21.7	在iCloud中储存和同步字典	762
21.8	在iCloud中创建并管理文件和文件夹	766
21.9	在iCloud中搜索文件和文件夹	769

大约一年前，我注意到苹果公司不再像过去几年那样大幅更新Objective-C，我得到暗示，苹果可能正在为iOS开发研究新的语言或框架。我和工作上的伙伴交流了这一想法，他们笑着说：“你的书得重写了。”还真让他们说对了，这一版的内容几乎是全新的。

本书的上一版新增了许多知识点，还根据iOS 7更新了所有的Objective-C代码，这看起来已经是个非常庞大的工程了。然而，那些工作量和这一版相比则显得微不足道：因为所有内容都得用Swift重写。除此之外，新知识点数不胜数。我敢肯定这一版是初版以来付出心血最多的一版。所有的代码都用Swift重写，不是简单的逐行翻译，而是要充分利用Swift的优良特性，比如extensions。

没有人料到苹果公司会在其2014年的世界开发者大会（WWDC）上推出Swift。大家都以为会像以往几届一样，发布大量的新API和一些Objective-C增强特性。不过这次给了我们一个惊喜，至少对我来说是个惊喜。

我认为Swift是一门伟大的语言，并被iOS开发界期待已久。我们这些伴随着初版iOS SDK（也就是所谓的iPhone SDK）成长的人，深知手动管理引用计数的痛苦。本书早期版本里对这些概念的解释，现如今已显得无用且碍手碍脚。我们不得不花费太多精力在避免应用因内存管理问题而崩溃，无法专注于开发一个优秀的应用。Swift解决了大量诸如此类的问题，让我们有更多精力处理更复杂的问题。

Swift并非仅适用于iOS开发，因为它的许多特性在普通应用里没有充分发挥，但是在某些复杂或者高性能场景下，比如游戏开发，Swift则显得尤为合适。当进行iOS开发时，框架似乎比语言更为重要，这也是开发人员经常遇到困难的地方。包括苹果在内的许多

少卡路里，有多重，颜色是橙色，产地是哪。而本书会告诉你如何用它做出一锅美味的萝卜汤。

苹果公司没有提供如何使用其API的基本指南，但这也是业界普遍现象。把所有API记录在文档中是一个巨大的工程，苹果已经做得很出色了。本书会帮助你使用这些API向客户交付惊艳的应用。

希望读者会喜欢本书，如果有什么地方没解释清楚，可以通过Facebook、Twitter或者邮件联系我，能为广大开发者提供帮助，我将无比荣幸。

读者对象

本书假定你了解Xcode的使用，并写过少量Swift代码。本书并非Swift语言入门教程，而是展示如何用Swift编写强大而有趣的iOS应用，这两者有很大的区别。本书不会具体讲解Swift语言，因为苹果已经通过长达500页的《Swift Programming Language Guide》进行了非常透彻的讲解。因此没有必要进行重复劳动。如果觉得Swift用着还不够顺手，不妨先读一读这本指南，苹果公司免费提供，上网搜索即可。

本书同样不会讲解最基础的iOS开发。希望你已经了解基础的软件工程和算法理论。本书的目标并非帮助你成为一名软件工程师。如果你已经了解函数、栈、数组、字典或者散列表等，本书很适合你。否则的话，建议你先成为某领域的软件工程师（即使语言并非Swift），然后再重拾本书，学习如何编写惊艳的iOS应用。

本书结构

每章内容概要如下：

第1章 基础知识

该章阐述iOS开发基本组成部分，例如消息、标签、滑块、按钮、文本框以及文本视图、导航栏等。建议阅读该章内容后实际试用一下，然后再进入更高级的主题。

第2章 扩展

苹果公司终于允许扩展iOS了！把应用附带的扩展程序整合进iOS而独立存在，无需应用在后台运行。例如，现在可以创建自定义键盘，并安装到用户的设备上。即使应用没有运行，用户也可以使用自定义键盘。在Android系统中，这些特性一开始就得到了很好的支持。所以，当苹果为iOS提供这项特性后，我没有说“噢，太棒了”，而是说“终于啊”。阅读该章并且思考它对你的价值吧。

HealthKit允许iOS应用与存储在用户设备上的健康信息进行集成。这些信息属于设备中当前的用户，可以包含非常具体的内容，比如用户上次跑步消耗的脂肪总量。该章讨论如何在应用中集成HealthKit，并对健康数据库进行读写。

第4章 使用HomeKit管理家用电器

HomeKit是SDK中的另一个令人拍案叫绝的框架。它允许iOS应用与开启HomeKit的外设进行对话。该章讨论如何发现这些外设，进行配置，并与之通信，等等。

第5章 创建有动态交互性的用户界面

创建一个生动的用户界面远不止一个表格视图和导航栏的一个标签。它需要模拟现实世界的物理规律。该章介绍如何模拟重力以及其他物理行为，并将这些行为附加到UI组件。

第6章 表格视图和集合视图

展示给用户的许多信息都能以结构化形式在屏幕上的不同单元中显示出来。表格视图和集合视图在iOS中被大量使用，从“照片”到“设置”等应用，它们随处可见。该章讨论如何使用SDK中的组件来实现非常实用的功能。

第7章 并发和多任务

当应用运行时，默认情况下应用委托在主线程上执行UI相关的任务。但是不要在UI线程上执行繁重的下载和计算任务，因为那样会毁掉用户体验。事实上，当UI线程被阻塞超过5秒时，iOS会杀掉应用进程。该章讨论如何使用并发与多线程技术让应用流畅运行，完成一切所需操作而不过多占用UI线程。

第8章 安全

还在用NSUserDefaults存储用户名和密码吗？如果答案是“是的”，那么义无反顾地读一下这章吧。该章会讨论有关Touch ID认证和Keychain的功能。还会讨论如何使用应用的用户接口更加安全。

第9章 Core Location、iBeacon和地图

当用户想找到去往超市的路或者当前处于建筑的第几层（严格地说，iOS总能知道这些信息），iOS的众多传感器可以派上用场。因此，本章将讨论关于iBeacon、地图，以及Core Location的知识点。

第10章 手势识别

当史蒂夫·乔布斯介绍iPhone时，他向世界展示了如何在iPod音乐库中进行滚动，只需要上下滑动页面。我仍然记得当时人们的掌声和欢呼声。苹果公司的工程师将滑动手势加入视图中，这才使乔布斯先生能如此平滑地滚动页面。该章将讨论手势相关的知识。

iOS设备在设备与Internet连接的时候是什么？只是一部普通的手机或者一台平板电脑罢了。网络连接真的给予了智能机太多。该章将讨论如何使用SDK中提供的一些类，在后台和前台状态进行网络下载或上传文件。

第12章 多媒体

在iOS应用的内部，一旦获得了用户的许可，就可以访问他们的音频和视频文件，并且为用户播放这些文件，或者单纯抓取这些文件的数据，然后进行任意想要的处理。比如，创建一个游戏，可能会希望为用户播放一些背景音乐以增加游戏的吸引力。该章将讨论如何载入并播放音频和视频文件。

第13章 通讯录

通讯录框架仍然由C语言API组成。有人说这是出于性能考虑，但是我坚信苹果公司只是为这个框架设置了很低的优先级，还没有将它用SDK的最新技术进行升级。所以，该章会介绍如何使用Swift将通讯录框架集成到应用中，从而可以访问用户的联系人信息，当然，前提是用户授予了访问的权限。

第14章 文件与文件夹管理

通常编写iOS应用无需烦琐的操作文件和文件夹。但是，如果需要在文件中存储信息并且将文件用文件夹分类存放，可以开始阅读该章。该章会讨论如何写入文件、读取文件、遍历文件，等等。

第15章 摄像头和照片库

iOS为用户保留了一个包含着设备中所有照片和视频的库。允许用户查看这些照片和视频的应用叫作照片（Photos）。该章将讨论如何访问存放在设备中的照片和视频的原始数据，以便可以将此功能集成到应用中，而无需让用户再切换到照片应用。

第16章 通知

iOS中的不同部分之间会以通知来进行交互。例如，当应用进入后台时，iOS会向应用的内存空间发送一条通知。开发者可以在应用内的任何对象中捕获这条通知，来得知应用已经进入后台，进而做出必要的处理。该章会讨论所有关于本地、推送以及应用通知的知识。

第17章 Core Data

Core Data为苹果的数据库技术。你可以存储、读取、排序以及展示数据，并为不同的数据部分创建关系，等等。有什么理由不喜欢它呢？Core Data是一项出奇简便的技术，但仍需要开发人员对其底层的架构有所了解。该章便会带你了解这些内容。

第18章 日期、日历和事件

日期（Date，双关语，可指日期或约会）很重要，无论是讨论可编辑的日期，或是

可以在该章了解到日历中的日期。同时该章会讨论如何构造日期对象，以及如何从用户日历中读取事件，等等。

第19章 图形和动画

开发人员时刻都想通过酷炫的图形和动画效果来打动用户。这正是该章的目的。你可以在屏幕上绘制图像，加入动画，进行旋转，等等。让用户们赞叹吧。

第20章 Core Motion

计步器是非常了不起的设备，可以对用户的行走步数进行实时计数。一旦获得了用户的记步数据，并了解到用户年龄及其他一些简单的信息，就可以计算出用户所消耗的卡路里数，并向他们展示激励信息，等等。该章会讨论有关计步器、加速计和陀螺仪这些极为有用的传感器，苹果公司已将它们内置于如今市面上的绝大部分iOS设备之中。

第21章 Cloud

想象一下，你能够像在Core Data中一样轻松地在云端存储数据。CloudKit便完美地做到了这一点。它是在iCloud之上的一层。该章将要讨论关于iCloud的知识。

更多资源

Swift是一门相对较新的语言，关注iOS开发的读者很可能想要熟悉这门语言。正如我前面提到过的，推荐你阅读苹果公司的Swift编程语言指南。读过之后，你一定不会对它失望的。

笔者会经常引用苹果公司的官方文档。其中一些苹果公司的描述说得很到位，重复描述它们实在没有意义。苹果公司官方文档中最重要文档和指南贯穿本书，每个专业的iOS开发人员都应该阅读它们。

作为开始，建议阅读针对各种iOS设备的“iOS Human Interface Guidelines” (<http://bit.ly/QbdY0B>)。想要为各种iOS设备开发迷人且交互性好的用户界面，这份文档已经包含了一切。每个iOS开发人员都应该阅读这份文档。事实上，我相信任何开发iOS应用的公司都应该要求产品设计和开发团队阅读这份文档。

另外建议快速阅读iOS Developer Library (<http://bit.ly/Qi7JaZ>) 中的“iOS App Programming Guide”，其中包含了一些对于开发高质量应用很有价值的提示和建议。

以下是本书的印刷惯例：

斜体

用于标识新条款，URLs，email地址，文件名和文件扩展名。

等宽字体 (Constant width)

用于程序列表，同时还用于在段落中引用程序元素，比如，变量名或函数名，数据库，数据类型，环境变量，状态和关键字。

等宽粗体 (Constant width bold)

显示命令或其他文本，需要用户逐字键入。

等宽斜体 (Constant width italic)

展示的文字应该被用户提供的值或者上下文中提供的值所替换。



这个图标表示一条提示，建议，或者一般性的注解。



这个图标表明警告或需要格外注意。

使用本书的例子代码

辅助材料（例子代码，练习，等等）可以通过进行下载。

本书的目的是帮助读者解决任务。一般来说，如果本书提供的例子代码，读者可以在程序和文档中使用。而无需联系我们申请版权，除非是对包含了代码主要部分的二次发表。例如，编写程序，包含多处本书代码的片段，无需申请授权。但是，出售和分发来自 O'Reilly图书的示例的CD-ROM，则需要申请授权。回答本书提出的问题，并且引用例子代码无需申请授权。在产品文档中包含了本书的大部分例子代码，则需要申请授权。

我们会很感激，但不强制要求，在引用本书内容时，提供本书的引用出处。引用出处通常包含：标题，作者，出版社，和ISBN。

Safari在线图书

Safari在线图书 (www.safaribooksonline.com) 是一个随选的数字库, 用图书和视频两种形式, 讲述了来自世界行业领导者的技术和商业专业知识。

技术专家、软件开发者、网站设计者和创业者都将Safari在线图书作为他们进行搜索、解决问题、学习和证书培训的首选资源。

Safari在线图书为企业、政府代理和个人进行产品融合和系统定价提供了基础知识。用户已经访问了许多图书、培训视频和全文检索数据库中已正式出版的原稿, 这些原稿来自很多出版者, 例如, O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FTPress、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology等, 关于Safari在线图书的更多信息, 请在线联系我们。

如何联系我们

美国:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国:

北京市西城区西直门南大街2号成铭大厦C座807室 (100035)
奥莱利技术咨询(北京)有限公司

我们为本书提供了网页, 该网页上面列出了勘误表、范例和任何其他附加的信息。您可以访问如下网页获得:

<http://oreil.ly/HP-Drupal>

要询问技术问题或对本书提出建议, 请发送电子邮件至:

bookquestions@oreilly.com

致谢

Andy Oram是本书的编辑，他的表现一如既往的优异。他以光的速度进行工作，为的是尽早把本书分享给读者。他逐字逐句地检查了本书的内容，使本书能够适合更多的读者。这里感谢他的辛勤劳动。同样要感谢Niklas Saers，他进行了大量的工作，在技术方面对本书进行了审读。

O'Reilly出版社的Rachel Roumeliotis同样提供了大量重要的帮助。她对我的工作始终给予支持。当我决定用Swift重新编写本书时，她非常高兴。所以，在做出这个重大决定时，非常感谢她能站在我这边。

感谢Sara，我可爱的搭档，感谢她在我写本书时付出的耐心。我简直不敢相信会有这样可爱和有耐心的搭档。我衷心地感谢她在这段时间为我做的一切。

还要感谢来自O'Reilly出版社的Heather Scherer和Amy Jollymore。他们帮助梳理了本书相关问题和即将到来的系列视频。

感谢Ulla、Leif、Bella和陪伴我度过每一秒的孩子们。这段时间对我意味着很多，我会永远感激。最后但并非不重要，我要感谢Molly的参与和支持，尤其是她每天和我一起散步时带给我的迷人的脸庞。你总是抽时间陪伴我，纵然你自己也有许多的工作，我始终深爱着你。“Dukhtig tjej”！

1.0 介绍

为了编写iOS应用，我们需要了解Swift编程语言基础。Swift是苹果在Xcode 6和iOS 8 SDK中引入的全新编程语言，也是贯穿本书使用的语言。对象和类是许多面向对象编程语言的基本要素，如Swift、Objective-C、Java、C++等。

一切iOS应用本质上都使用了MVC（模型-视图-控制器）架构。从架构的视角看，每个iOS应用都包含模型、视图和控制器这三个主要组件。

模型是应用的大脑。它进行计算，并在视图和控制器之外创建了一个虚拟世界。简言之，模型是应用的虚拟拷贝，只是没有外观。

视图是应用程序与用户交互的窗口。它在大部分时间里负责显示模型内部的东西，但视图的职责不仅如此，它还接受用户的交互。任何用户与应用进行的交互，都先发送给视图，然后会被视图控制器所捕获，并最终发送给模型。

在iOS编程中，控制器通常指的是前面提到的视图控制器。可以将视图控制器看作模型和视图之间的一座桥梁。控制器根据桥梁某一侧发生的事件，按需对另一侧进行修改。打个比方，如果用户修改了视图里的内容，控制器要保证在响应时更新模型；如果模型获取到新的数据，控制器也会通知视图做出相应的变化。

本章将介绍如何创建一个iOS应用的结构，以及如何使用视图和视图控制器创建直观的应用程序。

同时，笔者希望介绍少量的Swift语言基础，但是在开始之前，需要格外强调一点：本

籍，用于介绍Swift的用法。考虑到你可能同时使用本书和其他资源学习Swift，还是帮助你简单温习一下基础知识吧。

在Swift中定义常量和变量

使用let关键字定义常量，像这样：

```
let integerValue = 10
let stringValue = "Swift"
let doubleValue = 10.0
```

赋给常量（或者赋给变量）的值有明确定义的类型。在示例代码中，由于Swift编译器能够在赋值时进行类型推断，因而不需要再定义常量的类型。然而，如果想要手动定义数据的类型，就需要使用如下方式。

```
let integerFromDouble = 10.7 as Int
/* 由于编译器会截断数值为整型，所赋的值变成了10*/
```

常量一旦被赋值，不能再改变。如果需要修改，应使用变量。用var关键字定义变量：

```
var myString = "Swi"
myString += "ft"
/* myString会变成"Swift" */
```

变量的值是可变的，常量的值是不可变的。常量不能被修改或者追加内容，而变量可以。

在Swift中创建和使用数组

[数据类型]语法可以用来创建数组。如下是一个不可变数组：

```
let allStrings = ["Swift", "Objective-C"]
```

如果要创建一个数组变量，可将其初始化为空数组之后再向其追加数据，就像示例中那样。使用var关键字，创建的“allStrings”数组是一个变量而不是常量：

```
var allStrings = [String]()
allStrings.append("Swift")
allStrings.append("Objective-C")
/* 这个数组此时是 ["Swift", "Objective-C"] */
```

如需访问数组中的值，使用“[]”标记的subscripting语法：

```
var allStrings = [String]()
allStrings.append("Swift")
```

```
println(allStrings[0]) /* 打印出 "Swift" */  
  
allStrings.insert("C++", atIndex: 0)  
  
println(allStrings[0]) /* 打印出 "C++" */
```

在Swift中定义和访问字典

字典是一种散列表。字典中的每条记录指定一个对象作为键，而另一对象作为与其对应的值。字典在Swift中的类型是动态的，它的类型取决于用什么类型的数据对它进行填充。“[key: value]”语义用于创建字典，示例如下：

```
let allFullNames = [  
    "Vandad" : "Nahavandipoor",  
    "Andy"   : "Oram",  
    "Molly"  : "Lindstedt"  
]
```

要访问某个键的值，使用下标语法，如下所示：

```
println(allFullNames["Vandad"]) /* 打印出 "Nahavandipoor" */
```

由于使用了let关键字，刚才创建的字典是不可变的。要创建内容与其相同的可变字典，需要使用var关键字，就像这样：

```
var allFullNames = [  
    "Vandad" : "Nahavandipoor",  
    "Andy"   : "Oram",  
    "Molly"  : "Lindstedt"  
]  
  
allFullNames["Rachel"] = "Roumeliotis"
```

该字典的类型为[String:String]，这是由键和值的具体类型所决定的。将任意类型的值或键加入字典中，观察其类型的变化：

```
let personInformation = [  
    "numberOfChildren" : 2,  
    "age"               : 32,  
    "name"              : "Random person",  
    "job"               : "Something cool",  
] as [String : AnyObject]
```

AnyObject类型，顾名思义，可以代表任意“类”类型的实例。在这个示例中，字典的键为字符串，而值的类型则可以混合多种“类”的类型。Swift中的字典和数组可以自由地与Cocoa Touch中的NSDictionary和NSArray做转换。

作者介绍

Vandad Nahavandipoor是一位iOS和OS X工程师，就职于一个范围覆盖29个国家、拥有超过7 000员工的国际传媒公司。此前他曾在英国劳埃德银行工作，为数百万英国用户提供他们的iOS应用程序。Vandad曾经领导了一支由30多个iOS开发者组成的国际团队，他所指导的项目包括NatWest和RBS在内的iOS应用，在英国数以百万计的iPhone和iPad上运行。Vandad获得了英国萨塞克斯大学电子商务信息技术学科的学士学位和硕士学位。

Vandad的编程体验始于第一次在父亲的Commodore 64上学习Basic。然后，他把这个体验应用在叔叔的英特尔186计算机上，在DOS下运行Basic。他发现在个人计算机上编程令人十分兴奋，随后就开始学习Object Pascal。这些使得他学习Borland Delphi变得非常容易。他使用Borland Delphi完成了一个简短的400页的书，并把这本书贡献给了宝蓝公司。从那以后，他拾起了x86汇编编程，出于个人兴趣，编写了一个名叫Vandior的32位操作系统。直到2007年年底，他才重点关注iOS编程。

除去编程，Vandad也是一名道路自行车手，喜欢英国的自行车友谊赛，并且经常长距离骑行到很多其他欧洲的国家。他还喜欢摆弄钢琴和电吉他。他的一些电吉他表演在YouTube上可以找到。

译者介绍

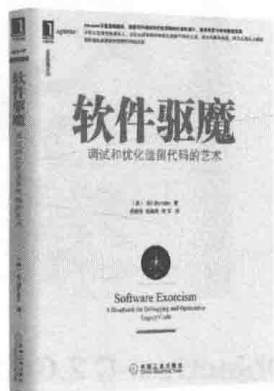
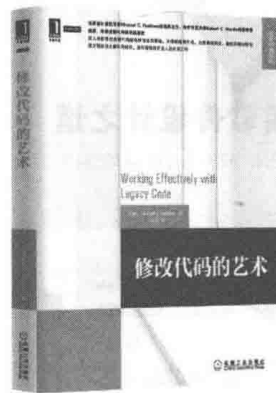
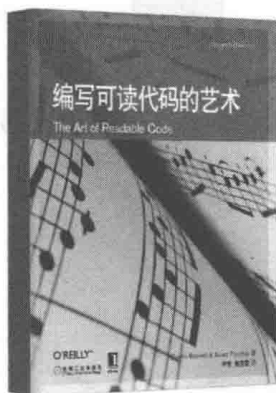
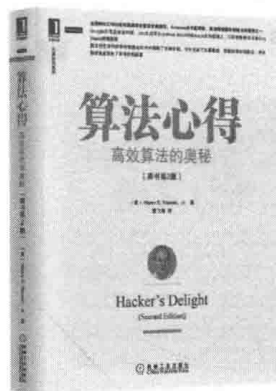
梁士兴，多年奋战在一线的iOS研发工程师，现就职于美团酒店旅游事业群，是终端iOS研发团队的负责人；长期专注iOS研发技术与架构设计，同时关注大前端方向的技术演进。

封面中的红嘴热带鸟（*Phaethon aethereus*）也被称作水手长鸟。热带鸟看起来像是燕鸥，但在遗传学上并没有联系。实际上，热带鸟没有非常相近的现存物种，这令它们的进化多了几分神秘色彩。2009年开始，红嘴热带鸟被选作了百慕大50美元钞票的图案，但随后便被本地的白尾热带鸟所替代，因为白尾鸟在百慕大更为出名。

红嘴热带鸟体型庞大，有长长的尾巴，身体为白色，同时有着向下弯曲的红色鸟喙。算上尾羽的话，差不多能有40英寸长。一米长的翼展可以让身体保持平衡，成为优雅的飞行者。它们的飞羽和眼睛上有黑色的斑。雄鸟和雌鸟看起来很相似，只不过雄鸟的尾巴会更长。红嘴热带鸟的脚非常靠近身体的后方，因此它们在陆地上移动起来非常滑稽笨拙，而腹部几乎贴到地面。它们也不是敏捷的游泳者，但可以舒适地盘旋在海洋上空，伺机捕捉飞鱼。飞鱼看起来是这些热带鸟的最爱，但它们同样也会吃其他的鱼甚至是头足类动物。

红嘴热带鸟生活在如加拉帕戈斯群岛、佛得角、西印度群岛，甚至是波斯湾等地。尽管它们偏爱温暖和热带海水，但有一类特定的红嘴热带鸟每年都会返回缅因州沿海的海豹岛。缅因州的这一带有大量的海鸟，但这一类则是只有在北部才能找到的种类。几年前，本地人摆放了一个木雕的热带鸟模型，有个拜访者竟然令人费解地尝试向它求爱并进行交配。而对于带领游客观看本地海鸭和黑海鸕的船主来说，参观这只鸟无疑是一桩非常好的生意。

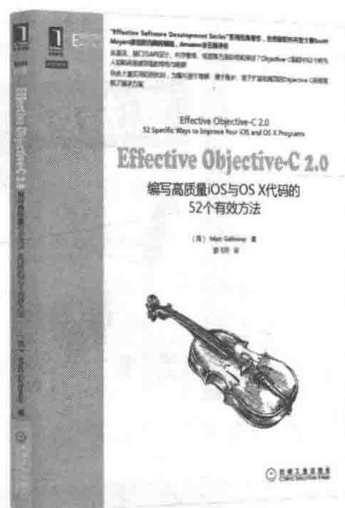
本书封面图片来自《Riverside Natural History》。





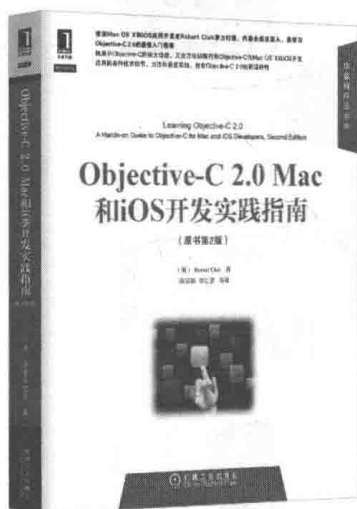
iOS 应用软件设计之道

作者: William Van Hecke ISBN: 978-7-111-47833-1 定价: 69.00元



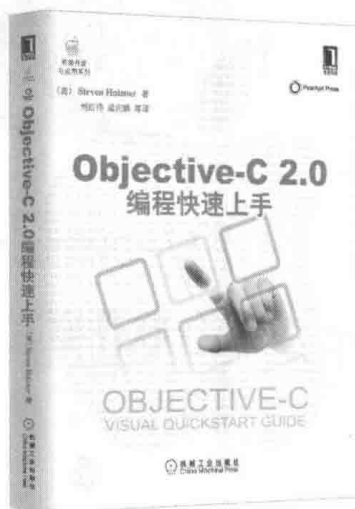
Effective Objective-C 2.0: 编写高质量iOS与OS X代码的52个有效方法

作者: Matt Galloway ISBN: 978-7-111-45129-7 定价: 69.00元



Objective-C 2.0 Mac和iOS开发实践指南 (原书第2版)

作者: Robert Clair ISBN: 978-7-111-48456-1 定价: 79.00元



Objective-C 2.0编程快速上手

作者: Steven Holzner ISBN: 978-7-111-30874-4 定价: 35.00元

Swift编程实战: iOS应用开发实例及完整解决方案

本书完全用苹果公司的Swift编程语言进行了重写, 读者在创建iOS应用时遇到的许多烦人问题, 都可以通过本书介绍的方法得到解决。

本书使用了iOS 8的SDK, 读者会在书中找到数百个全新的、改进过的知识点, 包括使用健康应用数据和HomeKit配件、增强的动画和绘图、存储和保护数据、收发通知、管理文件和文件夹, 等等。每节的示例代码都在Github上, 读者可以立即使用。

- 使用CloudKit API轻松地cloud中存储信息
- 创建自定义的键盘和扩展
- 使用HealthKit访问用户健康相关的信息
- 使用HomeKit与用户家中的配件设备进行交互
- 使用UIKit Dynamic创建生机勃勃、栩栩如生的用户界面
- 使用钥匙串保护用户的数据
- 开发位置和多任务相关的应用
- 使用iOS 8的音频和视频API进行工作
- 使用Event Kit UI管理日历、日期和事件
- 使用加速计和陀螺仪
- 获得手势识别的实用技巧
- 从地址簿获取和修改联系人及分组信息
- 检测相机是否可用并访问照片库

“从使用iOS的新特性进行编码到处理复杂的需求, 本书总是带给我所需要的资源和优秀、清晰、有实践性的示例。现在, 我开始使用Swift, 本书成为我的绝佳参考, 没有它我会迷失。”

——João Duarte

iOS高级软件工程师

Vandad Nahavandipoor 资深iOS和OS X工程师, 现就职于一个范围覆盖29个国家、拥有超过7000名员工的国际传媒公司。此前他曾在英国劳埃德银行工作, 为数百万英国用户提供iOS应用程序。



MOBILE DEVELOPMENT/IOS

O'Reilly Media, Inc. 授权机械工业出版社出版

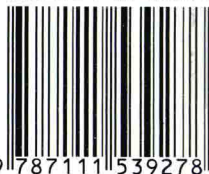
此简体中文版仅限于在中华人民共和国境内（但不允许在中国香港、澳门特别行政区和中国台湾地区）销售发行
This Authorized Edition for sale only in the territory of People's Republic of China (excluding Hong Kong, Macao and Taiwan)

投稿热线: (010) 88379604
客服热线: (010) 88379426 88361066
购书热线: (010) 68326294 88379649 68995259

华章网站: www.hzbook.com
网上购书: www.china-pub.com
数字阅读: www.hzmedia.com.cn

上架指导: 计算机/程序设计

ISBN 978-7-111-53927-8



定价: 139.00元