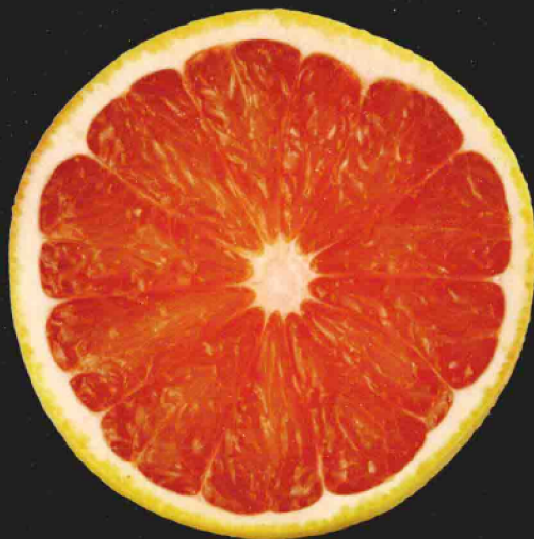


中文版累计销量逾50 000册!

全球数百万iOS开发者交口称赞的iOS开发圣经!



精通iOS开发

(第6版)

[瑞典] Jack Nutting [法] Fredrik Olsson
[美] David Mark [美] Jeff LaMarche

著

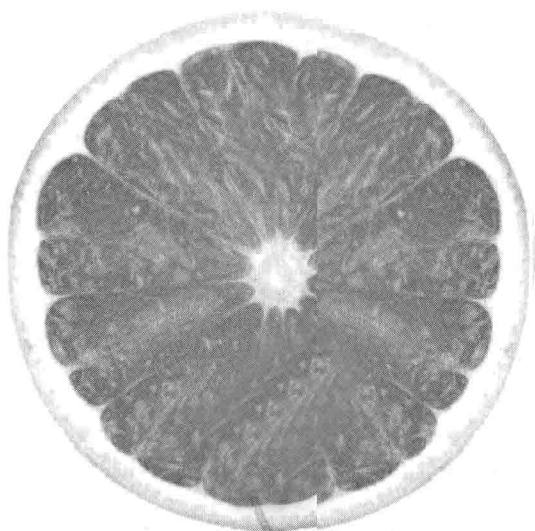
周庆成 邓强 武海峰 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书



精通iOS开发

(第6版)

[瑞典] Jack Nutting [法] Fredrik Olsson 著
[美] David Mark [美] Jeff LaMarche

周庆成 邓强 武海峰 译

人民邮电出版社

北京

图书在版编目 (C I P) 数据

精通iOS开发：第6版 / (瑞典) 纳丁 (Nutting, J.)
等著；周庆成，邓强，武海峰译. — 北京：人民邮电
出版社，2014.10

(图灵程序设计丛书)

ISBN 978-7-115-36826-3

I. ①精… II. ①纳… ②周… ③邓… ④武… III.
①移动电话机—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191738号

内 容 提 要

本书是 iOS 应用开发基础教程，内容翔实，语言生动。作者结合大量实例，循序渐进地讲解了适用于 iPhone/iPad 开发的基本流程。新版介绍强大的 iOS 7 操作系统，涵盖 Xcode 4 以来的新功能，书中所有案例全部重新编写。

本书具有较强通用性，iOS 开发新手可通过本书快速入门进阶，经验丰富的 iOS 开发人员也能从中找到令人耳目一新的内容。

-
- ◆ 著 [瑞典] Jack Nutting [法] Fredrik Olsson
[美] David Mark [美] Jeff LaMarche
译 周庆成 邓 强 武海峰
责任编辑 朱 巍
执行编辑 李 鑫
责任印制 焦志炜
- ◆ 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>
三河市中晟雅豪印务有限公司印刷
- ◆ 开本：800×1000 1/16
印张：34.25
字数：809千字 2014年10月第1版
印数：1-3 000册 2014年10月河北第1次印刷
- 著作权合同登记号 图字：01-2014-5227号

定价：109.00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

版 权 声 明

Original English language edition, entitled *Benginning iOS 7 Development: Exploring the iOS SDK* by Jack Nutting, Fredrik Olsson, David Mark, Jeff LaMarche, published by Apress, 2855 Telegraph Avenue, Suite600, Berkeley, CA94705 USA.

Copyright © 2014 by Jack Nutting, Fredrik Olsson, David Mark and Jeff LaMarche.Simplified Chinese-language edition copyright © 2014 by Posts&Telecom Press.All rights reserved.

本书中文简体字版由 Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

iOS 7 是 iOS 系统自 2007 年 iPhone 发布以来最大一次升级，由苹果公司在 2013 年 6 月 10 日举办的全球开发者大会（WWDC）上对外公布，其支持 iPhone 4 及以后机型、第 5 代 iPod touch、iPad 2 及以后机型，等等。

iOS 7 重新设计了用户界面并对操作系统的功能进行了改进，它采用全新的扁平化界面设计，总计有上百项改动，其中包括控制中心、通知中心、多任务处理能力等。iOS 7 还特别针对中国用户的输入习惯改进了中文输入法，增加了九宫格输入方式。就在翻译此书时，iOS 7 的安装率已经达到 91%，成为 iOS 设备上主流的操作系统。

iOS 7 在用户界面上有着与之前版本完全不同的视觉设计，应用程序的图标变得更锐利，在许多地方采用了较为纤细的字体，以往的拟物风格在 iOS 7 中不再出现。虽然扁平化风格界面并非苹果最早采用，但在视觉体验上，iOS 7 依然拥有自己的特色。新的界面色彩更为鲜艳，饱和度更加突出，通过渐变和半透明更好地阐释了苹果一贯的简单至上的设计原则。iOS 7 的画面采用类 3D 的效果，在锁定画面及主屏幕会有立体的效果。所有的内置应用程序、解锁画面与通知中心也经过重新设计。

此外 iOS 7 也新增了控制中心界面，让用户能够快速控制各种系统功能的开关（包括飞行模式、蓝牙、Wi-Fi 网络等）并调整屏幕亮度、播放或暂停音乐等。后台多任务处理功能也经过了强化，现在能够支持每一种应用程序，切换程序时也采用了新的交互方式。

本书全面介绍了 iOS 7 开发的基础知识，结构清晰，内容丰富，通过 22 章由浅入深涵盖了 iOS 7 所有的关键内容。与上一版相比，本书增加了游戏开发 SDK 教程，并对许多章节进行了大量的优化改进，其中第 9 章为了配合 iOS 7 的新特性重写了全部内容，从而能够更好地展示使用新系统开发的优势。

随着 iOS 的不断升级，这本书也迎来了第 6 版。我曾阅读过从本书最初到现在的所有版本，学习过程中难免会遇到一些翻译失当或难以理解的地方，也向图灵公司提出了一些勘误的意见。这次新版出来时，我向出版社自荐来改善这本书一些遗留的历史问题，希望让读者可以更轻松地理解原作者的想法，简化程序员向移动开发入门的难度。虽然在校订过程中修改了很多细节，尽可能地保证质量，但仍可能会有遗漏，希望读者能给出合理的意见。

这本书能够顺利翻译完成，首先要感谢我的家人对我的支持，以及出版社对我的信任。还要特别感谢图灵公司的编辑李鑫，他认真地对全稿进行了审阅。此外还要感谢好友蒋振华，他对书中许多内容提出了自己的看法。当然，还要感谢苹果公司所提供的如此便利的开发平台以及原作者的辛勤劳动。

——周庆成

关于本书

本书内容非常丰富。如果你读的是纸质书，就能明白我的意思，如果买的是电子书，就不容易发现这点。我只想说，它比我见过的各种经典书籍还要厚。当然，这并非是说，我是经典书籍或某方面的行家，但你们只需要明白它很大很重就对了。如果你想知道这本书是否比面包箱还大，我可以明确地告诉你，差不多真的有那么大。然而，本书结构划分得条理明确，各部分内容也十分简洁扼要，你完全可以在某个愉快的下午尽情揣摩某一章节的内容。这种“揣摩”并不仅仅指的是通过阅读来了解每章内容，还需要在你的 Mac 电脑上，用自己的方式来学习与实践本书的知识。如果在阅读每一章时，能够构建其中所有的示例应用，则有助于加深对使用模式和相关概念的理解，而光靠阅读显然是无法做到这一点的。如果完整通读此书，就会对 iOS 应用程序开发的基本概念有很好的理解，也就可以更好地构建自己的 iOS 应用了。

许多年前，我曾与 Torfrid Olsson（已故）见过一面，他是一位瑞典北方农村的雕刻家。对于他生活的某一方面，我表示羡慕和钦佩，但他的回答也令我不禁深思：“嗯，这些只是书上讲的。每个人的生活都是独特的，你怎么会认为你的生活是残缺的呢？”我希望你要善于利用书中的知识，不要满足于读懂。阅读当然是必要的，但也要实践和理解其中的内容，甚至要敢于质疑或推翻其中的内容，真正掌握这些知识。

——Jack Nutting
斯德哥尔摩，2014 年

致 谢

感谢 Apress 所有员工，正是由于他们的努力工作，才能完成这本如此厚实的书。他们的付出与帮助是至关重要的。尤其需要感谢 Anamika Panchoo，直到出版的最后阶段，她都在鉴定全书插图的正确性——要知道，当时她已经不用负责这方面工作了，大部分工作的负责人是 Melissa Maldonado。没有这些编辑勤奋耐心的校正，本书是不可能完成的。Patrick Meader 修改了大量的文法错误，而 Nick Waynik 则把杂乱的代码排版得井井有条。Tom Welsh、Douglas Pundick 和 Matt Moodie 三人轮流担当执行编辑，确保了每章内容条理分明，易于理解。当然，对于负责编排索引、页面排版、印刷零售（以及很多我不知道的工作）的工作人员来说，尽管你们经常被人忽视，我们也要衷心对你们表达莫大的感激之情。尽管身为作者，我们没能和你们直接接触过，甚至也不知道你们的名字，但我明白你们做的事都非常有意义，再接再厉！

目 录

第 1 章 欢迎来到 iOS 世界	1	第 3 章 实现基本交互	34
1.1 关于本书	1	3.1 MVC 方法	34
1.2 必要条件	1	3.2 创建项目	35
1.2.1 开发者选项	3	3.3 视图控制器	36
1.2.2 必备知识	3	3.3.1 输出接口和操作方法	37
1.3 编写 iOS 应用程序有何不同	4	3.3.2 清理视图控制器	40
1.3.1 只能同时运行一个应用	4	3.3.3 设计用户界面	40
1.3.2 只有一个窗口	4	3.3.4 运行应用	50
1.3.3 有限的访问权限	5	3.3.5 样式文本	50
1.3.4 有限的响应时间	5	3.4 应用程序委托	51
1.3.5 有限的屏幕大小	5	3.5 小结	54
1.3.6 有限的系统资源	5	第 4 章 更丰富的用户界面	55
1.3.7 不支持垃圾回收	6	4.1 满是控件的屏幕	55
1.3.8 新功能	6	4.2 活跃控件、静态控件和被动控件	57
1.3.9 与众不同的交互方法	6	4.3 创建应用程序	58
1.4 本书内容	7	4.4 实现图像视图和文本框	59
1.5 这一版的新内容	8	4.4.1 添加图像视图	59
1.6 准备开始吧	9	4.4.2 调整图像视图的大小	62
第 2 章 创建项目	10	4.4.3 设置视图属性	63
2.1 在 Xcode 中创建项目	10	4.4.4 添加文本框	65
2.1.1 Xcode 项目窗口	14	4.4.5 创建并关联输出接口	71
2.1.2 深入研究项目	22	4.5 关闭键盘	73
2.2 界面构建器简介	23	4.5.1 输入完成后关闭键盘	73
2.2.1 文件格式	24	4.5.2 通过触摸背景关闭键盘	74
2.2.2 分镜	25	4.5.3 添加滑动条和标签	77
2.2.3 库	26	4.5.4 添加顶部约束	78
2.2.4 在视图中标签	27	4.5.5 创建并关联操作方法和输出接口	79
2.2.5 属性修改	29	4.5.6 实现操作方法	79
2.3 画龙点睛——美化 iPhone 应用	31	4.6 实现开关、按钮和分段控件	80
2.4 小结	33		

4.6.1 添加两个带标签的开关	81	6.3.3 修改 BIDSwitchView Controller.m	120
4.6.2 为开关创建并关联输出接口和 操作方法	83	6.3.4 添加视图控制器	120
4.6.3 实现开关的操作方法	83	6.3.5 构建带有工具栏的视图	122
4.7 美化按钮	85	6.3.6 编写根视图控制器	124
4.7.1 可拉伸图像	85	6.3.7 实现内容视图	127
4.7.2 控件状态	86	6.3.8 转换过程的动画效果	130
4.7.3 为按钮创建并关联输出接口和 操作方法	87	6.4 小结	132
4.8 实现分段控件的操作方法	87	第 7 章 分页栏与选取器	133
4.9 实现操作表单和警告视图	88	7.1 Pickers 应用程序	134
4.9.1 遵从操作表单委托方法	88	7.2 委托和数据源	136
4.9.2 显示操作表单	89	7.3 创建 Pickers 应用程序	136
4.9.3 最终调整	92	7.3.1 创建视图控制器	136
4.10 小结	92	7.3.2 添加分镜	137
第 5 章 自动旋转和自动调整大小	93	7.3.3 创建分页栏控制器	138
5.1 自动旋转机制	93	7.3.4 初次运行	140
5.1.1 点、像素和 Retina 显示屏	94	7.4 实现日期选取器	141
5.1.2 自动旋转的实现方式	95	7.5 实现单滚轮选取器	144
5.2 选择视图支持的方向	95	7.5.1 构建视图	144
5.2.1 应用级支持的方向	96	7.5.2 将控制器实现为数据源和 委托	145
5.2.2 单个控制器的旋转支持	97	7.6 实现多滚轮取器	149
5.3 使用约束设计界面	99	7.6.1 声明输出接口和操作方法	149
5.3.1 覆盖默认的约束	101	7.6.2 构建视图	149
5.3.2 与屏幕等宽的标签	102	7.6.3 实现控制器	150
5.4 旋转时重构视图	104	7.7 实现内容取决于滚轮	152
5.4.1 创建并关联输出接口	106	7.8 使用自定义选取器创建一个简单 游戏	158
5.4.2 旋转时移动按钮	106	7.8.1 编写控制器头文件	158
5.5 小结	109	7.8.2 构建视图	159
第 6 章 多视图应用	110	7.8.3 添加图像资源	160
6.1 多视图应用的常见类型	110	7.8.4 实现控制器	160
6.2 多视图应用的体系结构	113	7.8.5 最后的细节	163
6.2.1 根控制器	115	7.9 小结	166
6.2.2 内容视图剖析	116	第 8 章 表视图简介	167
6.3 构建 View Switcher 项目	116	8.1 表视图基础	167
6.3.1 创建视图控制器和分镜	117	8.1.1 表视图和表视图单元	168
6.3.2 修改应用委托	119	8.1.2 分组表和无格式表	168

8.2 实现一个简单表	170	9.7 创建字体信息视图控制器	225
8.2.1 设计视图	170	9.7.1 设计字体信息视图控制器的 分镜	226
8.2.2 编写控制器	171	9.7.2 设置约束	227
8.2.3 添加一个图像	175	9.7.3 调整字体列表视图控制器的 转场	228
8.2.4 表视图单元样式	176	9.7.4 我的收藏字体	229
8.2.5 设置缩进级别	178	9.8 改善表视图	229
8.2.6 处理行的选择	179	9.8.1 实现轻扫删除	230
8.2.7 更改字体大小和行高	180	9.8.2 实现拖动排序	231
8.3 定制表视图单元	182	9.9 小结	232
8.3.1 向表视图单元添加子视图	182	第 10 章 集合视图	233
8.3.2 创建 UITableViewCell 子类	183	10.1 创建 DialogViewer 项目	233
8.3.3 从 nib 文件加载 UITableViewCell	187	10.2 修补视图控制器类	234
8.4 分组分区和索引分区	191	10.3 自定义单元	235
8.4.1 构建视图	191	10.4 配置视图控制器	237
8.4.2 导入数据	191	10.5 内容单元	239
8.4.3 实现控制器	192	10.6 实现流式布局	240
8.4.4 添加索引	196	10.7 分区标题视图	241
8.5 解决状态栏干扰	197	10.8 小结	243
8.6 实现搜索栏	198	第 11 章 iPad 开发注意事项	244
8.7 小结	203	11.1 分割视图和浮动窗口	244
第 9 章 导航控制器和表视图	204	11.1.1 创建 SplitView 项目	246
9.1 导航控制器	204	11.1.2 在分镜中定义结构	248
9.1.1 栈的概念	204	11.1.3 使用代码定义功能	250
9.1.2 控制器栈	205	11.2 显示总统信息	256
9.2 一个简单的字体浏览器: Fonts	206	11.3 创建浮动窗口	261
9.2.1 子控制器	207	11.4 小结	267
9.2.2 Font 应用的基础框架	209	第 12 章 应用设置及用户默认设置	268
9.3 创建根视图控制器	213	12.1 设置捆绑包入门	268
9.4 初始化分镜	216	12.2 应用: Bridge Control	269
9.5 第一个子控制器: 字体列表视图	218	12.2.1 创建项目	272
9.5.1 设定字体列表的分镜	220	12.2.2 使用设置捆绑包	273
9.5.2 对根视图控制器的转场进行 设置	221	12.2.3 读取应用中的设置	286
9.6 创建字体尺寸视图控制器	222	12.2.4 在应用中修改默认设置	289
9.6.1 设计字体尺寸视图控制器的 分镜	224	12.2.5 注册默认值	292
9.6.2 对字体列表视图控制器的转 场进行设置	224	12.2.6 保证设置有效	293
		12.3 小结	295

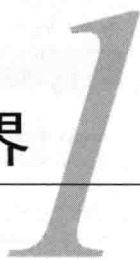
第 13 章 数据持久化基础知识	296	第 15 章 Grand Central Dispatch 和 后台处理	359
13.1 应用的沙盒	296	15.1 Grand Central Dispatch	359
13.1.1 获取 Documents 目录	297	15.2 SlowWorker 简介	360
13.1.2 获取 tmp 目录	298	15.3 线程基础知识	362
13.2 文件保存方案	298	15.4 工作单元	363
13.2.1 单文件持久化	299	15.5 GCD: 底层队列	364
13.2.2 多文件持久化	299	15.5.1 傻瓜式操作	364
13.3 属性列表	299	15.5.2 改进 SlowWorker	365
13.3.1 属性列表序列化	299	15.6 后台处理	370
13.3.2 Persistence 应用的第一个 版本	300	15.6.1 应用生命周期	371
13.4 对模型对象进行归档	305	15.6.2 状态更改通知	372
13.4.1 遵循 NSCoding 协议	305	15.6.3 创建 State Lab 项目	373
13.4.2 实现 NSCopying 协议	306	15.6.4 执行状态的变化	374
13.4.3 对数据对象进行归档和取消 归档	307	15.6.5 利用执行状态更改	376
13.4.4 归档应用	308	15.6.6 处理不活跃状态	377
13.5 使用 iOS 内嵌的 SQLite3	311	15.6.7 处理后台状态	380
13.5.1 创建或打开数据库	312	15.7 小结	387
13.5.2 绑定变量	313	第 16 章 使用 Core Graphics 绘图	388
13.5.3 SQLite3 应用	314	16.1 Quartz 2D 基础概念	388
13.6 使用 Core Data	320	16.2 Quartz 2D 绘图方法	388
13.6.1 实体和托管对象	321	16.2.1 Quartz 2D 图形环境	389
13.6.2 Core Data 应用	324	16.2.2 坐标系统	390
13.7 小结	333	16.2.3 指定颜色	391
第 14 章 iCloud 之旅	334	16.2.4 在环境中绘制图像	393
14.1 使用 UIDocument 管理文档存储	334	16.2.5 绘制形状: 多边形、直线 和曲线	393
14.1.1 构建 TinyPix	335	16.2.6 Quartz 2D 样例: 图案、渐 变色、虚线图	394
14.1.2 创建 BIDTinyPixDocument 类	336	16.3 QuartzFun 应用程序	394
14.1.3 主控制器代码	338	16.3.1 构建 QuartzFun 应用程序	395
14.1.4 初始分镜	345	16.3.2 添加 Quartz 2D 绘制代码	404
14.1.5 创建 BIDTinyPixView 类	347	16.3.3 优化 QuartzFun 应用程序	409
14.1.6 设计分镜	351	16.4 小结	412
14.2 添加 iCloud 支持	353	第 17 章 Sprite Kit 游戏框架	413
14.2.1 创建授权文件	353	17.1 基础入门	413
14.2.2 如何查询	354	17.1.1 自定义初始场景	414
14.2.3 保存在哪里	356	17.1.2 隐藏状态栏	415
14.2.4 将首选项保存到 iCloud	357		
14.3 小结	358		

17.2 场景设置	415	18.6 Swipes 应用	454
17.3 玩家动作	418	18.6.1 自动手势识别	457
17.3.1 向场景中插入玩家	418	18.6.2 实现多指轻扫	459
17.3.2 触摸处理	419	18.7 检测多次轻点	460
17.3.3 玩家移动	420	18.8 检测捏合和旋转	465
17.3.4 几何运算	421	18.9 自定义手势	469
17.3.5 轻微摆动	421	18.9.1 CheckPlease 应用	470
17.4 创建你的敌人	422	18.9.2 CheckPlease 的触摸方法	472
17.5 在场景中放入敌人	423	18.10 小结	474
17.6 开始射击	425	第 19 章 Core Location 和 Map Kit	475
17.6.1 定义物理类别	425	19.1 位置管理器	476
17.6.2 创建 BDBulletNode 类	425	19.1.1 设置精度	476
17.6.3 应用物理知识	427	19.1.2 设置距离筛选器	476
17.6.4 在场景中添加导弹	427	19.1.3 启动位置管理器	477
17.7 利用物理引擎攻击敌人	429	19.1.4 合理使用位置管理器	477
17.8 实现关卡	431	19.2 位置管理器委托	477
17.8.1 注意敌人	431	19.2.1 获取位置更新	477
17.8.2 进入下一关卡	431	19.2.2 使用 CLLocation 获取纬度和经度	477
17.9 自定义碰撞	433	19.2.3 错误通知	479
17.9.1 为 SKNode 添加类别	434	19.3 开始构建 Core Location	480
17.9.2 向敌人添加自定义碰撞行为	435	19.4 将移动路线展现在地图上	484
17.9.3 准确显示玩家生命	436	19.5 小结	488
17.10 粒子系统	437	第 20 章 陀螺仪和加速计	489
17.10.1 第一个粒子	437	20.1 加速计物理特性	489
17.10.2 向场景中加入粒子	440	20.2 陀螺仪旋转特性	490
17.11 游戏结束	441	20.3 Core Motion 和动作管理器	490
17.12 开始场景	443	20.3.1 基于事件的动作	491
17.13 播放音乐	445	20.3.2 主动动作访问	496
17.14 小结	446	20.3.3 加速计结果	498
第 18 章 轻点、触摸和手势	447	20.4 检测摇动	498
18.1 多点触控术语	447	20.4.1 内嵌的摇动检测	499
18.2 响应者链	448	20.4.2 摇动与击碎	500
18.2.1 响应事件	448	20.5 将加速计用做方向控制器	502
18.2.2 转发事件: 保持响应者链的活动状态	449	20.5.1 滚弹珠程序	503
18.3 多点触控体系结构	450	20.5.2 实现 BDBallView 类	505
18.4 4 个手势通知方法	450	20.5.3 计算弹珠运动	508
18.5 TouchExplorer 应用	451	20.6 小结	510

第 21 章 摄像头和照片库.....	511	22.2 字符串文件.....	521
21.1 图像选取器和 UIImagePickerController- Controller.....	511	22.2.1 字符串文件.....	522
21.2 实现图像选取器控制器委托.....	513	22.2.2 本地化的字符串宏.....	522
21.3 实际测试摄像头和照片库.....	514	22.3 现实中的 iOS 本地化应用.....	523
21.3.1 设计界面.....	515	22.3.1 创建 LocalizeMe.....	523
21.3.2 实现摄像头视图控制器.....	515	22.3.2 测试 LocalizeMe.....	526
21.4 小结.....	519	22.3.3 本地化项目.....	527
第 22 章 应用本地化.....	520	22.3.4 初始化分镜.....	530
22.1 本地化体系结构.....	520	22.3.5 创建并本地化字符串文件.....	532
		22.3.6 应用显示名称的本地化.....	535
		22.4 小结.....	536

第 1 章

欢迎来到 iOS 世界



这么说，你想编写 iPhone、iPod touch 和 iPad 应用程序？哦，当然可以。作为所有苹果设备的核心软件，iOS 从 2007 年发布以来发展异常迅速，是个极具吸引力的平台。移动应用平台的崛起意味着人们无时无刻不在使用各种软件。随着 iOS 7、Xcode 5 以及最新 iOS 软件开发工具包（SDK）的发布，开发 iOS 应用正变得更加高效、更加有趣。

1.1 关于本书

本书是 iOS 应用编程的入门指南，旨在帮助你克服入门的困难，帮助你理解 iOS 应用程序的运行和构建方式。

在学习的过程中，你将会创建一系列小型应用程序，每个应用程序都会突出某些 iOS 特性，展示如何控制这些特性，以及如何与其交互。如果你扎实地掌握了本书的基础知识，充分发挥自己的创造力，并且持之以恒，再借助大量条理清晰的苹果文档，你就可以创建出专业的 iPhone 和 iPad 应用。

提示 Dave、Jack、Jeff 和 Fredrik 为本书创办了一个论坛。志趣相投者可会聚于此，相互答疑解惑。一定要访问这个论坛哦！地址是 <http://forum.learncocoa.org>。

1.2 必要条件

开始编写 iOS 应用程序之前，需要做一些准备工作。初学者需要一台安装了 Mountain Lion（OS X 10.8）或 Mavericks（OS X 10.9）甚至更高版本的基于 Intel 架构的 Macintosh 计算机。任何最近上市的基于 Intel 架构的 Macintosh 计算机（台式机或笔记本）均可。

如果想访问到苹果更强大的全新开发工具，你需要注册成为 iOS 开发者。只需要访问 <http://developr.apple.com/ios/> 链接就可以创建你的开发者账号了。你会看到如图 1-1 所示的页面。

首先，点击 Log in 按钮，页面将提示你输入 Apple ID。如果没有 Apple ID，就点击 Join now 创建一个 ID，然后再登录。登录之后就进入了 iOS 开发中心的主页面。其中有各类文档、视频和示例代码等的链接，所有这些都可以帮助你更好地进行 iOS 应用开发。

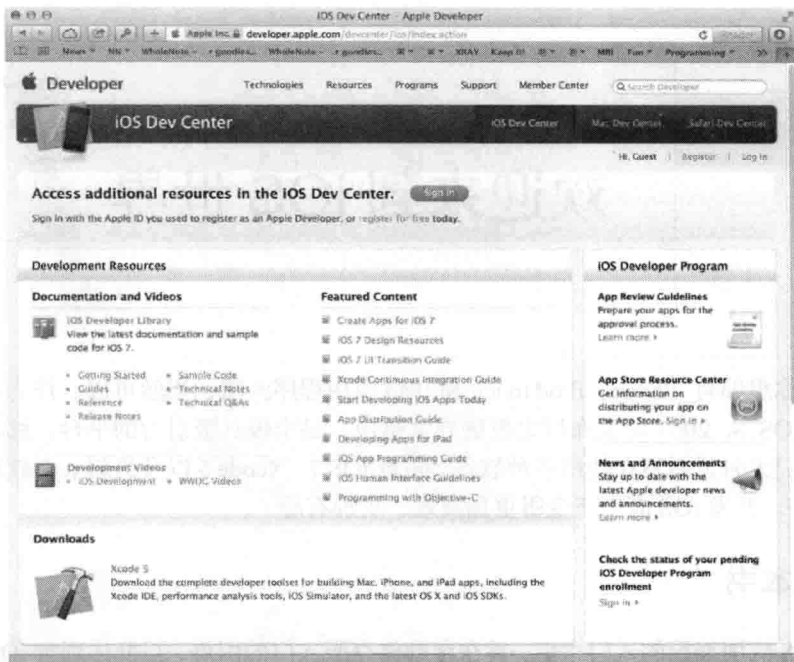


图 1-1 苹果的 iOS 开发中心网站

iOS 开发中最重要的工具是 Xcode，它是苹果的集成开发环境（IDE）。Xcode 提供了一些实用工具，用于创建和调试源代码、编译应用程序，以及对应用程序进行性能调优。

可以从 Mac App Store 下载 Xcode（可以通过 Mac 的 Apple 菜单访问 App Store）。

书中示例使用的 SDK 版本和源代码

随着 SDK 和 Xcode 版本的不断更新，它们的下载方式也在发生变化。在最近几年内，苹果已经开始把当前“稳定”版本的 Xcode 和 iOS SDK 放在 Mac App Store 中了，同时也会在开发者网站上提供预览版供开发者下载。总之，如果想下载 Xcode 和 iOS SDK 的最新版本（非 beta 版），应该使用 Mac App Store。

本书面向最新版 SDK。在一些地方，我们会使用 iOS 7 中引入的新函数或方法，它们可能与旧版 SDK 不兼容。当然，出现这些情况时我们会特别指出。

请从 <http://learn.cocoa.org> 或者本书论坛 <http://forum.learn.cocoa.org> 上下载最新的源代码^①。当有新版 SDK 发布时，我们会及时更新代码，所以你应该定期登录这些网站，看看是否有更新内容。

① 本书源代码也可从图灵社区本书页面“随书下载”部分下载，网址是 <http://www.it-ebooks.com.cn/book/1390>。

1.2.1 开发者选项

免费下载的 Xcode 中包含一个模拟器, 使用这个模拟器, 就可以在 Mac 上创建并运行 iPhone 和 iPad 应用。这个模拟器对于学习编写 iOS 程序极其有用。但是, 模拟器不支持那些依赖于硬件的特性, 比如加速计和摄像头。另外, 免费下载的 SDK 不支持把应用程序安装到 iPhone 等 iOS 真机设备中, 也不能在苹果的 App Store 上出售应用。如果想要这些功能, 需要注册一个付费选项。

- 标准版计划的价格为 99 美元/年。它提供了大量的开发工具和资源, 以及技术支持, 还可以通过苹果的 App Store 出售应用。最重要的是, 它允许在 iOS 设备上 (而不只是在模拟器上) 测试和调试代码。
- 企业版计划的价格为 299 美元/年。如果需要开发公司内部使用的私有 iOS 应用程序, 可以选择这个计划。

可以访问 <http://developer.apple.com/programs/ios> (标准版计划) 和 <http://developer.apple.com/programs/ios/enterprise> (企业版计划) 来查看这两个计划的详细信息。

对于 iOS 所支持的始终联网的移动设备 (比如 iPhone), 由于这种设备使用的是其他公司的无线基础设施, 因此苹果对 iOS 开发人员的限制比对 Mac 开发人员严格得多 (目前, Mac 开发人员无需经过苹果的审查或批准就可以编写并销售程序)。虽然 iPod touch 和那种只支持 Wi-Fi 的 iPad 不使用其他公司的基础设施, 但是它们也受到同样的限制。

苹果添加这些限制, 主要是为了尽量避免发布恶意程序和不良程序, 因为这类程序可能会在共享网络中降低性能。开发 iOS 应用似乎有很高的门槛, 但是苹果也为简化开发过程做出了巨大努力。值得一提的是, 99 美元比 Visual Studio (微软公司的软件开发 IDE) 的价格低得多。

显然, 你还需要一部 iPhone、iPod touch 或 iPad。虽然很多代码都可以通过 iOS 模拟器进行测试, 但并非所有程序都是如此。即便是那些可以在模拟器上运行的应用程序, 也需要在真实设备上进行全面测试后才能对外发布。

注意 如果你准备注册标准版计划或企业版计划, 最好马上注册。因为审批过程可能需要一些时间, 通过审批之后才能在真实设备上运行应用程序。不过不必担心, 前几章中的所有项目以及本书的大多数应用程序都可以在 iOS 模拟器上运行。

1.2.2 必备知识

学习本书应该具备一定的编程知识。我们假设你已经理解基本编程尤其是面向对象编程的基础知识 (例如, 知道类、对象、循环、变量这些基本概念)。你还应该熟悉 Objective-C 编程语言。本书大部分章节都需要用到 Cocoa Touch (它是 SDK 的一部分), Cocoa Touch 使用的是最新版的 Objective-C, 这个版本有一些新特性。不过不了解 Objective-C 的新特性也没有关系。本书在用到 Objective-C 的新特性时会特别指出, 并解释它们的工作原理和使用原因。

作为用户, 你还应该熟悉 iOS 系统本身。就像在其他平台中编写应用程序一样, 你需要熟悉

iPhone、iPad 或 iPod touch 的一些细微差别和怪异模式。花点时间去熟悉 iOS 界面以及 iPhone/iPad 应用的风格吧。

Objective-C 的学习资源

如果你从未使用 Objective-C 编写程序，下面这些资源可以作为入门读物。

- 《Objective-C 基础教程(第2版)》一书浅显易懂，是非常优秀的 Objective-C 入门读物，作者是 Mac 编程专家 Scott Knaster、Waqar Malik 和 Mark Dalrymple。
- *Programming with Objective-C*，这本是苹果出的 Objective-C 语言入门书。你可以在 <https://developer.apple.com/library/mac/documentation/cocoa/conceptual/ProgrammingWithObjectiveC> 网站上找到更多的内容。

1.3 编写 iOS 应用程序有何不同

如果从未使用 Cocoa(或者它的前辈 NeXTSTEP 和 OpenStep)写过程序，你可能会发现 Cocoa Touch(用于编写 iOS 应用程序的应用程序框架)比较新奇。它与其他常用应用程序框架(如用于构建.NET 或 Java 应用程序的框架)之间存在一些根本差异。起初你可能会觉得有点不得要领，不过不必担心，只要多加练习，很快就能对 Cocoa Touch 运用自如了。

如果曾经使用 Cocoa 或 NeXTSTEP 写过程序，你会发现 iOS SDK 中有许多熟悉的身影。有很多类都是从 Mac OS X 版本的 Cocoa 中原样移植过来的。即便是那些不同的类，也遵循相同的基本原则和相似的设计模式。但是，Cocoa 和 Cocoa Touch 之间还是有一些不同的。

无论你的知识背景如何，都需要时刻牢记 iOS 开发与桌面应用程序开发之间的重要差异。接下来几小节讨论这些差异。

1.3.1 只能同时运行一个应用

在 iOS 中，任意时刻只能有一个应用处于活动状态并显示在屏幕上。从 iOS 4 开始，用户按下 Home 键后，应用程序可以在后台继续运行，但这也只限于少数情况，而且必须专门为此编写代码。

当应用程序不处于活动状态也不是在后台运行时，它不会占用任何 CPU 资源，因此也会断开网络连接。iOS 允许在后台进行处理，但要做到这一点，开发者需要多做些努力。

1.3.2 只有一个窗口

在台式机及笔记本的操作系统中，多个程序可以同时运行，每个程序可以创建并控制多个窗口。而 iOS 只允许应用程序操作一个窗口。应用程序与用户的所有交互都在这个窗口中完成，而且这个窗口的大小就是 iOS 设备屏幕的大小，是固定的。