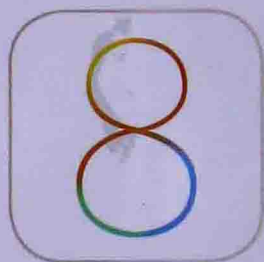




最新 iOS8 开发详解，Xcode 6.X 正式版，涵盖 iOS8 新特性、新技术
Objective-C 和 Swift 双语讲解，完美展示不同语言进行 iOS8 开发的技巧
一百多个实例 + 综合案例开发，涵盖 WatchKit、HealthKit、HomeKit 热门框架
实例全部视频讲解，新增 90 个 iOS 开发常见问题解答，帮助读者尽快进入实战角色

iOS



开发指南

(第2版)

管蕾 编著



源程序+视频



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

ios 8

开发指南

(第2版)

管蕾 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

iOS 8开发指南 / 管蕾编著. -- 2版. -- 北京: 人民邮电出版社, 2015.7
ISBN 978-7-115-39331-9

I. ①i… II. ①管… 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(2015)第111574号

内 容 提 要

本书共 36 章, 循序渐进地讲解了 iOS 8 应用开发的基本知识。本书从搭建开发环境讲起, 依次讲解了 Objective-C 语言基础、Swift 语言基础、Cocoa Touch、Xcode Interface Builder 界面开发、使用 Xcode 编写 MVC 程序、文本框和文本视图、按钮和标签、滑块、步进和图像、使用开关控件和分段控件、Web 视图控件和可滚动视图控件、提醒和操作表、工具栏、日期选择器、表视图、活动指示器、进度条和检索条、UIView 详解、视图控制器、实现多场景和弹出框、iPad 弹出框和分割视图控制器、界面旋转、图形、图像、图层和动画、声音服务、多媒体应用、定位处理、HomeKit 智能家居应用开发、读写应用程序数据、触摸和手势识别、和硬件之间的操作、开发通用的项目程序、推服务和多线程、Touch ID 详解、游戏开发、HealthKit 健康应用开发、WatchKit 智能手表开发等高级知识。本书内容全面, 几乎涵盖了 iOS 8 应用开发所需要的主要内容, 全书内容言简意赅, 讲解方法通俗易懂, 特别适合于初学者学习。

本书适合 iOS 初学者、iOS 程序员、iPhone 开发人员、iPad 开发人员学习, 也可以作为相关培训学校和大专院校相关专业的教学用书。

-
- ◆ 编 著 管 蕾
责任编辑 张 涛
责任印制 张佳莹 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>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 53.75
字数: 1631 千字 2015 年 7 月第 2 版
印数: 7 501 - 10 500 册 2015 年 7 月河北第 1 次印刷

定价: 108.00 元 (附光盘)

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

前 言

2014年6月3日，苹果公司在 WWDC 2014 开发者大会上正式发布了全新的 iOS 8 操作系统。iOS 8 系统采用了一套全新的配色方案，整个界面呈现半透明果冻色，拨号、天气、日历、短信等几乎所有应用的交互界面都进行了重新设计，整体看来更为动感、时尚。为了帮助读者迅速掌握 iOS 8 应用开发的核心技术，笔者特意编写了本书。

第二版升级说明

本书第一版自从 2014 年 9 月上市以来，得到了广大 iOS 8 学习者的认可。由于 Xcode 6 的正式发布，以及 Swift 正式版的出现，为了及时适应技术的变化，笔者特意对本书进行了升级，本版和第一版相比，主要有如下突出变化。

(1) 解决了第一版中 Xcode 6 Beta+Swift 1.0 程序无法在 Xcode 正式版运行的问题。

(2) 添加了 HomeKit、HealthKit 和 WatchKit 这 3 个苹果公司力推的应用框架，这 3 种应用必将成为未来几年的最火热技术之一。

(3) 在本书附带光盘中配备了书中所有实例的调试运行视频，以方便读者顺利进行开发与学习。

(4) 附录部分提供了 90 个 iOS 开发常见问题疑难解答（见光盘），这些问题是初学者在学习过程中最容易遇到的。

本书特色

本书内容丰富，实例全面。我们的目标是通过一本图书，提供多本图书的价值。在内容的编写上，本书具有以下特色。

(1) Objective-C 和 Swift 实现

本书中的实例不仅使用 Objective-C 语言实现，而且使用了苹果公司推出的 Swift 语言。通过学习本书，读者可以掌握使用 Objective-C 语言和 Swift 语言开发 iOS 程序的方法。

(2) 讲解苹果公司力推的新应用技术

本书内容新颖全面，既讲解了 iOS 常用开发知识，也讲解了发展起来的新技术，这些新技术是苹果公司所力推的。例如，HomeKit、HealthKit、WatchKit 和 Touch ID，这些内容是市面中同类书籍所没有的。

(3) 结构合理

从读者的实际需要出发，科学安排知识结构，内容由浅入深，叙述清楚。全书详细地讲解了和 iOS 开发有关的知识点。

(4) 易学易用

读者既可以按照本书编排的章节顺序进行学习，也可以根据自己的需求对某一章节进行有针对性的学习。书中提供的丰富实例可以帮助读者学以致用。

(5) 实用性强

本书彻底摒弃枯燥的理论和简单的操作，注重实用性和可操作性，通过实例的实现过程，详细讲解了各个知识点的具体应用。

(6) 内容全面

无论是搭建开发环境，还是控件接口，或是网络、多媒体和动画以及游戏应用开发，在本书中你

都能找到解决问题的答案。

本书的内容安排

第一篇 必备技术篇

iOS 开发入门、Objective-C 语言基础、Swift 语言基础、Cocoa Touch 框架、Xcode Interface Builder 界面开发、使用 Xcode 编写 MVC 程序。

第二篇 核心技术篇

文本框和文本视图，按钮和标签，滑块、步进和图像，开关控件和分段控件，Web 视图控件、可滚动视图控件和翻页控件，提醒和操作表，工具栏、日期选择器，表视图 (UITable)，活动指示器、进度条和检索条，UIView 详解，视图控制器。

第三篇 技术进阶篇

实现多场景和弹出框，iPad 弹出框和分割视图控制器，界面旋转、大小和全屏处理，图形、图像、图层和动画，声音服务，多媒体应用，定位处理，HomeKit 智能家居应用开发，读写应用程序数据。

第四篇 技术提高篇

触摸和手势识别、和硬件之间的操作、地址簿、邮件和 Twitter、开发通用的项目程序、推服务和多线程、指纹识别技术——Touch ID 详解、游戏开发、HealthKit 健康应用开发、WatchKit 智能手表开发。

第五篇 综合实战篇

公交线路查询系统

附录中提供了 90 个 iOS 开发常见疑难解答 (见光盘)，光盘中配有所有书中实例的讲解视频，帮助读者尽快进入实战角色。

读者对象

初学 iOS 编程的自学者；
大中专院校的老师和学生；
毕业设计的学生；
iOS 编程爱好者；
相关培训机构的老师和学员；
从事 iOS 开发的程序员。

售后服务

为了更好地为读者服务，为大家提供一个完善的学习和交流平台，本书提供了读者交流与答疑 QQ 群 (28316661)，大家可以在里面学习交流。另外还提供了问题答疑和本书源程序的下载地址 www.toppr.net。欢迎广大读者提出宝贵的建议。

本书在编写过程中，得到了人民邮电出版社工作人员的大力支持，正是基于各位编辑的求实、耐心和效率，才使得本书在这么短的时间内出版。另外，也十分感谢我的家人，在我写作的时候给予的大力支持。由于作者水平有限，纰漏和不尽如人意之处在所难免，诚请读者提出意见或建议，以便修订并使之更臻完善，编辑联系邮箱为 zhangtao@ptpress.com.cn。

编者

目 录

第一篇 必备技术篇

第 1 章 iOS 开发入门	2
1.1 iOS 系统介绍	2
1.1.1 iOS 发展史	2
1.1.2 全新的版本——iOS 8	2
1.2 iOS 开发之旅	4
1.3 工欲善其事，必先利其器——搭建开发环境	5
1.3.1 Xcode 6 介绍	5
1.3.2 iOS SDK 介绍	6
1.3.3 下载并安装 Xcode 6	7
1.4 使用 Xcode 开发环境	9
1.4.1 使用 Xcode 创建程序的基本步骤	9
1.4.2 Xcode 的基本功能	10
1.5 启动 iOS 8 模拟器	16
1.6 iOS 的常用开发框架	18
1.6.1 Foundation 框架简介	18
1.6.2 Cocoa 框架简介	19
1.7 常用的第三方工具	20
1.7.1 模拟器 iPhone Simulator	20
1.7.2 Interface Builder	21
第 2 章 Objective-C 语言基础	22
2.1 最耀眼的新星	22
2.1.1 看一份统计数据	22
2.1.2 究竟何为 Objective-C	22
2.1.3 为什么选择 Objective-C	23
2.2 Objective-C 的优点及缺点	23
2.3 一个简单的例子	24
2.3.1 使用 Xcode 编辑代码	24
2.3.2 基本元素介绍	26
2.4 数据类型和常量	30
2.4.1 int 类型	30
2.4.2 float 类型	31
2.4.3 double 类型	32
2.4.4 char 类型	32
2.4.5 字符常量	33
2.4.6 id 类型	34

2.4.7 限定词	35
2.4.8 总结基本数据类型	37
2.5 字符串	37
2.6 算数表达式	38
2.6.1 运算符的优先级	38
2.6.2 整数运算和一元负号运算符	39
2.6.3 模运算符	40
2.6.4 整型值和浮点值的相互转换	41
2.6.5 类型转换运算符	42
2.7 表达式	42
2.7.1 常量表达式	42
2.7.2 条件运算符	43
2.7.3 sizeof 运算符	43
2.7.4 关系运算符	44
2.7.5 强制类型转换运算符	45
2.8 位运算符	45
2.8.1 按位与运算符	45
2.8.2 按位或运算符	46
2.8.3 按位异或运算符	46
2.8.4 一次求反运算符	47
2.8.5 向左移位运算符	48
2.8.6 向右移位运算符	48
2.8.7 总结 Objective-C 的运算符	48
第 3 章 Swift 语言基础	50
3.1 Swift 概述	50
3.1.1 Swift 的创造者	50
3.1.2 Swift 的优势	50
3.2 数据类型和常量	52
3.2.1 int 类型	52
3.2.2 float 类型	52
3.2.3 double 类型	53
3.2.4 char 类型	53
3.2.5 字符常量	53
3.3 变量和常量	53
3.3.1 常量详解	53
3.3.2 变量详解	54
3.4 字符串和字符	55
3.4.1 字符串字面量	55
3.4.2 初始化空字符串	55
3.4.3 字符串可变性	56
3.4.4 值类型字符串	56
3.4.5 计算字符数量	56
3.4.6 连接字符串和字符	57
3.4.7 字符串插值	57
3.4.8 比较字符串	58
3.4.9 Unicode	58
3.5 流程控制	60

3.5.1 for 循环 (1)	61
3.5.2 for 循环 (2)	62
3.5.3 while 循环	62
3.6 条件语句	63
3.6.1 if 语句	63
3.6.2 switch 语句	64
3.7 函数	65
3.7.1 函数的声明与调用	65
3.7.2 函数的参数和返回值	66
3.7.3 函数参数名	68
3.8 实战演练——使用 Xcode 创建 Swift 程序	70
第 4 章 Cocoa Touch 框架	72
4.1 Cocoa Touch 基础	72
4.1.1 Cocoa Touch 概述	72
4.1.2 Cocoa Touch 中的框架	73
4.1.3 Cocoa Touch 的优势	73
4.2 iPhone 的技术层	73
4.2.1 Cocoa Touch 层	73
4.2.2 多媒体层	77
4.2.3 核心服务层	78
4.2.4 核心 OS 层	79
4.3 Cocoa Touch 中的框架	79
4.3.1 Core Animation (图形处理) 框架	79
4.3.2 Core Audio (音频处理) 框架	80
4.3.3 Core Data (数据处理) 框架	80
4.4 iOS 程序的生命周期	81
4.4.1 从一段代码看 iOS 程序的生命周期	81
4.4.2 iOS 程序生命周期的原理	82
4.4.3 UIViewController 的生命周期	84
4.5 Cocoa 中的类	86
4.5.1 核心类	86
4.5.2 数据类型类	88
4.5.3 UI 界面类	91
4.6 国际化	94
4.7 使用 Xcode 学习 iOS 框架	95
4.7.1 使用 Xcode 文档	95
4.7.2 快速帮助	96
第 5 章 Xcode Interface Builder 界面开发	98
5.1 Interface Builder 基础	98
5.1.1 Interface Builder 的作用	98
5.1.2 Interface Builder 的新特色	98
5.2 Interface Builder 采用的方法	100
5.3 Interface Builder 的故事板	100
5.3.1 推出的背景	100
5.3.2 故事板的文档大纲	101
5.3.3 文档大纲的区域对象	102

5.4 创建一个界面	103
5.4.1 对象库	103
5.4.2 将对象加入视图中	104
5.4.3 使用 IB 布局工具	105
5.5 定制界面外观	107
5.5.1 使用属性检查器	107
5.5.2 设置辅助功能属性	108
5.5.3 测试界面	108
5.6 实战演练——将 Xcode 界面连接到代码	109
5.6.1 打开项目	109
5.6.2 输出口和操作	110
5.6.3 创建到输出口的连接	110
5.6.4 创建到操作的连接	112
第 6 章 使用 Xcode 编写 MVC 程序	114
6.1 MVC 模式基础	114
6.1.1 诞生背景	114
6.1.2 分析结构	114
6.1.3 MVC 的特点	115
6.1.4 使用 MVC 实现程序设计的结构化	115
6.2 Xcode 中的 MVC	116
6.2.1 原理	116
6.2.2 模板就是给予 MVC 的	116
6.3 在 Xcode 中实现 MVC	117
6.3.1 视图	117
6.3.2 视图控制器	117
6.4 数据模型	119
6.5 实战演练——使用模板 Single View Application	120
6.5.1 创建项目	120
6.5.2 规划变量和连接	123
6.5.3 设计界面	124
6.5.4 创建并连接输出口和操作	126
6.5.5 实现应用程序逻辑	128
6.5.6 生成应用程序	129
第二篇 核心技术篇	
第 7 章 文本框和文本视图	131
7.1 文本框 (UITextField)	131
7.1.1 文本框基础	131
7.1.2 实战演练——在屏幕中显示一个文本输入框	131
7.1.3 实战演练——设置文本输入框的边框线样式	132
7.1.4 实战演练——设置文本输入框的字体和颜色	133
7.1.5 实战演练——在文本输入框中设置一个清空按钮	133
7.1.6 实战演练——为文本输入框设置背景图片	134
7.1.7 实战演练——实现 UITextField 控件 (基于 Swift 实现)	135
7.2 文本视图 (UITextView)	140
7.2.1 文本视图基础	140

7.2.2 实战演练——在屏幕中换行显示文本	141
7.2.3 实战演练——在屏幕中显示可编辑的文本	142
7.2.4 实战演练——设置屏幕中文本的对齐方式	143
7.2.5 实战演练——使用 UITextView 控件（基于 Swift 实现）	145
第 8 章 按钮和标签	147
8.1 标签 (UILabel)	147
8.1.1 标签 (UILabel) 的属性	147
8.1.2 实战演练——使用标签 (UILabel) 显示一段文本	147
8.1.3 实战演练——在屏幕中显示指定字体和指定大小的文本	150
8.1.4 实战演练——设置屏幕中文本的对齐方式	150
8.2 按钮 (UIButton)	151
8.2.1 按钮基础	151
8.2.2 实战演练——按下按钮后触发一个事件	152
8.2.3 实战演练——在屏幕中显示不同样式的按钮	153
8.3 实战演练——联合使用文本框、文本视图和按钮	154
8.3.1 创建项目	154
8.3.2 设计界面	156
8.3.3 创建并连接输出和操作	162
8.3.4 实现按钮模板	163
8.3.5 隐藏键盘	164
8.3.6 实现应用程序逻辑	166
8.3.7 总结执行	167
8.4 实战演练——显示一个指定样式的文本（基于 Swift 实现）	169
第 9 章 滑块、步进和图像	171
9.1 滑块控件 (UISlider)	171
9.1.1 使用 Slider 控件的基本方法	171
9.1.2 实战演练——滑动滑块时显示对应的值	172
9.1.3 实战演练——滑动滑块控制文字的大小	173
9.1.4 实战演练——实现各种各样的滑块	174
9.1.5 实战演练——实现 UISlider 控件效果（基于 Swift 实现）	180
9.2 步进控件 (UIStepper)	182
9.3 图像视图控件 (UIImageView)	183
9.3.1 UIImageView 的常用操作	183
9.3.2 实战演练——在屏幕中显示图像	185
9.3.3 实战演练——在屏幕中绘制一幅图像	186
9.3.4 实战演练——在屏幕中绘图时设置透明度	187
9.3.5 实战演练——使用 UIImageView 控件（基于 Swift 实现）	188
第 10 章 开关控件和分段控件	191
10.1 开关控件 (UISwitch)	191
10.1.1 开关控件基础	191
10.1.2 实战演练——改变 UISwitch 的文本和颜色	191
10.1.3 实战演练——显示具有开关状态的开关	193
10.1.4 实战演练——联合使用 UISlider 与 UISwitch 控件	195
10.1.5 实战演练——控制是否显示密码明文（基于 Swift 实现）	196
10.2 分段控件	198
10.2.1 分段控件的属性和方法	199

10.2.2 实战演练——使用 UISegmentedControl 控件	200
10.2.3 实战演练——选择一个分段卡后可以改变屏幕的背景颜色	202
10.2.4 实战演练——设置分段卡的显示样式	203
10.2.5 实战演练——设置不显示分段卡的选择状态	204
10.2.6 实战演练——改变分段卡的显示颜色	204
10.2.7 实战演练——选择某个选项时在此分段卡中显示一幅图片	205
10.2.8 实战演练——使用 UISegmentedControl 控件（基于 Swift 实现）	206
第 11 章 Web 视图控件、可滚动视图控件和翻页控件	208
11.1 Web 视图（UIWebView）	208
11.1.1 Web 视图基础	208
11.1.2 实战演练——在屏幕中显示指定的网页	209
11.1.3 实战演练——控制屏幕中的网页	210
11.1.4 实战演练——在网页中加载显示 PDF、Word 和 JPEG 图片	211
11.1.5 实战演练——在网页中加载 HTML 代码	213
11.1.6 实战演练——在网页中实现触摸处理	214
11.1.7 实战演练——使用 UIWebView 控件（基于 Swift 实现）	216
11.2 可滚动的视图（UIScrollView）	220
11.2.1 UIScrollView 的基本用法	220
11.2.2 实战演练——使用可滚动视图控件	224
11.2.3 实战演练——使用 UIScrollView 控件（基于 Swift 实现）	227
11.3 实战演练——联合使用开关、分段控件和 Web 视图控件	228
11.3.1 创建项目	228
11.3.2 设计界面	229
11.3.3 创建并连接输出口和操作	232
11.3.4 实现应用程序逻辑	233
11.3.5 生成应用程序	236
11.4 翻页控件（UIPageControl）	236
11.4.1 PageControll 控件基础	236
11.4.2 实战演练——联合使用 UIPageControl 控件和 UIScrollView 控件（基于 Swift 实现）	237
第 12 章 提醒和操作表	240
12.1 提醒视图（UIAlertView）	240
12.1.1 UIAlertView 基础	240
12.1.2 不同的提醒效果	243
12.1.3 实战演练——实现一个自定义提醒对话框	244
12.1.4 实战演练——使用 UIAlertView 控件（基于 Swift 实现）	246
12.2 操作表（UIActionSheet）	249
12.2.1 操作表的基本用法	250
12.2.2 响应操作表	251
12.2.3 使用 UIActionSheet 的流程	252
12.2.4 实战演练——联合使用 UIAlertController 控件和 UIActionsheet 控件（基于 Swift 实现）	254
第 13 章 工具栏、日期选择器	256
13.1 工具栏（UIToolbar）	256
13.1.1 工具栏基础	256
13.1.2 修改 UIToolBar 的背景颜色	257
13.1.3 实战演练——联合使用 UIToolBar 控件和 UIView 控件	258
13.1.4 实战演练——实现一个播放、暂停按钮	259

13.2 选择器视图 (UIPickerView)	261
13.2.1 选择器视图基础	261
13.2.2 实战演练——实现两个 UIPickerView 控件间的数据依赖	264
13.2.3 实战演练——自定义一个选择器	266
13.2.4 实战演练——实现一个数据选择器 (基于 Swift 实现)	274
13.3 日期选择 (UIDatePicker)	279
13.3.1 UIDatePicker 基础	279
13.3.2 实战演练——使用 UIDatePicker	281
13.3.3 实战演练——实现一个日期选择器	283
13.3.4 实战演练——使用 UIDatePicker 控件 (基于 Swift 实现)	289
第 14 章 表视图 (UITableView)	293
14.1 表视图基础	293
14.1.1 表视图的外观	293
14.1.2 表单元格	293
14.1.3 添加表视图	293
14.1.4 UITableView 详解	297
14.2 实战演练	299
14.2.1 实战演练——拆分表视图	299
14.2.2 实战演练——列表显示 18 条数据	301
14.2.3 实战演练——分段显示列表中的数据	302
14.2.4 实战演练——删除单元格	304
14.2.5 实战演练——自定义 UITableViewCell	305
14.3 实战演练——在表视图中使用其他控件 (基于 Swift 实现)	309
14.4 实战演练——在表视图中显示远程信息 (基于 Swift 实现)	311
第 15 章 活动指示器、进度条和检索条	314
15.1 活动指示器 (UIActivityIndicatorView)	314
15.1.1 活动指示器基础	314
15.1.2 实战演练——实现一个播放器的活动指示器	314
15.1.3 实战演练——使用 UIActivityIndicatorView 控件 (基于 Swift 实现)	316
15.2 进度条 (UIProgressView)	318
15.2.1 进度条基础	318
15.2.2 实战演练——实现一个蓝色进度条效果	318
15.2.3 实战演练——在进度条中显示进度百分比	319
15.2.4 实战演练——实现自定义进度条效果 (基于 Swift 实现)	321
15.3 检索条 (UISearchBar)	325
15.3.1 检索条基础	325
15.3.2 实战演练——在屏幕中实现一个检索框效果	326
15.3.3 实战演练——实现一个实时显示检索框效果	327
15.3.4 实战演练——设置检索框的背景颜色	327
15.3.5 实战演练——在检索框中添加一个书签按钮	328
15.3.6 实战演练——在检索框中添加一个范围条	330
15.3.7 实战演练——使用 UISearchBar 控件 (基于 Swift 实现)	331
第 16 章 UIView 详解	333
16.1 UIView 基础	333
16.1.1 UIView 的结构	333
16.1.2 视图架构	335

16.1.3 视图层次和子视图管理	336
16.1.4 视图绘制周期	336
16.2 实战演练——设置 UIView 的位置和尺寸	336
16.3 实战演练——隐藏指定的 UIView 区域	337
16.4 实战演练——改变背景颜色	338
16.5 实战演练——实现背景透明	340
16.6 实战演练——定位屏幕中的图片	341
16.7 实战演练——旋转和缩放视图	344
第 17 章 视图控制器	347
17.1 导航控制器 (UINavigationController) 简介	347
17.1.1 UINavigationController 基础	347
17.1.2 实战演练——实现不同界面之间的跳转处理	348
17.2 使用 UINavigationController	349
17.2.1 导航栏、导航项和栏按钮项	350
17.2.2 UINavigationController 详解	350
17.2.3 在故事板中使用导航控制器	353
17.2.4 使用 UINavigationController 的流程	354
17.2.5 实战演练——实现不同视图的切换	355
17.2.6 实战演练——通过列表实现不同界面之间的跳转	357
17.2.7 实战演练——使用导航控制器展现 3 个场景	359
17.3 选项卡栏控制器	363
17.3.1 选项卡栏和选项卡栏项	364
17.3.2 在选项卡栏控制器管理的场景之间共享数据	366
17.3.3 UITabBarController 使用详解	366
17.3.4 实战演练——实现不同场景的切换	368
17.3.5 实战演练——使用选项卡栏控制器构建 3 个场景	371
17.4 实战演练——使用 UINavigationController 控件 (基于 Swift 实现)	375

第三篇 技术进阶篇

第 18 章 实现多场景和弹出框	380
18.1 多场景故事板	380
18.1.1 多场景故事板基础	380
18.1.2 创建多场景项目	381
18.1.3 实战演练——使用第二个视图来编辑第一个视图中的信息	385
18.1.4 实战演练——实现多个视图之间的切换	389
18.2 实战演练——多场景视图数据传输 (基于 Swift 实现)	393
第 19 章 iPad 弹出框和分割视图控制器	398
19.1 iPad 弹出框	398
19.1.1 创建弹出框	398
19.1.2 创建弹出切换	398
19.1.3 手工显示弹出框	400
19.1.4 响应用户关闭弹出框	400
19.1.5 以编程方式创建并显示弹出框	401
19.1.6 实战演练——使用弹出框更新内容	403
19.2 探索分割视图控制器	405
19.2.1 分割视图控制器基础	406

19.2.2 表视图实战演练	408
第 20 章 界面旋转、大小和全屏处理	413
20.1 启用界面旋转	413
20.2 设计可旋转和调整大小的界面	414
20.2.1 自动旋转和自动调整大小	414
20.2.2 调整框架	414
20.2.3 切换视图	414
20.2.4 实战演练——使用 Interface Builder 创建可旋转和调整大小的界面	414
20.2.5 实战演练——在旋转时调整控件	417
20.2.6 实战演练——旋转时切换视图	421
第 21 章 图形、图像、图层和动画	424
21.1 图形处理	424
21.1.1 iOS 的绘图机制	424
21.1.2 实战演练——在屏幕中绘制一个三角形	425
21.2 图像处理	427
21.2.1 实战演练——实现颜色选择器/调色板功能	428
21.2.2 实战演练——实现滑动颜色选择器/调色板功能	429
21.3 图层	431
21.3.1 视图和图层	432
21.3.2 实战演练——在屏幕中实现 3 个重叠的矩形	432
21.3.3 实战演练——实现图层的变换	433
21.3.4 实战演练——在指南针上实现某种“页面翻转”的效果	436
21.4 实现动画	439
21.4.1 UIImageView 动画	439
21.4.2 视图动画 UIView	439
21.4.3 Core Animation 详解	444
21.4.4 实战演练——联合使用图像动画、滑块和步进控件	446
21.4.5 实战演练——实现“烟花烟花满天飞”的效果	458
21.5 实战演练——图形图像的人脸检测处理（基于 Swift 实现）	460
第 22 章 声音服务	464
22.1 访问声音服务	464
22.1.1 声音服务基础	464
22.1.2 实战演练——播放声音文件	465
22.2 提醒和振动	469
22.2.1 播放提醒音	469
22.2.2 实战演练——实用 iOS 的提醒功能	469
第 23 章 多媒体应用	478
23.1 MediaPlayer 框架	478
23.1.1 MediaPlayer 框架中的类	478
23.1.2 使用电影播放器	479
23.1.3 处理播放结束	480
23.1.4 使用多媒体选择器	480
23.1.5 使用音乐播放器	481
23.1.6 实战演练——使用 MediaPlayer 播放视频	482
23.2 AV Foundation 框架	484
23.2.1 准备工作	484

23.2.2 使用 AV 音频播放器	484
23.2.3 使用 AV 录音机	485
23.2.4 使用 AV Foundation 框架播放音频的流程	485
23.3 图像选择器 (UIImagePickerController)	487
23.3.1 使用图像选择器	487
23.3.2 图像选择器控制器委托	487
23.3.3 UIImagePickerController 在 iPhone 和 iPad 上的区别	488
23.3.4 用 UIImagePickerController 调用系统相机	489
23.3.5 防止 UIImagePickerController “坏死”	489
23.4 实战演练——一个多媒体的应用程序	490
23.4.1 实现概述	490
23.4.2 创建项目	490
23.4.3 设计界面	491
23.4.4 创建并连接输出口和操作	492
23.4.5 实现电影播放器	492
23.4.6 实现音频录制和播放	494
23.4.7 使用照片库和相机	497
23.4.8 实现 Core Image 滤镜	498
23.4.9 访问并播放音乐库	499
23.5 实战演练——实现 UIImagePickerController 功能 (基于 Swift 语言)	502
23.6 实战演练——实现一个音乐播放器 (基于 Swift)	513
第 24 章 定位处理	520
24.1 Core Location 框架	520
24.1.1 Core Location 基础	520
24.1.2 使用流程	520
24.2 获取位置	522
24.2.1 位置管理器委托	523
24.2.2 处理定位错误	524
24.2.3 位置精度和更新过滤器	524
24.2.4 获取航向	525
24.3 地图功能	526
24.3.1 Map Kit 基础	526
24.3.2 为地图添加标注	527
24.4 实战演练——创建一个支持定位的应用程序	528
24.4.1 创建项目	528
24.4.2 设计视图	529
24.4.3 创建并连接输出口	530
24.4.4 实现应用程序逻辑	531
24.4.5 生成应用程序	532
24.5 实战演练——在屏幕中实现一个定位系统	533
24.5.1 设计界面	533
24.5.2 具体编码	534
24.6 实战演练——使用谷歌地图	536
24.6.1 添加打开地图功能	536
24.6.2 升级视图控制器	538
24.6.3 添加自定义地图标注对象	540

第 25 章 HomeKit 智能家居应用开发	542
25.1 HomeKit 基础	542
25.1.1 苹果 HomeKit 如何牵动全国智能硬件格局	542
25.1.2 给开发者和厂家提供的巨大机会	543
25.1.3 苹果正式推出 HomeKit 硬件标准	543
25.2 HomeKit 开发基础	544
25.2.1 HomeKit 应用程序的层次模型	544
25.2.2 HomeKit 程序架构模式	545
25.2.3 HomeKit 中的类	546
25.3 实战演练——实现一个 HomeKit 控制程序	547
25.4 实战演练——实现一个 HomeKit 控制程序（基于 Swift 语言）	563
第 26 章 读写应用程序数据	578
26.1 iOS 应用程序和数据存储	578
26.2 用户默认设置	579
26.3 设置束	579
26.3.1 设置束基础	579
26.3.2 实战演练——通过隐式首选项实现一个手电筒程序	581
26.4 直接访问文件系统	584
26.4.1 应用程序数据的存储位置	584
26.4.2 获取文件路径	585
26.4.3 读写数据	585
26.4.4 读取和写入文件	586
26.4.5 通过 plist 文件存取文件	587
26.4.6 保存和读取文件	589
26.4.7 文件共享和文件类型	589
26.4.8 传递一个文档	590
26.4.9 实战演练——实现一个收集用户信息的程序	590
26.5 iCloud 存储	594
26.6 使用 SQLite3 存储和读取数据	595
26.7 核心数据	599
26.7.1 Core Data 基础	599
26.7.2 Core Data 基本架构	600
26.7.3 托管对象（Managed Objects）和托管对象模型（Managed Object Model）	603
26.7.4 实战演练——在 iOS 中使用 CoreData	604
26.8 互联网数据	606
26.8.1 XML 和 JSON	606
26.8.2 实战演练——使用 JSON 获取网站中的照片信息	609

第四篇 技术提高篇

第 27 章 触摸和手势识别	614
27.1 多点触摸和手势识别基础	614
27.2 触摸处理	614
27.2.1 触摸事件和视图	615
27.2.2 实战演练——触摸屏幕中的按钮	620
27.2.3 实战演练——同时滑动屏幕中的两个滑块	621
27.2.4 实战演练——触摸屏幕检测	622

27.2.5 实战演练——触摸屏幕中的文字标签	622
27.3 手势处理	623
27.3.1 手势处理基础	623
27.3.2 实战演练——实现一个手势识别器	629
第 28 章 和硬件之间的操作	636
28.1 加速计和陀螺仪	636
28.1.1 加速计基础	636
28.1.2 陀螺仪	640
28.1.3 实战演练——检测倾斜和旋转	641
28.2 访问朝向和运动数据	645
28.2.1 两种方法	645
28.2.2 实战演练——检测朝向	647
28.3 实战演练——使用 Motion 传感器（基于 Swift 语言）	649
第 29 章 地址簿、邮件和 Twitter	651
29.1 地址簿	651
29.1.1 框架 Address Book UI	651
29.1.2 框架 Address Book	652
29.2 电子邮件	653
29.3 使用 Twitter 发送推特信息	654
29.4 实战演练——联合使用地址簿、电子邮件、Twitter 和地图	654
29.4.1 创建项目	655
29.4.2 设计界面	655
29.4.3 创建并连接输出 和 操作	656
29.4.4 实现地址簿逻辑	656
29.4.5 实现地图逻辑	658
29.4.6 实现电子邮件逻辑	660
29.4.7 实现 Twitter 逻辑	661
29.4.8 生成应用程序	662
29.5 实战演练——开发一个 Twitter 客户端（基于 Swift 语言）	662
第 30 章 开发通用的项目程序	666
30.1 开发通用应用程序	666
30.1.1 在 iOS 6 中开发通用应用程序	666
30.1.2 在 iOS 7、iOS 8 中开发通用应用程序	667
30.1.3 图标文件	674
30.1.4 启动图像	675
30.2 实战演练——使用通用程序模板创建通用应用程序	675
30.2.1 创建项目	675
30.2.2 设计界面	675
30.2.3 创建并连接输出	676
30.2.4 实现应用程序逻辑	676
30.3 实战演练——使用视图控制器	677
30.3.1 创建项目	677
30.3.2 设计界面	678
30.3.3 创建并连接输出	678
30.3.4 实现应用程序逻辑	678
30.3.5 生成应用程序	679