

iOS 实战 (Swift版)

苹果“生态圈”编程卷

关东升 著



清华大学出版社





关东升 著

iOS 实战 苹果“生态圈”编程卷

(Swift版)



清华大学出版社
北京

内 容 简 介

本书是智捷课堂团队编著的“iOS 实战”系列图书中的一本,本书系统介绍了苹果产品“生态圈”的相关知识。全书共分为 13 章,分别介绍了苹果网络编程、Core Data 技术、iCloud 编程、推送通知、社交网络编程、Newsstand、应用内购买、Passbook、HomeKit、Handoff、定位服务和地图应用开发。书中全部采用基于 iOS 8 的最新 API。

本书适合作为从事 iOS 应用开发的软件工程师的参考用书,也可以作为计算机科学与技术、软件工程等专业移动开发类课程的教学用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

iOS 实战: 苹果“生态圈”编程卷: Swift 版/关东升著. --北京: 清华大学出版社, 2016
(清华开发者书库)

ISBN 978-7-302-41774-3

I. ①i… II. ①关… 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 243381 号

责任编辑: 盛东亮

封面设计: 李召霞

责任校对: 胡伟民

责任印制: 杨 艳

出版发行: 清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址: 北京清华大学学研大厦 A 座

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者: 三河市少明印务有限公司

经 销: 全国新华书店

开 本: 186mm×240mm 印 张: 20.5

字 数: 512 千字

版 次: 2016 年 1 月第 1 版

印 次: 2016 年 1 月第 1 次印刷

印 数: 1~2500

定 价: 49.00 元

产品编号: 064658-01

目录

CONTENTS

第 1 章 准备开始	1
1.1 本书结构	1
1.2 本书中的约定	2
1.2.1 实例代码约定	2
1.2.2 图示约定	3
1.2.3 图中手势	5
1.2.4 方法命名约定	5
1.2.5 构造器命名约定	7
第 2 章 苹果网络	8
2.1 网络结构	8
2.1.1 客户端服务器结构网络	8
2.1.2 对等结构网络	8
2.2 基于 Socket 的 C/S 结构网络通信	9
2.2.1 Socket 概念	9
2.2.2 Socket 编程	10
2.2.3 实例: CFStream 实现 TCP Socket 服务器端	11
2.2.4 实例: NSStream 实现 TCP Socket 客户端	17
2.3 Bonjour 服务发现	23
2.3.1 发布服务	23
2.3.2 解析服务	24
2.3.3 查找服务	25
2.3.4 实例: 基于服务发现的 Socket 通信服务器端	26
2.3.5 实例: 基于服务发现的 Socket 通信客户端	31
2.4 对等结构网络	34
2.4.1 使用 Multipeer Connectivity 开发对等结构网络应用	34
2.4.2 实例: P2PGame 游戏	37

本章小结	44
第3章 Core Data 数据持久化技术	45
3.1 对象关系映射	45
3.2 Core Data 堆栈	46
3.3 建模和生成实体	50
3.3.1 建模	50
3.3.2 生成实体	53
3.4 实例：采用 Core Data 实现 MyNotes 应用	55
3.4.1 采用 Core Data 分层架构设计	56
3.4.2 查询数据	59
3.4.3 修改数据	62
本章小结	64
第4章 iCloud 编程	65
4.1 苹果云服务——iCloud	65
4.2 键值数据存储	69
4.2.1 键值数据存储编程	70
4.2.2 实例：iCloud 键值数据存储	70
4.3 文档存储	75
4.3.1 iCloud 存储运行过程	75
4.3.2 实例：iCloud 文档存储编程	76
4.4 iCloud Core Data 技术	85
4.4.1 实例：iCloud Core Data 实现 MyNotes 应用	86
4.4.2 配置 Xcode 工程	86
4.4.3 修改持久化存储协调者	86
4.4.4 修改被管理的对象上下文	88
4.4.5 合并 iCloud 数据	89
4.4.6 更新 UI	90
4.5 CloudKit 存储编程	91
4.5.1 CloudKit 术语	91
4.5.2 iCloud Dashboard	91
4.5.3 实例：CloudKit 实现 MyNotes 应用	92
本章小结	102

第 5 章 推送通知	103
5.1 苹果中究竟有多少个“通知”	103
5.1.1 广播通知	103
5.1.2 本地通知	103
5.1.3 推送通知	108
5.2 配置推送通知	109
5.2.1 证书签名请求	110
5.2.2 创建 App ID	110
5.2.3 配置 App ID 支持推送通知	113
5.2.4 生成 APNS 的 SSL 证书	113
5.2.5 下载 SSL 证书	116
5.2.6 为 SSL 证书导出 p12 文件	116
5.3 iOS 客户端编程	120
5.3.1 代码实现	120
5.3.2 可选择的推送通知	121
5.4 推送服务端编程	124
5.4.1 使用 PHP 实现推送服务	124
5.4.2 使用 Java 推送服务	127
本章小结	128
第 6 章 社交网络编程	129
6.1 苹果的社交网络	129
6.1.1 设置账户信息	129
6.1.2 社交网络编程 API	130
6.2 使用 iOS 标准界面编写客户端	131
6.2.1 撰写视图控制器 SLComposeViewController	131
6.2.2 实例：新浪微博客户端	131
6.3 编写自己的社交网络客户端	133
6.3.1 编写社交网络客户端应用程序流程	133
6.3.2 实例：编写新浪微博客户端	137
本章小结	146
第 7 章 报刊杂志——Newsstand 应用编程	147
7.1 报刊杂志应用	147
7.2 报刊杂志应用编程	147

7.2.1	配置报刊杂志应用	149
7.2.2	使用 Newsstand Kit API	152
7.3	实例：报刊杂志应用	153
7.3.1	实例设计原型	153
7.3.2	实例设计类图	154
7.3.3	初始化杂志界面	154
7.3.4	主动式下载资源	163
7.3.5	后台下载资源	168
7.3.6	断点续传	172
7.3.7	读取杂志内容	173
	本章小结	175
第 8 章	应用内购买	176
8.1	应用内购买实例设计	177
8.2	搭建应用内购买开发环境	177
8.2.1	添加测试用户	178
8.2.2	创建 App ID	179
8.2.3	添加空白应用 IAPDemo	179
8.2.4	为 IAPDemo 应用添加购买项目	182
8.3	内置产品类型应用内购买	186
8.3.1	创建工程	186
8.3.2	初始化处理	186
8.3.3	获得产品信息	188
8.3.4	处理交易结果	191
8.3.5	恢复交易	194
8.3.6	测试应用内购买	194
8.4	服务器产品类型应用内购买	197
8.4.1	工作流程	197
8.4.2	应用内购买 iOS 客户端	199
8.4.3	PHP 服务器端实现	204
	本章小结	206
第 9 章	Passbook 开发	207
9.1	Passbook 与 Pass	208
9.1.1	Pass 的 5 种风格	210
9.1.2	Pass 的内部结构	211

9.2 设计 Pass	212
9.2.1 pass.json 文件结构	212
9.2.2 Pass 类型和字段	213
9.2.3 二维条形码	216
9.2.4 位置与时间	217
9.2.5 本地化	218
9.2.6 图标和图片的使用	219
9.3 创建 Pass	221
9.3.1 创建 Pass 证书	221
9.3.2 数字签名与打包 Pass 文件	225
9.4 发布 Pass	226
9.4.1 Email 发布	226
9.4.2 Web 网站发布	228
9.5 Pass Kit 编程	230
9.5.1 实例: MyPass 实例设计与说明	230
9.5.2 配置 Xcode 工程	231
9.5.3 访问 Passbook	232
9.5.4 添加与替换 Pass	234
9.5.5 查看 Pass 详细内容和删除 Pass	237
本章小结	238
第 10 章 智能家居与 HomeKit 编程	239
10.1 苹果配件开发	239
10.2 HomeKit 智能家居平台	240
10.3 HomeKit 开发框架	241
10.3.1 HomeKit 术语	241
10.3.2 HomeKitAPI	242
10.4 HomeKit 编程	242
10.4.1 实例: Philips Light bulb 控制设计与说明	243
10.4.2 配置 Xcode 工程	243
10.4.3 故事板设计实现	244
10.4.4 创建家庭和房间	245
10.4.5 配件查找和控制	252
10.4.6 实例测试	257
本章小结	262

第 11 章 Handoff 编程	263
11.1 Handoff 技术	263
11.1.1 使用 Handoff 的要求	264
11.1.2 设置 Handoff	264
11.1.3 Handoff 工作原理	264
11.2 Handoff 编程	266
11.2.1 Handoff API	266
11.2.2 Handoff 编程的一般流程	267
11.3 实例：灯泡控制器	268
11.3.1 实例描述	269
11.3.2 故事板设计实现	269
11.3.3 代码实现	270
本章小结	273
第 12 章 定位	274
12.1 获得用户的位置	274
12.1.1 定位服务编程	275
12.1.2 实例：定位服务	276
12.1.3 距离过滤器和精度	277
12.1.4 启动和停止定位服务	278
12.1.5 获得位置信息	279
12.1.6 关于定位服务的测试	281
12.2 地理信息编码与反编码	285
12.2.1 地理信息反编码	285
12.2.2 实例：地理信息反编码	286
12.2.3 地理信息编码查询	287
12.2.4 实例：地理信息编码查询	288
12.3 微定位技术	290
12.3.1 地理围栏	290
12.3.2 iBeacon 技术	290
12.3.3 实例：iBeacon 技术实现微定位	292
本章小结	299
第 13 章 地图应用开发	300
13.1 使用 iOS 苹果地图	300

13.1.1 显示地图	300
13.1.2 添加标注	303
13.1.3 跟踪用户位置变化	307
13.2 使用程序外地图	309
本章小结	314

在开始具体介绍苹果产品生态圈相关知识之前,笔者有必要向广大读者介绍一下本书的知识结构和书中的约定。

1.1 本书结构

本书向广大读者介绍苹果产品生态圈的相关知识,其中包括苹果网络编程、Core Data 技术、iCloud 编程、社交网络编程、推送通知、Newsstand、应用内购买、Passbook、HomeKit、Handoff、定位服务和地图应用开发。书中全部采用基于 iOS 8 最新的 API。

全书包括内容:

- 第 2 章 苹果网络,介绍了 Socket 编程、Bonjour 服务发现和 Multipeer Connectivity 等内容。
- 第 3 章 Core Data 数据持久化技术,介绍了对象关系映射、Core Data 堆栈、建模和生成实体。
- 第 4 章 iCloud 编程,介绍了 iCloud 技术,然后展开介绍了键值数据存储、文档存储和 iCloud Core Data 技术,以及 CloudKit 存储编程。
- 第 5 章 推送通知,介绍了推送通知、推送通知客户端开发和推送服务端编程。
- 第 6 章 社交网络编程,介绍了新浪微博、腾讯微博、Facebook 和 Twitter 等社交网络编程。
- 第 7 章 报刊杂志应用编程,介绍了报刊杂志应用编程过程。
- 第 8 章 应用内购买,介绍了内置产品类型应用内购买和服务器产品类型应用内购买。
- 第 9 章 Passbook 开发,介绍了 Passbook 和 Pass,创建 Pass 以及 Pass Kit 编程。
- 第 10 章 智能家居与 HomeKit 编程,介绍了 HomeKit 技术和具体编程过程。
- 第 11 章 Handoff 编程,介绍了 Handoff 技术和具体编程过程。
- 第 12 章 定位,介绍了定位服务编程、地理信息编码和地理信息反编码。最后还介绍了苹果的微定位技术 iBeacon。

- 第13章 地图应用开发,首先介绍了 iOS 苹果地图的使用,包括显示地图、添加标注以及跟踪用户位置变化等。最后,还介绍了程序外地图的使用。

1.2 本书中的约定

为了方便大家使用本书,本节介绍一下本书中实例代码和图示的相关约定。

1.2.1 实例代码约定

本书作为一本介绍编程知识的图书,书中有很多实例代码,读者下载本书代码并解压代码,可以看到下面的目录结构:

```
.
├── ch2
│   ├── 2.2 Socket
│   │   ├── TCPClient
│   │   └── TCPServer
│   └── 2.3 Bonjour服务发现
│       ├── BonjourClient_Resolve
│       ├── BonjourClient_Search
│       └── BonjourServer
├── 2.4
│   └── P2PGame (标准界面)
├── ch3
│   ├── 3.2 CoreDataStack
│   └── 3.3~3.4 MyNotes
├── ch4
│   ├── 4.2.2 DoucmentDemo
│   ├── 4.3.2 KeyValueDemo
│   ├── 4.4 MyNotes
│   └── 4.5 MyNotes
├── ch5
│   ├── 5.1.3 LocalNotifi
│   ├── 5.3.1 PushNotifiDemo
│   ├── 5.3.2 PushNotifiDemo
│   ├── 5.4.1 phpPNs
│   └── 5.4.2 JavaPNs
├── ch6
│   ├── 6.2.2 SinaWeiBo
│   └── 6.3.2 SinaWeiBo
├── ch7
│   ├── 7.3 NewsstandDemo
│   └── 7.3 PhpPNs
├── ch8
│   ├── 8.3 IAPDemo
│   └── 8.4
│       └── IAPDemo
├── ch9
│   ├── 9.3
│   ├── 9.4.2
│   └── 9.5 PassDemo
├── ch10
│   └── Philips Light bulb
├── ch11
│   └── HandoffDemo
```

```

|___ch12
| |___12.1 MyLocation
| |___12.2.2 ReverseGeocode
| |___12.2.4 GeocodeQuery
| |___12.3.3
| |___APPClient
| |___BeaconDevice
|___ch13
|___13.1.1~13.1.2 GeocodeQuery
|___13.1.3 UserTrackingMode
|___13.2 GeocodeQuery

```

目录 ch2~ch13 代表第 2 章~第 13 章的实例代码或一些资源文件,其中工程或工作空间的命名有如下几种形式:

- 二级目录标号,如“9.5 PassDemo”说明是第 9 章第 9.5 节中使用的 PassDemo 工程实例;
- 三级目录标号,如“13.1.3 UserTrackingMode”说明是第 13 章第 13.1.3 节中使用的 UserTrackingMode 工程实例;
- 没有标号情况下,由所在父目录说明是哪个章节的实例工程。

1.2.2 图示约定

为了更形象、更有效地说明知识点或描述操作,本书添加了很多插图,下面对插图的一些符号含义进行简要说明:

1. 图中的圈框

有时读者会看到图 1-1 所示的圈框,圈框中的内容是选中的内容或重点要说明的内容。

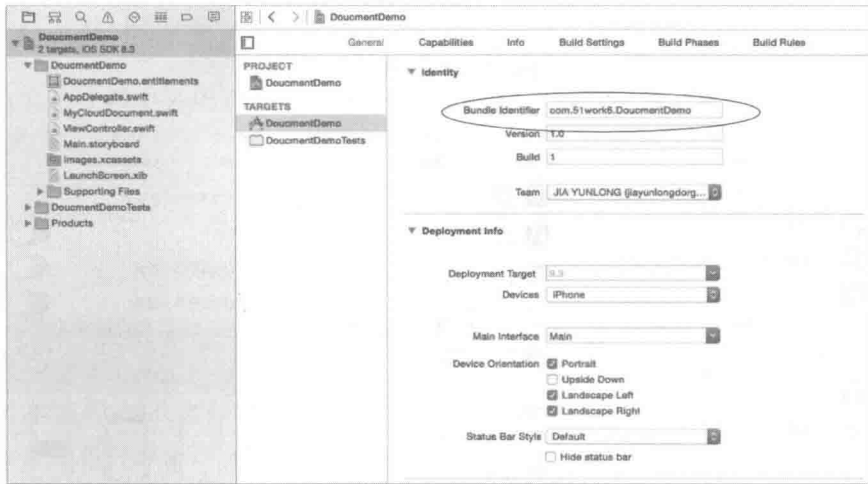


图 1-1 图中圈框

2. 图中的箭头

如图 1-2、图 1-3 和图 1-4 所示,箭头用于说明用户的动作,一般箭尾是动作开始的地

方,箭头指向动作结束的地方。图 1-2 使用实线箭头表示选中控件修改属性。图 1-3 使用虚线箭头表示操作。图 1-4 所示箭头代表界面的跳转。

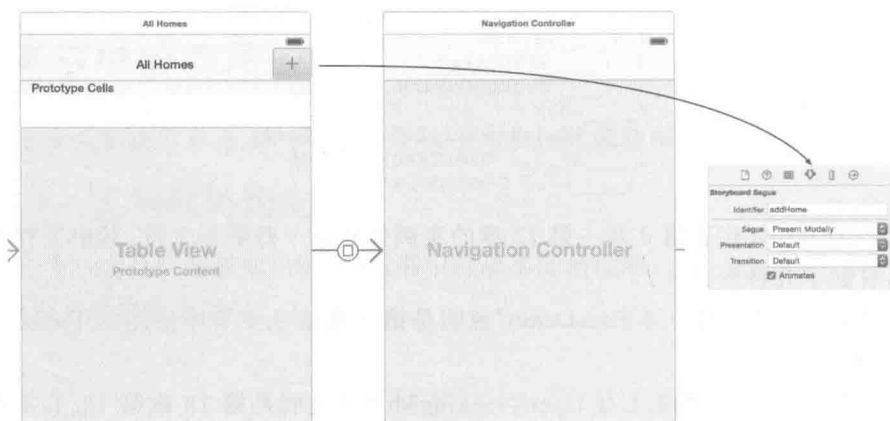


图 1-2 图中箭头 1

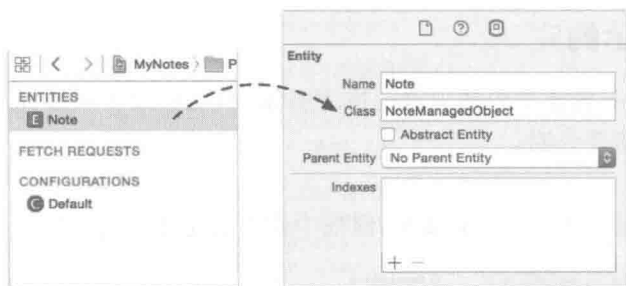


图 1-3 图中箭头 2



图 1-4 图中箭头 3

1.2.3 图中手势

为了描述操作动作,我们在图中放置了 ☞ 等手势符号,这说明点击或操作该处的按钮等控件,如图 1-5 所示。



图 1-5 图中手势

上述内容包含本书使用的大部分符号,一些个别符号我们会在使用的地方具体说明。

1.2.4 方法命名约定

苹果在官方文档中采用 Objective-C 多重参数描述 API,它将方法名按照参数的个数分成几个部分。

提示 关于 Objective-C 多重参数,下面的代码实现了在一个集合中按照索引插入元素。

```
-(void)insertObject:(id)anObject atIndex:(NSInteger)index
```

图 1-6 说明了 Objective-C 多重参数方法定义,第一个参数是 anObject,参数类型是 id 类型;第二个参数是 index,参数类型是 NSInteger,这叫做多重参数。它的返回类型是 void,方法签名是 insertObject:atIndex:。方法类型标识符中用“-”代表方法是实例方法,“+”代表方法是类方法,关于实例方法和类方法我们将在后面内容中讨论。如果上面的方法变成 C 或 C++ 形式,则是下面的样子的:

```
void insertObjectAtIndex(id anObject, NSInteger index)
```

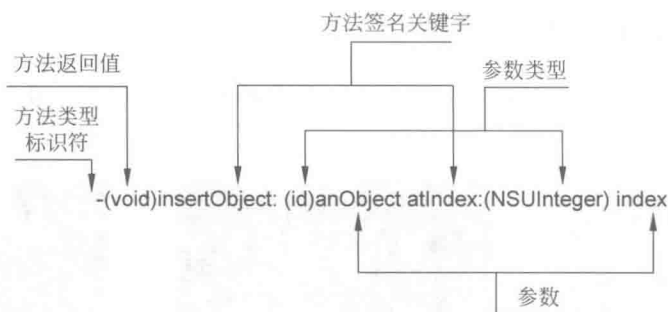



图 1-6 Objective-C 多重参数方法定义

苹果公司在推出 Swift 语言后，仍然采用多重参数描述 API，苹果 API 文档如图 1-7 所示。



图 1-7 苹果 API 文档

图 1-7 所示是 UITableView 类的 `-numberOfRowsInSection:` 方法，其中“-”表示实例方法，“+”表示静态方法，“:”表示有参数。

该方法表示成为 Swift 语言是如下方法：

```
func numberOfRowsInSection(_ section: Int) -> Int
```

该方法表示成为 Objective-C 语言是如下方法：

- (NSInteger)numberOfRowsInSection:(NSInteger)section

为了统一命名我们也采用苹果官方的提法,即在本书中提到 Swift 方法时也采用 -numberOfRowsInSection: 多重参数形式,特殊情况下我们会加以说明。

1.2.5 构造器命名约定

构造器是特殊的方法,它也采用 Objective-C 多重参数描述 API,但是更为特殊,例如: UITableView 的构造器是“-initWithFrame:style:”,如图 1-8 所示。

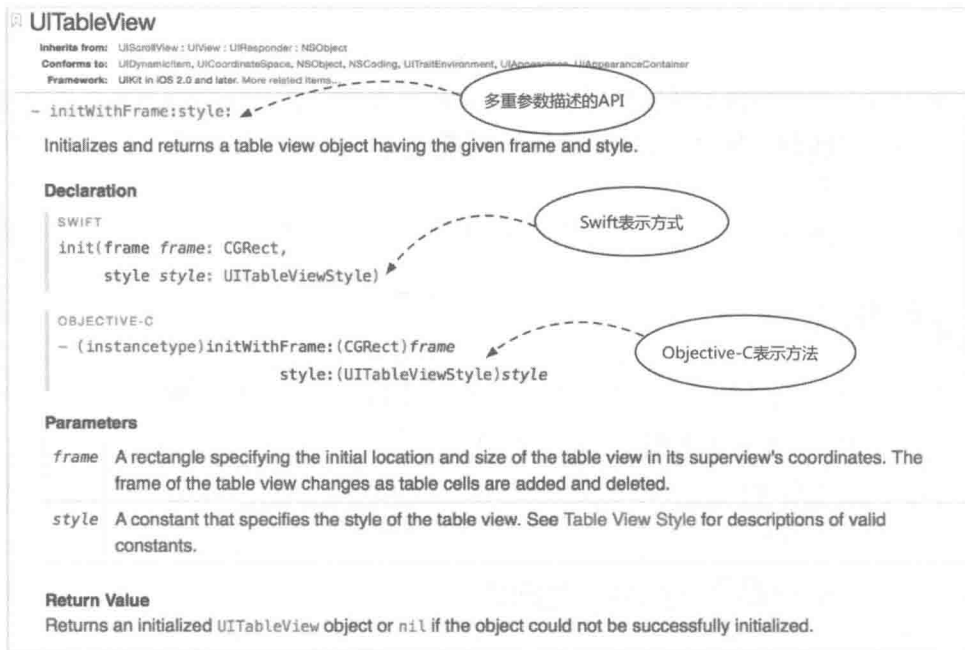


图 1-8 苹果构造器 API

该构造器表示成为 Swift 语言形式如下:

```
init(frame frame: CGRect, style style: UITableViewStyle)
```

该构造器表示成为 Objective-C 语言是如下方法:

```
-(instancetype)initWithFrame:(CGRect)frame style:(UITableViewStyle)style
```

为了统一命名我们也采用苹果官方的提法,即在本书中提到 Swift 构造器时也采用 -initWithFrame:style: 多重参数形式,特殊情况下我们会加以说明。