



# iOS 实战 (Swift版)

## 传感器卷

关东升 著



清华大学出版社





关东升 著

# iOS 实战 传感器卷

(Swift版)



清华大学出版社

## 内 容 简 介

本书系统论述了 iOS 传感器应用开发的相关知识,全部采用最新的基于 iOS 8 的 API。全书分为 9 章:第 1 章介绍开发准备工作、本书结构及书中的一些约定;第 2 章介绍手势识别,通过两种方式实现 iOS 常用手势,包括 Tap(单击)、Long Press(长按)、Pan(拖动)、Swipe(滑动)、Rotation(旋转)和 Pinch(手指的合拢和张开);第 3 章介绍加速度计与陀螺仪,通过这两种可以感知设备运动状态的传感器,可以开发出很多有趣的应用和游戏;第 4 章介绍指南针与磁力计,磁力计可以感知周围的磁场,可以开发指南针、罗盘等导航应用,还可以开发特斯拉计(或高斯计)等感知磁场强度的应用;第 5 章介绍照相机与摄像头,包括图片抓取、视频捕获等技术;第 6 章介绍音频设备访问,重点介绍了使用 AVFoundation 框架实现音频播放和音频录制;第 7 章介绍使用蓝牙,重点介绍了低功耗蓝牙、Game Kit 和蓝牙通信;第 8 章介绍定位技术,包括定位服务编程、地理信息编码、地理信息反编码、微定位技术 iBeacon 和接近传感器;第 9 章是项目实战,通过增强现实版本铅锤应用介绍 iOS 敏捷开发过程,本项目中综合用到了视频捕获实现的增强现实技术、加速度计等传感器。

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:传感器卷(Swift 版)/关东并著. —北京:清华大学出版社,2015

(清华开发者书库)

ISBN 978-7-302-40187-2

I. ①i… II. ①关… III. ①传感器—应用开发 IV. ①TN929.53 ②TP212

中国版本图书馆 CIP 数据核字(2015)第 119777 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:梁毅

责任印制:沈露

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:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:186mm×240mm 印 张:12.75

字 数:322 千字

版 次:2015 年 8 月第 1 版

印 次:2015 年 8 月第 1 次印刷

印 数:1~2500

定 价:49.00 元

产品编号:064628-01

# 前言

## PREFACE

在 iOS 应用开发中,有很多内容涉及硬件访问和传感器编程。本书将向读者系统介绍 iOS 设备的传感器应用开发。由于苹果公司推出了 iOS 开发的新语言——Swift;而我们智捷课堂团队以前编写的一系列 iOS 经典图书,也都需要升级为 Swift 语言版本以满足开发者需求,基于这样的背景,我们智捷课堂与清华大学出版社联合策划了 5 本有关 iOS 开发的图书:

- 《iOS 实战:入门与提高卷(Swift 版)》
- 《iOS 实战:图形图像、动画与多媒体卷(Swift 版)》
- 《iOS 实战:传感器卷(Swift 版)》
- 《iOS 实战:苹果“生态圈”编程卷(Swift 版)》
- 《iOS 实战:Apple Watch 卷(Swift 版)》

有关“iOS 开发实战”系列图书的具体进展,请读者关注智捷课堂官方网站 <http://www.5lwork6.com>。

### 本书网站

为了更好地为广大读者提供服务,我们专门为本书建立了一个网站 <http://www.5lwork6.com/ios4.php>,读者可以查看相关出版进度,并对书中内容发表评论,提出宝贵意见。

### 源代码

书中提供了 100 多个完整的项目案例源代码,全部采用最新的 iOS 8 API 和操作界面,读者可以到本书网站 <http://www.5lwork6.com/ios4.php> 下载。

### 勘误与支持

我们在本书网站 <http://www.5lwork6.com/ios4.php> 中建立了一个勘误专区,及时地把书中的错误、纰漏和修正方案反馈给广大读者。如果读者在学习过程中,发现了什么问题,可以在网上留言,也可以发送电子邮件到 [eorient@sina.com](mailto:eorient@sina.com),我们会在第一时间回复您。读者也可以在新浪微博(@tony\_关东升)或微信(tony 关东升)中与我们联系。

本书主要由关东升执笔撰写。此外,智捷课堂团队的贾云龙、赵大羽、赵志荣、李玉超、关珊和李政刚也参与了本书部分内容的编写工作。感谢清华大学出版社的盛东亮编辑给我们提供了宝贵的意见。感谢赵大羽老师手绘了书中全部草图,并从专业的角度修改书中图

片,力求更加真实完美地呈现给广大读者。感谢我的家人对我的理解和支持,使我能投入全部精力,专心编写此书。

由于时间仓促,书中难免存在不妥之处,请读者谅解并提出宝贵意见。



2015年5月于北京

# 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 .....                       | 1  |
| 1.2.1 实例代码约定 .....                     | 2  |
| 1.2.2 图示约定 .....                       | 3  |
| 1.2.3 方法命名约定 .....                     | 5  |
| 1.2.4 构造器命名约定 .....                    | 6  |
| 第 2 章 手势识别 .....                       | 8  |
| 2.1 手势种类 .....                         | 8  |
| 2.2 使用手势识别器 .....                      | 10 |
| 2.2.1 视图对象与手势识别 .....                  | 10 |
| 2.2.2 手势识别状态 .....                     | 10 |
| 2.2.3 检测 Tap(单击) .....                 | 12 |
| 2.2.4 检测 Long Press(长按) .....          | 22 |
| 2.2.5 检测 Pan(平移) .....                 | 26 |
| 2.2.6 检测 Swipe(滑动) .....               | 29 |
| 2.2.7 检测 Rotation(旋转) .....            | 32 |
| 2.2.8 检测 Pinch(手指的合拢和张开) .....         | 36 |
| 2.2.9 检测 Screen Edge Pan(屏幕边缘平移) ..... | 39 |
| 2.3 触摸事件与手势识别 .....                    | 40 |
| 2.3.1 事件处理机制 .....                     | 40 |
| 2.3.2 响应者对象与响应链 .....                  | 41 |
| 2.3.3 触摸事件 .....                       | 42 |
| 2.3.4 手势识别 .....                       | 48 |
| 本章小结 .....                             | 51 |

|                                     |    |
|-------------------------------------|----|
| <b>第3章 加速度计与陀螺仪</b>                 | 52 |
| 3.1 加速度计                            | 53 |
| 3.1.1 访问加速度计                        | 53 |
| 3.1.2 实例：通过 CMMotionManager 访问加速度计  | 54 |
| 3.1.3 数据的“滤波”                       | 56 |
| 3.1.4 实例：数据的“滤波”                    | 58 |
| 3.1.5 感知设备方向                        | 62 |
| 3.1.6 实例：感知设备方向                     | 62 |
| 3.2 陀螺仪                             | 66 |
| 3.2.1 访问陀螺仪                         | 67 |
| 3.2.2 检测晃动设备                        | 69 |
| 3.2.3 实例：检测晃动设备                     | 70 |
| 本章小结                                | 71 |
| <b>第4章 指南针与磁力计</b>                  | 72 |
| 4.1 访问磁力计                           | 72 |
| 4.1.1 访问导航方面的 API                   | 73 |
| 4.1.2 磁力计与设备方向                      | 73 |
| 4.1.3 实例：指南针应用                      | 75 |
| 4.2 检测磁场                            | 78 |
| 4.2.1 检测磁场 API                      | 79 |
| 4.2.2 实例：特斯拉计应用                     | 79 |
| 本章小结                                | 82 |
| <b>第5章 照相机与摄像头</b>                  | 83 |
| 5.1 iOS 设备摄像头                       | 84 |
| 5.2 图像抓取                            | 84 |
| 5.2.1 图像选择器                         | 84 |
| 5.2.2 实例：照相机抓取图片                    | 85 |
| 5.2.3 编辑图片                          | 89 |
| 5.2.4 保存图片                          | 90 |
| 5.2.5 添加照相机图层                       | 92 |
| 5.3 视频捕获                            | 94 |
| 5.3.1 实例：视频捕获                       | 94 |
| 5.3.2 使用 UIImagePickerController 捕获 | 95 |

|                                    |            |
|------------------------------------|------------|
| 5.3.3 使用 AV Foundation 捕获 .....    | 98         |
| 本章小结 .....                         | 100        |
| <b>第 6 章 音频设备访问</b> .....          | <b>101</b> |
| 6.1 音频 API 简介 .....                | 101        |
| 6.2 AVFoundation 框架实现音频录制与播放 ..... | 102        |
| 6.2.1 音频播放 .....                   | 102        |
| 6.2.2 实例：音频播放 .....                | 103        |
| 6.2.3 音频录制 .....                   | 106        |
| 6.2.4 实例：音频录制 .....                | 107        |
| 本章小结 .....                         | 111        |
| <b>第 7 章 使用蓝牙</b> .....            | <b>112</b> |
| 7.1 低功耗蓝牙 .....                    | 112        |
| 7.1.1 Core Bluetooth API .....     | 112        |
| 7.1.2 中心角色 .....                   | 113        |
| 7.1.3 外设角色 .....                   | 116        |
| 7.2 实例：低功耗蓝牙通信 .....               | 119        |
| 7.2.1 中心角色编程实现 .....               | 121        |
| 7.2.2 外设角色编程实现 .....               | 127        |
| 7.3 使用 SensorTag 开发低功耗蓝牙应用 .....   | 132        |
| 7.3.1 SensorTag 开发套件 .....         | 132        |
| 7.3.2 实例：我的温度计 .....               | 133        |
| 本章小结 .....                         | 143        |
| <b>第 8 章 定位</b> .....              | <b>144</b> |
| 8.1 获得用户的位置 .....                  | 144        |
| 8.1.1 定位服务编程 .....                 | 145        |
| 8.1.2 实例：定位服务 .....                | 146        |
| 8.1.3 距离过滤器和精度 .....               | 147        |
| 8.1.4 启动和停止定位服务 .....              | 148        |
| 8.1.5 获得位置信息 .....                 | 149        |
| 8.1.6 关于定位服务的测试 .....              | 150        |
| 8.2 地理信息反编码与编码查询 .....             | 155        |
| 8.2.1 地理信息反编码 .....                | 155        |
| 8.2.2 实例：地理信息反编码 .....             | 155        |



|              |                                 |            |
|--------------|---------------------------------|------------|
| 8.2.3        | 地理信息编码查询                        | 157        |
| 8.2.4        | 实例：地理信息编码查询                     | 158        |
| 8.3          | 微定位技术                           | 160        |
| 8.3.1        | 地理围栏                            | 160        |
| 8.3.2        | iBeacon 技术                      | 160        |
| 8.3.3        | 实例：iBeacon 技术实现微定位              | 161        |
| 8.3.4        | 接近传感器                           | 169        |
|              | 本章小结                            | 171        |
| <b>第 9 章</b> | <b>iOS 敏捷开发项目实战——增强现实版本铅锤应用</b> | <b>172</b> |
| 9.1          | 应用分析与设计                         | 172        |
| 9.1.1        | 应用简介                            | 172        |
| 9.1.2        | 需求分析                            | 172        |
| 9.1.3        | 原型设计                            | 174        |
| 9.2          | iOS 敏捷开发                        | 174        |
| 9.2.1        | 敏捷开发宣言                          | 175        |
| 9.2.2        | iOS 是否可以敏捷开发                    | 175        |
| 9.2.3        | iOS 敏捷开发一般过程                    | 176        |
| 9.3          | 任务 1：创建应用并初始化工程                 | 177        |
| 9.3.1        | 任务 1.1：创建 PlumbBob 工程           | 177        |
| 9.3.2        | 任务 1.2：创建主视图控制器                 | 178        |
| 9.3.3        | 任务 1.3：创建子视图控制器                 | 181        |
| 9.3.4        | 任务 1.4：视图之间的导航                  | 183        |
| 9.3.5        | 任务 1.5：设置设备支持方向                 | 187        |
| 9.3.6        | 任务 1.6：添加铅锤图片                   | 187        |
| 9.4          | 任务 2：铅锤功能                       | 189        |
| 9.4.1        | 任务 2.1：添加铅锤 UIImageView 到主视图    | 189        |
| 9.4.2        | 任务 2.2：通过加速度计传感器控制铅锤            | 191        |
| 9.5          | 任务 3：增强现实功能                     | 193        |
| 9.6          | 任务 4：关于我们                       | 194        |
|              | 本章小结                            | 196        |

移动设备与 PC 不同之处很多,其中移动设备有很多传感器,这些传感器可以开发很多有意思的应用。可以使用麦克风获取音频,开发出一个测量风速的应用。也可以使用 GPS 定位开发出导航应用。本书将介绍苹果 iOS 设备主要的传感器应用开发。

## 1.1 本书结构

本书介绍苹果 iOS 设备主要的传感器应用开发的相关知识,其中包括手势识别、加速度计、陀螺仪、指南针、磁力计、照相机、摄像头、音频设备访问、定位和蓝牙通信等。

全书包括内容:

- (1) iOS 中的手势识别,包括手势识别的种类、如何使用手势识别器以及手势识别与事件触发。
- (2) 加速度计与陀螺仪的技术,涵盖加速度计与陀螺仪的访问与设备感知、设备检测。
- (3) 指南针与磁力计,介绍了访问磁力计的 API 与检测磁场的 API,并通过指南针应用和特斯拉计应用详细讲解这两个传感器的使用。
- (4) 照相机与摄像头,包括照片的捕捉与处理以及视频捕获的两种方式。
- (5) 音频设备访问,首先介绍了音频设备,然后详细介绍了音频 API,最后介绍了音频的播放与录制。
- (6) 定位,包括定位的 API、距离过滤器、精度定位以及地理信息反编码与编码查询。
- (7) 蓝牙通信,主要介绍了低功耗蓝牙技术。
- (8) 增强现实版的铅锤应用,总体介绍了 iOS 传感器实战开发技巧和 iOS 项目敏捷开发过程。

## 1.2 本书中的约定

为了方便使用本书,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方面的书，书中有很多实例代码。下载本书代码并解压代码，会看到图 1-1 所示的目录结构。

```

__ch2
|__2.2.3
|__TapGestureRecognizer(IB)
|__TapGestureRecognizer(代码)
|__2.2.4
|__LongPressGestureRecognizer(IB)
|__LongPressGestureRecognizer(编程)
|__2.2.5
|__PanGestureRecognizer(IB)
|__PanGestureRecognizer(编程)
|__2.2.6
|__SwipeGestureRecognizer(IB)
|__SwipeGestureRecognizer(编程)
|__2.2.7
|__RotationGestureRecognizer(IB)
|__RotationGestureRecognizer(编程)
|__2.2.8
|__PinchGestureRecognizer(IB)
|__PinchGestureRecognizer(编程)
|__2.2.9
|__ScreenEdgePanGestureRecognizer(编程)
|__2.3.3 EventInfo
|__EventInfo
|__2.3.4
|__PinchGestureRecognizer
|__TapGestureRecognizer
__ch3
|__3.1.2 MotionManager
|__3.1.4 MotionManager
|__3.1.6 DeviceOrientation
|__3.2.1 Gyroscope
|__3.2.3 MotionShake
__ch4
|__4.1.3 Compass
|__4.2.2 Teslameter
__ch5
|__5.2.2 TakePicture
|__5.2.3 TakePicture
|__5.2.4 TakePicture
|__5.2.5 TakePicture
|__5.3.2 Ballon
|__5.3.3 Ballon
__ch6
|__6.2.2 MediaPlayer
|__6.2.4 AudioRecorder
__ch7
|__7.2.2
|__BLECentral
|__BLEPeripheral
|__7.3.2 Temperature
__ch8
|__8.1 MyLocation
|__8.2.2 ReverseGeocode
|__8.2.4 GeocodeQuery
|__8.3.3
|__APIClient
|__BeaconDevice
|__8.3.4 ProximitySensorSample
__ch9
|__PlumbBob
  
```

图 1-1 本书实例目录结构

目录 ch2~ch9 代表第 2 章~第 9 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以下几种形式:

(1) 二级目录标号,如 8.1 My Location 说明是第 8 章第 8.1 节中使用的 My Location 工程实例。

(2) 三级目录标号,如 5.2.2 Take Picture 说明是第 5 章第 5.2.2 节中使用的 Take Picture 工程实例。

(3) 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,如 PlumbBob 说明是在第 9 章中使用的。

## 1.2.2 图示约定

为了更形象有效地说明知识点或描述操作,本书添加了很多图示。下面对图示中的一些符号含义进行简要说明。

### 1. 图中的圈框

有时会看到图 1-2 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、需要引起注意的内容。



图 1-2 图中圈框

### 2. 图中的箭头

如图 1-3~图 1-5 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 1-3 使用实线箭头表示选中控件修改属性。图 1-4 使用虚线箭头表示操作。图 1-5 所示箭头代表界面的跳转。

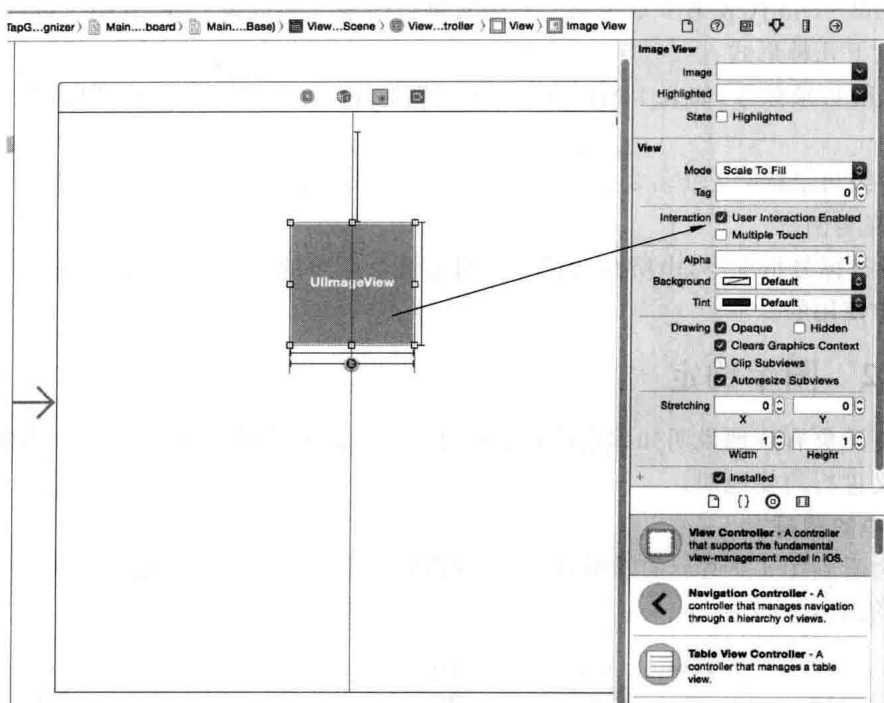


图 1-3 图中箭头 1

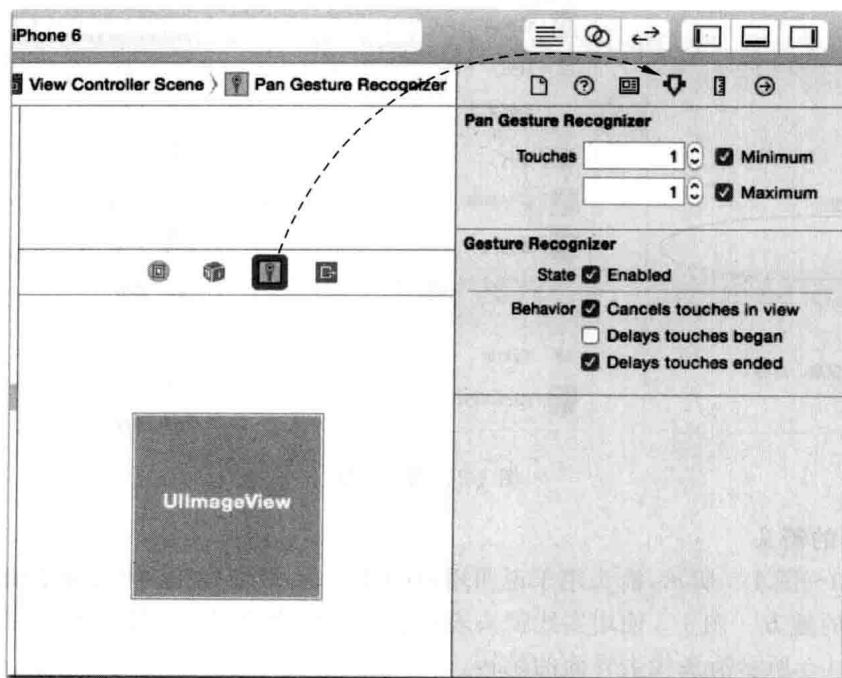


图 1-4 图中箭头 2

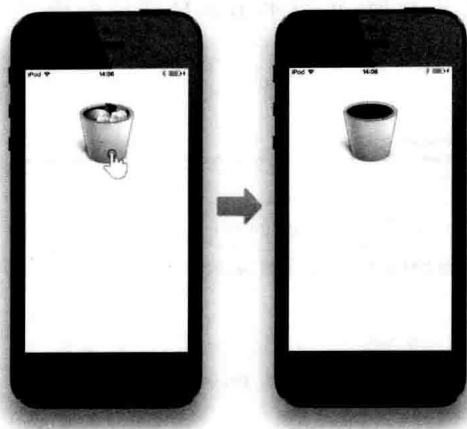


图 1-5 图中箭头 3

上述内容包含本书使用的大部分符号,一些个别符号会在使用时具体说明。

### 1.2.3 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 所示说明了 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,这叫作多重参数。它的返回类型是 void,方法签名是 insertObject:atIndex:。方法类型标识符中用“-”表示方法是实例方法,“+”表示方法是类方法,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:

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

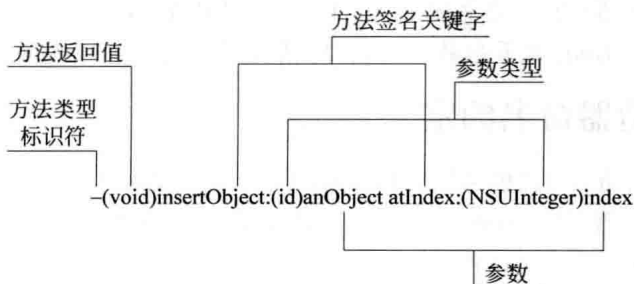


图 1-6 Objective-C 多重参数方法定义

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



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，“:”表示后面有参数。

该方法表示成 Swift 语言是如下方法：

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

该方法表示成 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。

### 1.2.4 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。

该构造器表示成 Swift 语言形式如下：

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

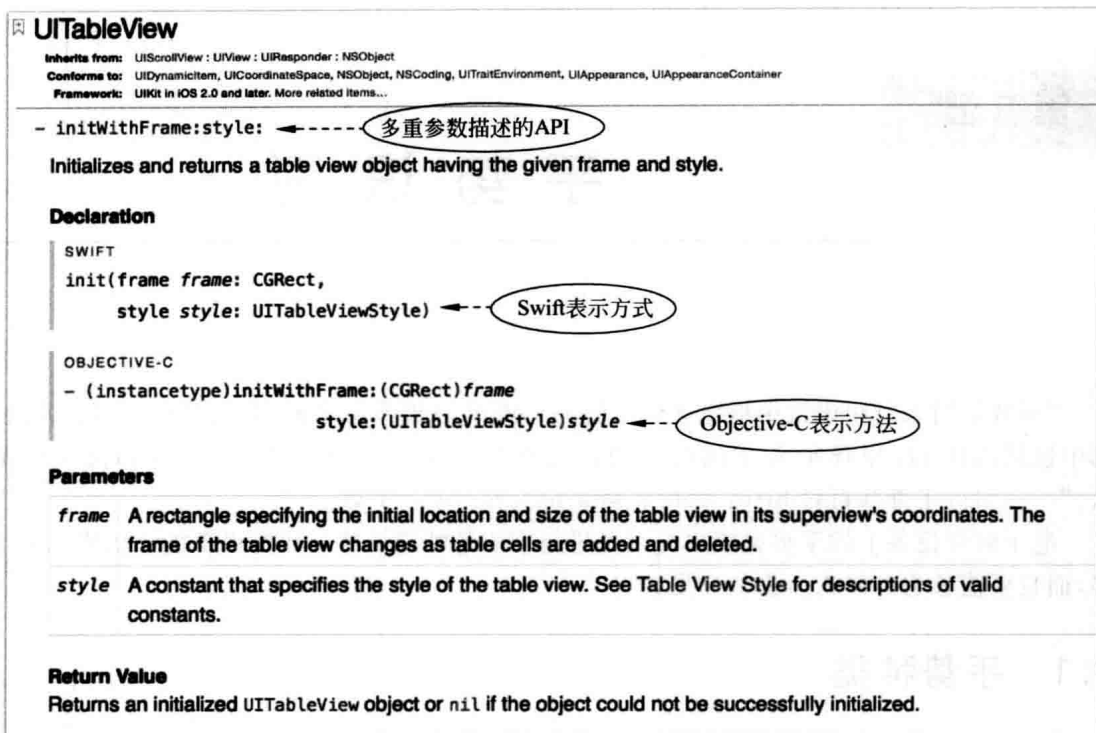


图 1-8 苹果构造器 API

该构造器表示成 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。



“手势是指人类用语言中枢建立起来的一套用手掌和手指位置、形状 of 特定语言系统。其中包括通用的,如聋哑人使用的手语。还有在特定情况下的该种系统,如海军陆战队。”——引自于维基百科 <http://zh.wikipedia.org/wiki/手势>。

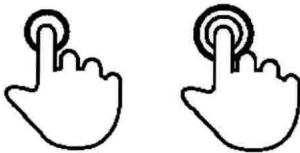
电子触屏设备上的手势是用户与设备进行交流的特定语言。作为设备能够识别这些手势,而且要能够为开发人员提供开发接口。

## 2.1 手势种类

在 iOS 设备上有极其丰富的手势,理论上说手势的种类是没有限制的,可以开发出很多诡异的手势,但是用户是否会用得 o 好就不得而知了。因此手势种类一般都是比较熟悉的几种。

在 iOS 设备中常用的手势有 Tap(单击)、Long Press(长按)、Pan(平移)、Swipe(滑动)、Rotation(旋转)、Pinch(手指的合拢和张开)和 Screen Edge Pan(屏幕边缘平移)等。这些手势如表 2-1 所示。

表 2-1 iOS 设备手势

| 手 势 名          | 手 势 图                                                                               | 说 明               |
|----------------|-------------------------------------------------------------------------------------|-------------------|
| Tap(单击)        |  | 选择、单击、碰触或连续碰触视图对象 |
| Long Press(长按) |  | 长时间按住屏幕上视图对象      |