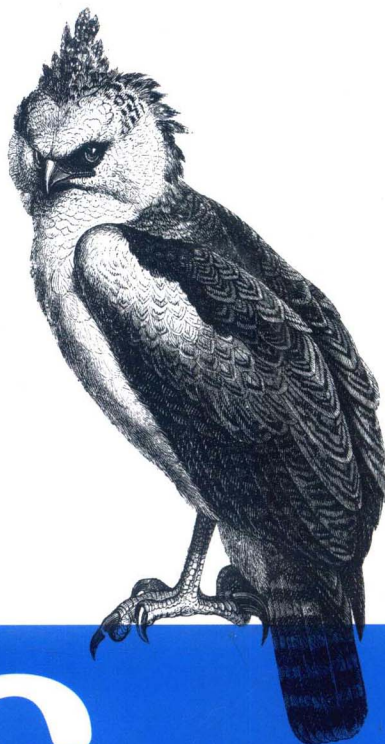




工业和信息化“十三五”
人才培养规划教材



iOS 开发基础教程

iOS Development Foundation Course

黄海 © 编著



本书从**最基础**的内容开始讲起，着重以**纯代码方式**构建各个界面；

真实案例，简单、易学；

配套 **PPT** 及**源代码**



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化部
人才培养规划教材



iOS 开发基础教程

iOS Development Foundation Course

黄海◎编著

人民邮电出版社

北京

图书在版编目 (CIP) 数据

iOS开发基础教程 / 黄海编著. — 北京: 人民邮电出版社, 2018.4

工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-44548-3

I. ①i… II. ①黄… III. ①移动终端—应用程序—程序设计—高等学校—教材 IV. ①TN929.53

中国版本图书馆CIP数据核字(2016)第326872号

内 容 提 要

本书共分13章,详细讲解iOS开发的各个知识点,内容包括iOS开发环境介绍,第一个iOS应用——“hello, world”, Cocoa Touch 框架的运行机制与开发流程, iOS 开发命名习惯与约定, iOS 用户界面元素之 UIView 与控件, 导航控制器, 故事板 Storyboard 与页面跳转, 提醒用户, 表视图之 UITableView, iOS 常用设计模式, iPad 开发之差异, 数据存储, 触摸与手势等内容。

本书既可作为高等院校本、专科计算机相关专业的教学用书,也可作为社会培训机构的参考用书,还可作为iOS开发爱好者的自学读物。

◆ 编 著 黄 海
责任编辑 范博涛
责任印制 马振武

◆ 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>
固安县铭成印刷有限公司印刷

◆ 开本: 787×1092 1/16
印张: 13.5 2018年4月第1版
字数: 334千字 2018年4月河北第1次印刷

定价: 39.80元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147号

2007年1月iPhone的诞生，引发了智能手机的革命，移动互联网浪潮席卷整个世界。如今人们的生活处处离不开智能手机，各类App已经深入人们的生活，人们开始习惯在手机上娱乐与消费，兴趣从桌面计算机全面地转向手机。苹果公司在智能手机市场中占据了相当比例的份额。很多公司开发产品，都是先开发苹果版，之后再做安卓版，因此iOS开发的重要性不言而喻。

然而各大院校，虽然有软件专业、移动应用开发专业，但都以Android开发为主，极少以iOS为主要教育方向的，这其中固然有硬件、成本等方面的原因，但有一个事实不能忽略，即缺乏合适的应用型教材。本书正由此而生。

本书共13章，从最基础的内容开始讲起，到讲完iOS的界面交互代码编写，着重于以纯代码方式构建各个界面，既要保证内容的质量，也不至于篇幅泛滥。各章节都配套了相关的源代码，方便读者学习使用。在各章节的末尾还设置了作业，供延展知识和练习使用。

iOS界面构建时，超过50%的工作都要用到UITableView，它是iOS精心设计、功能强大的视图控件，因此本书也花了最大的篇幅来详细讲解，读者宜多花心思钻研该控件。

可视化界面构建方面，使用Storyboard和自动约束的相关内容，已经足够再写一本书，因而本书只是大概涉及，毕竟掌握了纯代码的构建方式，再学可视化构建将手到擒来。

网络连接方面本书也未涉及，因涉及网络的各方面知识还有数据解析，均需大量篇幅，因此作罢，若掌握本书内容后，可自学掌握。

苹果公司每年都会更新iOS版本，本书出版时为iOS 10，使用开发工具为Xcode 8，所有的代码都在该平台下运行通过。

本书所涉及到的PPT和源代码文件可从出版社网站（www.rjjaoyu.com）下载使用。

编者

2017年11月

第1章 iOS 开发环境介绍 1	2.2 AppDelegate.m 添加代码..... 16
1.1 开发前准备..... 2	2.2.1 创建 UIWindow 对象..... 16
1.1.1 iOS 介绍..... 2	2.2.2 创建 ViewController 对象..... 17
1.1.2 Mac OS 介绍..... 2	2.2.3 例行代码..... 17
1.1.3 开发之软硬件环境要求..... 3	2.3 ViewController.m 添加代码..... 17
1.2 Xcode 集成开发环境的安装 与使用..... 3	2.3.1 创建 UILabel 对象..... 17
1.2.1 下载与安装 Xcode..... 3	2.3.2 设定 UILabel 对象的位置..... 17
1.2.2 Xcode 界面布局与功能介绍..... 3	2.3.3 显示 UILabel 对象..... 18
1.2.3 Xcode 常用快捷键..... 4	2.4 运行程序..... 18
1.3 Objective C 语言和 Swift 语言..... 5	2.4.1 在模拟器上运行程序..... 18
1.3.1 Objective C 语言介绍..... 5	2.4.2 模拟器操作介绍..... 18
1.3.2 Swift 语言介绍..... 5	2.5 小结与作业..... 19
1.4 如何使用文档及获取帮助..... 5	第3章 Cocoa Touch 框架的运行 机制与开发流程 20
1.4.1 Xcode 帮助文档的使用..... 5	3.1 理解 UI 程序运行的机制..... 21
1.4.2 开发中遇到问题如何寻求帮助..... 6	3.1.1 程序不是顺序运行..... 21
1.4.3 提问的方式..... 6	3.1.2 用户操作, 硬件中断与消息处理..... 21
1.5 iOS 程序调试、打包与发布 到 App Store..... 7	3.2 iOS 的消息处理机制..... 21
1.5.1 苹果开发人员计划..... 7	3.2.1 协议与代理..... 21
1.5.2 在模拟器中运行程序..... 7	3.2.2 target 与 selector..... 22
1.5.3 在真机中运行程序..... 8	3.2.3 消息中心 NSNotification..... 22
1.5.4 调试命令与技巧..... 8	3.3 iOS 程序开发流程简介..... 22
1.5.5 将程序打包发布至 App Store..... 9	3.3.1 创建窗口和控件..... 22
1.6 小结与作业..... 11	3.3.2 视图与控件事件绑定..... 22
第2章 第一个 iOS 应用—— “hello,world” 12	3.3.3 编写事件响应方法..... 23
2.1 新建项目..... 13	3.4 MVC 方法..... 23
2.1.1 新建 Single View 项目..... 13	3.4.1 MVC 概述..... 23
2.1.2 项目结构..... 15	3.4.2 模型..... 23
	3.4.3 视图..... 23
	3.4.4 控制器..... 23

3.4.5 再论 MVC 的意义	24	5.5.4 项目制作——制作设置界面	43
3.5 小结与作业	24	5.6 输入控件	44
第4章 iOS 开发命名习惯与约定	25	5.6.1 UITextField 控件	44
4.1 良好的编程习惯	26	5.6.2 UITextView 控件	45
4.1.1 命名方式与一致性	26	5.6.3 项目制作——制作一个输入界面	46
4.1.2 换行与缩进	26	5.7 日期选择器 UIDatePicker	50
4.1.3 编程风格有什么用	26	5.7.1 日期选择器概述	50
4.2 iOS 的命名习惯	27	5.7.2 日期选择器属性介绍	51
4.2.1 骆驼表示法	27	5.7.3 日期换算与格式化	51
4.2.2 C++式的下划线表示法	27	5.8 自定义选择器 UIPickerView	51
4.2.3 使用汉字命名	27	5.8.1 自定义选择器概述	51
4.3 小结与作业	27	5.8.2 UIPickerView 代理	52
第5章 iOS 用户界面元素		5.8.3 项目制作——制作一个选择器	53
之 UIView 与控件	28	5.9 键盘定制与遮挡问题	56
5.1 UIView 概述	29	5.9.1 键盘种类	56
5.1.1 UIView 家族	29	5.9.2 定制输入界面	57
5.1.2 UIView 基本属性	30	5.9.3 定制辅助输入界面	57
5.1.3 视图层次体系	30	5.9.4 键盘遮挡问题	58
5.1.4 视图变换	32	5.9.5 关闭键盘	58
5.1.5 项目制作——使用计时器		5.9.6 将焦点转移至下一个输入控件	58
NSTimer 制作简单动画	32	5.9.7 项目制作——制作一个	
5.2 UIControl 类	34	日期计算器	59
5.3 标签视图类 UILabel	34	5.10 网页控件 UIWebView	62
5.3.1 UILabel 概述	34	5.10.1 网页控件概述	62
5.3.2 UILabel 内容大小计算	35	5.10.2 UIWebView 代理	62
5.4 按钮控件 UIButton	35	5.10.3 项目制作——制作一个微型	
5.4.1 按钮分类	35	浏览器	63
5.4.2 按钮美化	35	5.11 工具栏与导航栏	66
5.4.3 添加事件响应方法	36	5.11.1 工具栏 UIToolBar	
5.4.4 项目制作——制作简单计算器	36	与 UIBarButtonItem	67
5.5 开关控件、滑块控件与分段控件	41	5.11.2 导航栏 UINavigationController	
5.5.1 开关控件	41	与 UINavigationController	68
5.5.2 滑块控件	42	5.11.3 总结 UIBarButtonItem	70
5.5.3 分段控件	42	5.12 UIView 动画	70
		5.12.1 动画概述	70
		5.12.2 动画的几种方式	70

5.13 子类化 UIControl:设计自己的控件	71	8.4 进度条控件 UIProgressView	101
5.14 小结与作业	74	8.5 小结与作业	102
第6章 导航控制器	76	第9章 表视图之 UITableView	103
6.1 导航控制器概述	77	9.1 表视图概述	104
6.1.1 UIViewController 的几个属性	77	9.1.1 表视图结构	105
6.1.2 导航设计	79	9.1.2 相关类	105
6.2 导航控制器		9.1.3 表视图种类	105
UINavigationController	80	9.1.4 单元格样式与定制	106
6.2.1 关于导航栏	80	9.1.5 表视图协议	108
6.2.2 关于工具栏	85	9.2 项目制作——第一个表视图	
6.3 标签页控制器		项目	109
UITabBarController	86	9.2.1 建立项目	109
6.3.1 设置图标	86	9.2.2 准备数据	112
6.3.2 多于五个标签的情况	86	9.2.3 实现表视图协议	113
6.4 小结与作业	87	9.3 分节表	114
第7章 故事板 Storyboard		9.3.1 分节表概述	114
与页面跳转	88	9.3.2 项目制作——简单通讯录制作	115
7.1 故事板 Storyboard 概述	89	9.3.3 建立表索引	117
7.2 使用 Storyboard 设计界面	89	9.4 搜索栏制作	119
7.2.1 通过鼠标拖拉建立控件	89	9.4.1 UISearchBar 和	
7.2.2 大小、位置等属性控制	91	UISearchController	119
7.3 Storyboard 如何与代码配合	91	9.4.2 NSPredicate 使用	123
7.3.1 控件指定类	91	9.5 表的增删改	124
7.3.2 UIStoryboard 类	92	9.5.1 删除单元格	125
7.3.3 UIStoryboardSegue 类	92	9.5.2 增加单元格	126
7.4 给控件绑定事件与实现跳转	95	9.5.3 改动及刷新单元格	131
7.5 小结与作业	97	9.6 项目制作——制作简单记事本	138
第8章 提醒用户	98	9.6.1 建立项目	138
8.1 警告框 UIAlertView	99	9.6.2 记事本列表页面制作	140
8.2 操作表 UIActionSheet	100	9.6.3 记事本添加与编辑页面制作	144
8.3 活动指示器		9.6.4 添加搜索框	149
UIActivityIndicatorView	101	9.7 小结与作业	151
		第10章 iOS 常用设计模式	152
		10.1 单例模式	153

10.2 委托模式	154	12.3 CoreData.....	181
10.3 观察者模式.....	154	12.3.1 建立数据模型.....	181
10.4 小结与作业.....	155	12.3.2 生成实体类.....	184
第11章 iPad 开发之差异.....	156	12.3.3 项目制作——制作一个表格.....	184
11.1 iPad 开发概述	157	12.4 小结与作业.....	190
11.2 iPad 专用 API	157	第13章 触摸与手势	191
11.2.1 UIPopOverController 控制器... 157		13.1 触摸与手势概述	192
11.2.2 UISplitViewController 控制器... 163		13.2 点按手势	193
11.3 小结与作业	167	13.2.1 点按手势概述.....	193
第12章 数据存储.....	168	13.2.2 项目制作——制作“按钮”	193
12.1 文件保存与恢复	169	13.3 捏拉缩放与旋转手势.....	197
12.1.1 沙盒机制	169	13.3.1 捏拉缩放与旋转手势概述.....	197
12.1.2 数据类 (NSArray、NSDictionary 等) 的保存与恢复.....	169	13.3.2 项目制作——制作图片查看器 ... 197	
12.1.3 文件操作相关类	174	13.4 手势冲突问题	204
12.2 内置数据库 sqlite3	176	13.5 滑动类手势.....	205
12.2.1 概述.....	176	13.5.1 滑动类手势概述	205
12.2.2 sqlite3 接口函数介绍.....	177	13.5.2 项目制作——制作竖向滚动条 ... 205	
		13.6 小结与作业.....	208

Development of iOS App

Chapter

1

第 1 章

iOS 开发环境介绍



1.1 开发前准备

首先需要准备一台 Mac 计算机，装上 Mac OS X 操作系统，本书定稿时，最新版 Mac OS 为 10.12 Sierra。Mac OS X 自版本 10.10 及以后的版本都是免费的，没有 Mac 计算机的话，在某些普通 PC 上可以安装所谓的“黑苹果”来开发，只是操作体验没有 Mac 计算机好。

1.1.1 iOS 介绍

iOS 是由苹果公司开发的移动操作系统。2007 年 iPhone 诞生，当时的操作系统还不叫 iOS，而是“iPhone OS”，直到 2010 年 iPad 诞生后，因为使用同一个系统，因此才改名为“iOS”。iOS 一般每年进行一次大的升级，本书写作时，iOS 10 为最新正式版。

iOS 是 iPhone、iPad、iPod Touch 等苹果设备的灵魂，正是有了 iOS，苹果产品的使用体验才出类拔萃，目前国内很多安卓 App 的风格，均受 iOS 的影响。

1.1.2 Mac OS 介绍

Mac OS 的历史非常悠久，20 世纪 80 年代就有了，其率先实现了窗口系统，并影响了微软的 Windows 的设计。Mac OS 自乔布斯回归苹果后，架构发生了很大的改变，使用了 NextStep(乔布斯出走苹果后创办的一家软件公司的名字) 的软件，现在的 Cocoa (即苹果官方开发框架之名) 库中的众多 NS 开头的方法，就是 NextStep 的首字母缩写，尤其苹果产品转用 intel CPU 后，将底层的内核改为 UNIX，与 Free BSD (UNIX 移植到普通 PC 后的一个衍生系统，非常著名，与之齐名的还有 Net BSD、Open BSD) 的关系最为接近，自此 Mac OS 开始在程序员中非常流行，因为其既有着华丽而舒适的操作界面，又接上了 UNIX 的地气，拥有了众多 UNIX/Linux 软件的支持，因而非常受开发人员的青睐。

另外，众所周知的开源 (即公开源代码) 软件运动，就是自 UNIX 闭源开始的，20 世纪的一群黑客，为了摆脱闭源以及专利的束缚，以 Richard Stallman (理查德·斯托曼) 为首带头发起了开源运动，其创办了“自由软件基金会”(Free Software Foundation)，组织了 GNU 项目，将 UNIX 的软件一一重写，企图制作一个完整的、开源的、完全自由的类 UNIX 操作系统。在这些突破的基础上，开源软件轰轰烈烈地一直健康发展到如今，连微软都不得不选择开源。Mac OS 建立在 UNIX 之上之后，拥有了完备的 UNIX 基础，对接上了这众多的开源软件，比较而言，Windows 平台却显得不方便了。

1.1.3 开发之软硬件环境要求

类似微软公司鼎鼎大名的 Visual Studio，苹果公司同样为开发人员准备了统一的免费的发展环境——Xcode。其功能很强大，本书所有的示例代码均在 Xcode 之下开发。

综上所述，开发需要的软件环境为：Mac OS X + Xcode，硬件则需要 Mac 计算机，或者可以安装“黑苹果”的 PC。

1.2 Xcode 集成开发环境的安装与使用

1.2.1 下载与安装 Xcode

最简单的方法是打开 Mac OS X 的 App Store，在其中搜索 Xcode，如图 1-1 所示。



图 1-1 Mac App Store

随后只需简单地选择安装即可，或者直接去苹果官网下载亦可。

1.2.2 Xcode 界面布局与功能介绍

Xcode 界面如图 1-2 所示。

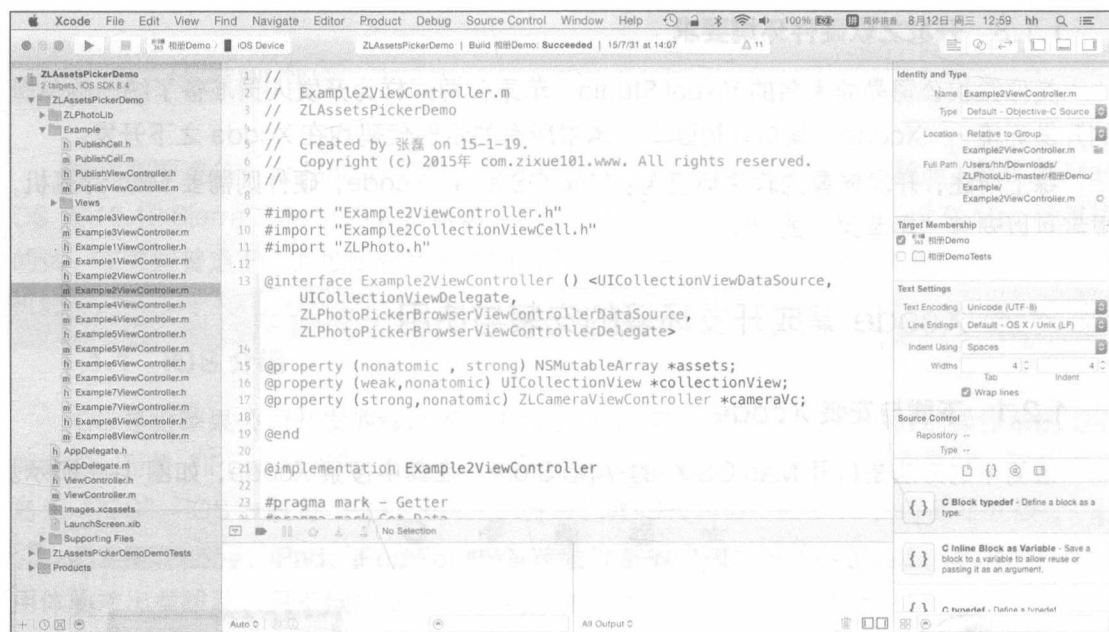


图 1-2 Xcode 界面

与一般的 IDE 布局差不多，左边是项目文件树列表，中间是编辑代码区域，下面是调试信息展示，右边上面是一些选项调整和帮助信息，下面是一些额外功能及控件列表等，并可以自由定义哪些界面不显示。

1.2.3 Xcode 常用快捷键

掌握快捷键对于快速的开发是非常有用的。

首先，编辑快捷键默认是 Emacs 格式的，比如：

- `control+A` 光标移动到行首；
- `control +E` 光标移动到行尾；
- `control +P` 光标移动到上一行；
- `control +N` 光标移动到下一行；
- `control +K` 从光标位置到行尾的内容都删除；
- `control +K` 删除当前行。

注：在 PC 键盘上 `control` 键等同于 `Ctrl` 键

复制粘贴操作则是：

- `command+A` 全选；
- `command+C` 复制；
- `command+V` 粘贴；
- `command+X` 剪切。

其他快捷键:

- command+单击找到光标下的符号的定义(重要!最常用);
- command+R 运行。

基本上掌握了这些快捷键就够用了。

1.3 Objective C 语言和 Swift 语言

1.3.1 Objective C 语言介绍

不同于市面上其他流行的语言, Objective C 语言是因为苹果公司才发展壮大的。其语言的特点与其他的主流语言相差较大, 但是功能一点都不弱, 用习惯了非常方便。Objective C 语言与 C++ 都是通过给 C 语言加上面向对象功能发展而来, C++ 的体量已经庞大到和 C 语言差不多, 而 Objective C 却保持了小巧的体积。Objective C 更多的是借鉴了面向对象语言 smallTalk 的特点, 强调对象之间通过发送消息来通信, 其独特的方括号语法处处都有, 比如: [object reload] 表示向 object 对象发送一个名为 reload 的消息 (reload 实质上是 object 对象的一个方法)。

1.3.2 Swift 语言介绍

Swift 是苹果公司于 2014 年 6 月的 WWDC 大会上发布的一种新语言, 由苹果公司完全开发, 未来将替代 Objective C。Swift 像脚本语言, 但又拥有编译运行的速度, 拥有很多现代语言的特征, 非常适合于快速开发。但从目前来看, 2 年多的时间, Swift 发展到 3.0 版本, 语法变化很大, 而且不能向前兼容, 导致之前用 Swift 2.0 编写的代码必须大量修改才能运行 (当然苹果公司在 Xcode 8 提供了自动转换新语法的功能, 但是仍然不能完全转换, 需人工介入), 因此 Swift 语言短时间内还替代不了 Objective C。

1.4 如何使用文档及获取帮助

1.4.1 Xcode 帮助文档的使用

在代码中, 可以按住 Command 键不放, 此时用鼠标单击某类或者变量, 可以跳到其定义处。读者可以通过这种方式查看 Cocoa 类的头文件, 查看有哪些属性和方法, 这对于熟悉某个类的使用方法非常有帮助。

还有一种方法, 鼠标停留在某处时, 查看 Xcode 右边的帮助界面, 如图 1-3 所示。



图 1-3 Xcode 帮助

右边的帮助界面，显示为鼠标此时所在位置（一般为类名、方法名等，图 1-3 中鼠标位置在最下面红圈处的 `initWithItems` 方法处）的帮助内容，可以单击其中蓝色的链接进行详细查看。

1.4.2 开发中遇到问题如何寻求帮助

开发 App 时，会不可避免地遇到难以解决的问题或错误。如果通过调试始终不能找到错误所在，第一种方法是把错误信息复制到搜索引擎进行搜索。一般而言，谷歌的搜索结果最好，但是因为某种原因，谷歌在中国大陆不能使用，可以代之以微软的搜索引擎“必应”（<http://www.bing.com>），百度的结果相对差点。

第二种方法是上论坛提问。这方面的资料一般国外的多，最有名的当属 Stack Overflow（<http://www.stackoverflow.com>），通常别的地方找不到的问题这上面都有。国内的有 cocoa china（<http://www.cocoachina.com>）较为不错。

第三种方法是找一些 iOS 开发方面的 qq 群。

第四种方法是上 github（<http://www.github.com>），它可以说是现在世界上最大的开源项目中心，还有无数的第三方库可供使用。有些问题，自己不好解决，完全可以通过第三方库来解决。另外，通过这些开源的项目，可以学习到高手的代码，对自己非常有用。

1.4.3 提问的方式

在网上求助时，要注意提问的方式。要把环境描述清楚，把错误信息粘贴完整，并突出重点。他人没有义务一定解答，所以要为别人考虑，尽量把信息组织得清晰易懂，他人提供帮助时也更加方便。

笔者曾经使用 github 的一个世界著名的开源库 YYKit，使用中我发现了一些 bug，当即

在 github 的该项目的主页上报告了该 bug (在 issue 选项卡中报告), 因为描述得很详细, 错误定位得很清楚, 该项目维护者在当天就回应并解决了该错误, 提交了新版本代码。

1.5 iOS 程序调试、打包与发布到 App Store

1.5.1 苹果开发人员计划

作为开发者, 我们首先得有一个 Apple ID (开发者账号), 然后去 <https://developer.apple.com> 申请开发人员身份。

iOS 的开发, 仅在模拟器上运行, 是免费的, 从 2014 年后, 在真机上运行, 也免费了, 但如果需要发布到 App Store 上进行销售, 需要申请开发人员计划, 这个就不是免费的了。

从 2015 年 6 月的 WWDC 大会之后, 苹果的开发人员计划进行了合并, 只区分个人开发者和企业开发者, 其中个人开发者一年需向苹果公司缴纳 99 美元或人民币 688 元的费用, 企业开发者则需要缴纳 299 美元。企业开发者可以开发私有 App, 但不能通过 App Store 分发, 只能通过网站的方式在企业内部分发。个人开发者则只能通过 App Store 进行发布。

只要获取了开发者资格, 就可以面向苹果所有平台 (Mac、iOS、Apple Watch、Safari) 进行开发和销售。

在 App Store 上架销售的 App, 默认是面向全世界销售, 也可以自己控制在哪些国家销售。销售所获得的收入, 苹果将获得三成, 开发者获得七成。

1.5.2 在模拟器中运行程序

Xcode 安装好后, 就默认安装了 iOS 模拟器, 可以方便地运行自己开发的 App。值得一提的是, 与一般人对模拟器的印象不同, iOS 模拟器的运行速度非常快, 有时候甚至比真机还快, 一点都不卡, 启动速度也很快, 使用起来令人感觉愉快, 如图 1-4 所示。

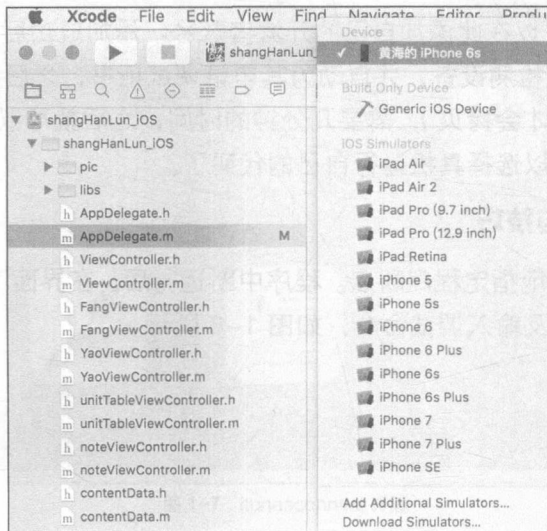


图 1-4 iOS 模拟器选择菜单

选择好运行的设备，按下快捷键 `command+R` 即可运行。

1.5.3 在真机中运行程序

目前要在真机中运行调试 App，必须拥有开发者证书。有些功能的开发用真机方便些，比如定位、照相、方向感知等用到硬件功能的地方。

有了开发者账号后，可以在 Xcode 之中按下 `command+,` (逗号)，打开设置，如图 1-5 所示。

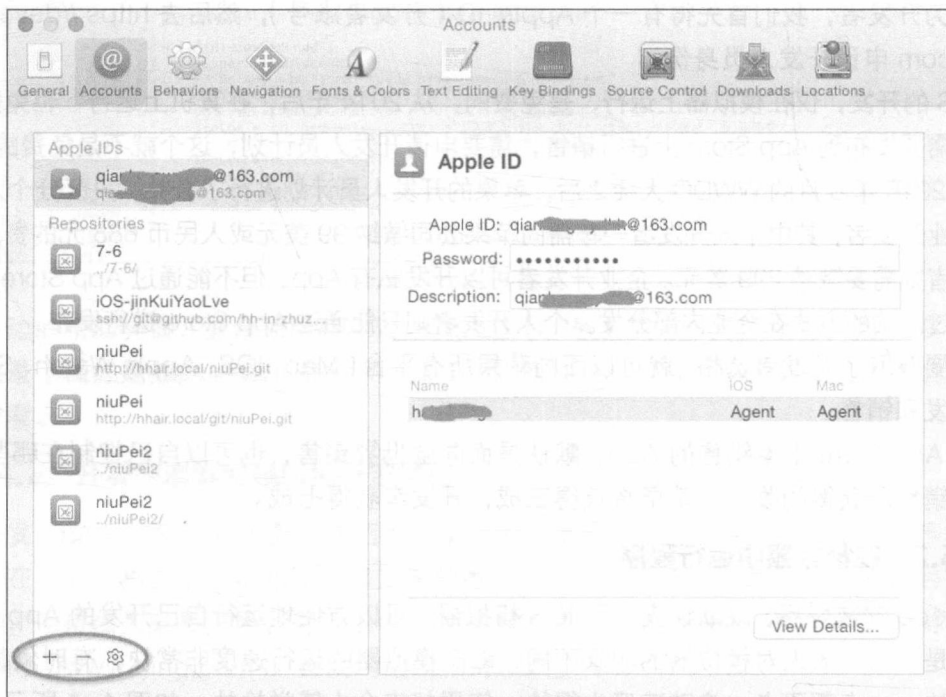


图 1-5 Xcode 设置

如图 1-5 所示，可以在此添加自己的开发者账号。添加成功后，将 iPhone 连接到计算机，Xcode 将会自动检测设备，并自动为其申请调试证书，之后会从 iPhone 中拷贝调试符号（第一次调试时才会拷贝），需要几分钟的时间，之后就可以在图 1-4 中看到自己的真机设备，此时就可以选择真机运行自己的代码了。

1.5.4 调试命令与技巧

Xcode 中可以方便地指定程序断点。程序中断运行后，在界面下面的调试小窗口中可以方便地查看变量值以及输入调试命令，如图 1-6 所示。



图 1-6 Xcode 调试

在代码左侧空白处单击鼠标即可插入断点；左下角可以查看在断点处的环境变量值；右下角的 (lldb) 处可以输入调试命令。lldb 是 Xcode 的调试器程序。常用的 lldb 命令有 p (打印基础类型变量值)、po (打印 NSObject 或其子类型的值) 等。

1.5.5 将程序打包发布至 App Store

App 在编写完毕后，便可以发布到 App Store 上。先要登录网址 <http://itunesconnect.apple.com>，登录后如图 1-7 所示。



图 1-7 itunesconnect 界面