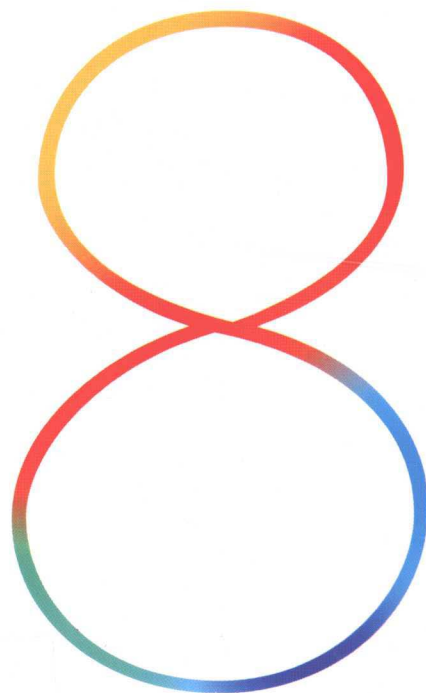


GOTOP

iOS

应用开发实战

205个快速上手的开发技巧



- ◆ 使用最新的iOS 8+Xcode 6进行App开发，并涵盖CloudKit、HealthKit、Extension、iBeacon等重要技术
- ◆ 从最核心并且极为关键的问题下手，有效降低学习门槛
- ◆ 让任何一位想要开发App的读者都可以快速上手

宏碁智能手机
事业部处长 **游鸿志**
104人力银行
软件开发部经理 **李闵隆**

凌网科技
资深经理 **徐长恺**
明道大学
信息传播系主任 **李绍群**
致理技术学院
国贸系副教授 **雷立伟**

**联袂
推荐**

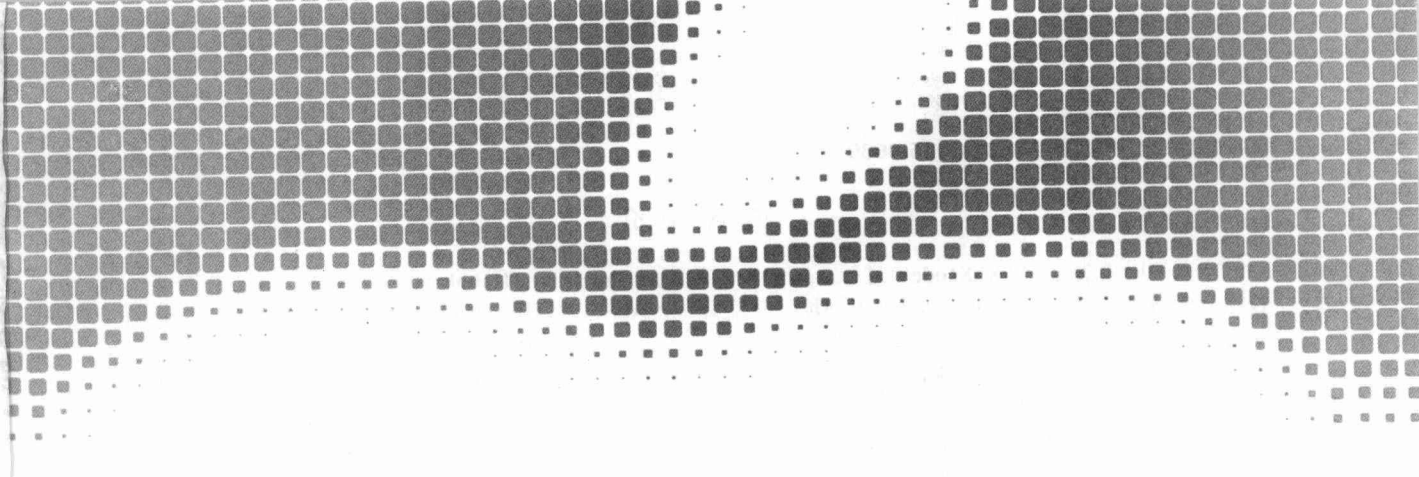
朱克刚
著



本书提供程序代码下载



清华大学出版社

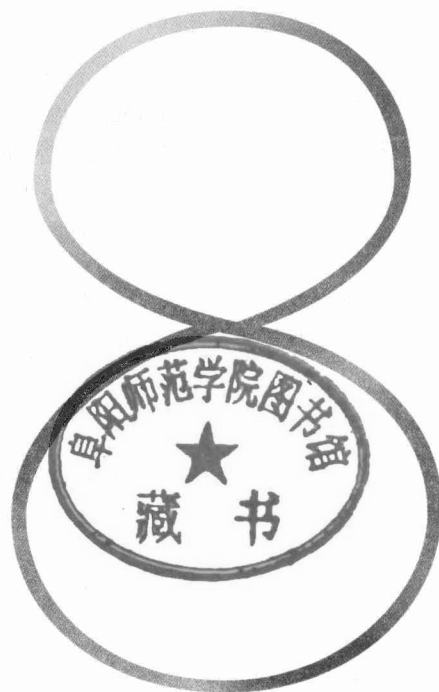


iOS

应用开发实战

205个快速上手的开发技巧

朱克刚 著



清华大学出版社
北京

本书版权登记号：图字：01-2015-0839

内 容 简 介

本书使用最新的 iOS 8+Xcode 6 进行 App 开发，并涵盖 CloudKit、HealthKit、Extension、iBeacon 等重要技术。全书共 22 章，内容包括熟悉开发环境、开发第一个 App、掌握用户界面，以及表格、动态行为、拍照与音乐、手势、传感器、绘图、文件管理、结构化数据访问、云端存储、多线程、后台运行、Internet、媒体获取、通信、CloudKit、HealthKit、Extension 等。本书从最核心并且极为关键的问题下手，以大量案例为导向，有效降低学习门槛与花费的时间，让广大 App 开发者能够快速上手。

本书适合作为 iOS 初学者、iOS 程序员、iPhone 开发人员、iPad 开发人员的参考书，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。

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 8 应用开发实战：205 个快速上手的开发技巧/朱克刚著. - 北京：清华大学出版社，2015
ISBN 978-7-302-40324-1

I. ①i… II. ①朱… 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 104901 号

责任编辑：夏非彼

封面设计：王 翔

责任校对：闫秀华

责任印制：李红英

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

印 装 者：清华大学印刷厂

经 销：全国新华书店

开 本：190mm×260mm

印 张：27.25

字 数：698 千字

版 次：2015 年 7 月第 1 版

印 次：2015 年 7 月第 1 次印刷

印 数：1~3500

定 价：69.00 元

产品编号：063030-01

推荐序

移动应用的程序设计就像它的手机配件一样，其发展是日新月异。

对于广大有需求的开发者而言，重点不是找到一本书教授用户如何编写手机程序，而是找到一本书，让用户可以紧跟最新的手机平台发展，让用户可以把手机最新的功能发挥得淋漓尽致，让用户可以和其他的竞争者拉开距离。

本书满足了用户寻找最新技术的强烈需求，所有 iOS 8 (iPhone 6) 上最新的功能 CloudKit、HealthKit、Extension，还有最新、最红的 iBeacon 应用，全部在本书的涵盖范围内。

本书以大量案例为导向，在被一堆理论淹没之前，马上打造属于自己的应用。从实践中学习，从而快速解决眼前的问题，这正是本书的精华。

在追逐技术的过程中，用户往往需要从一堆英文资料中学习跟得上潮流的技能。虽然吃力，但是也有太多无奈。在这种传统学习的趋势之下，竟然还有这样一本中文参考资料，而其分量不亚于英文信息，更是让我在受用之余，还多了一份小小的感动。

如果读者已经泡好一杯咖啡准备向长夜挑战，请记得你的手边要带上这本书。

游鸿志

宏碁智能手机事业部处长

序

每年 6 月 Apple Worldwide Developers Conference (WWDC, 苹果全球开发者大会) 过后, 就是跟时间赛跑的开始。除了要快速了解 Apple 到底在新版的 iOS 中加入了哪些新东西外, 还要了解他们删除了哪些旧东西, 并且还要忍受手机装了 Beta 版后偶尔死机的状况, 这是很忙碌的一段时间。

iOS 8 进行了非常大的改版, 增加了许多亮点功能, 大部分的亮点功能都已经写在本书中 (例如 Size Classes Tool、CloudKit、HealthKit、Extension 等), 少部分因为出版时间的压力, 只能暂时舍弃 (例如 Handoff), 我认为写在书中的亮点功能都是非常重要的, 极有机会引领一波新的科技浪潮与应用, 所以如果在这本书中不放进去, 恐怕有违读者的期待, 也让本书黯然失色。

感谢上一本 iOS 7 读者的捧场, 让该书创造了畅销排行榜第二名的佳绩, 当然这样的成果也让本书在改版到 iOS 8 时颇有压力。这里特别感谢几位朋友的帮忙, 帮助确认书中的程序代码在 iOS 8 中是有效的, 他们是江承翰、柯明伦与梁国基。

代码下载

本书配套源代码的下载地址: <http://pan.baidu.com/s/1c0tXlWc>, 若下载有问题, 请发送电子邮件至 booksaga@126.com, 邮件标题为“求代码, iOS 8 应用开发实战”。

最后, 祝福各位读者, 所有事情都能顺心愉快。

朱克刚

2015.03

目 录

第 1 章 绪 论.....	1
1-1 移动设备里的小确幸.....	1
1-2 注册与下载开发工具.....	2
1-3 使用的语言.....	4
1-4 Framework 概述.....	7
1-5 设计 App.....	8
1-6 本书在讲述什么.....	9
1-7 如何使用本书.....	10
第 2 章 开发环境.....	11
2-1 下载集成开发环境.....	13
2-2 在 Mac 计算机上制作开发专用证书.....	14
2-3 将 App 发布到手机上测试.....	18
2-4 在 Mac 计算机上创建发布专用证书.....	20
2-5 在 iTunes Connect 中创建 App 的上架信息.....	21
2-6 上传 App 进行审核.....	26
2-7 模拟器的基本功能.....	29
2-8 如何增加函数库.....	31
第 3 章 Storyboard.....	32
3-1 我的第一个 App——Hello World.....	37
3-2 让两个按钮调用同一个事件处理方法.....	42
3-3 一个 App 多个画面.....	44
3-4 画面切换.....	45
3-5 使用导航控件.....	47
3-6 使用 Tab Bar 切换画面.....	48
3-7 传递参数到下一个画面.....	52
3-8 取消画面切换.....	54
3-9 不使用 Segue 的画面切换.....	55
3-10 在运行阶段加载 XIB 文件.....	56
3-11 使用 Auto Layout 布局.....	58
3-12 开发同时适合 iPhone 与 iPad 的 App.....	60
3-13 Image 在 Size Classes 上的应用.....	63
3-14 判断设备类型以及取得位置信息.....	65

第 4 章 用户界面	67
4-1 弹出警告信息框	69
4-2 利用警告信息框输入账号和密码	70
4-3 使用 Picker View 选择数据	72
4-4 使用 Date Picker 设置时间	76
4-5 使用 Date Picker 设置倒数时间	78
4-6 使用 Image View 显示图片	79
4-7 使用 Image View 连续播放图片	82
4-8 在运行阶段动态产生可视化组件	83
4-9 使用 Slider	85
4-10 使用 Switch	87
4-11 让图片加上滚动条	88
4-12 使用滚动条分页浏览	91
4-13 关闭虚拟键盘	93
4-14 使用 Web View 加载外部网页	94
4-15 使用 Web View 显示 HTML 字符串	96
4-16 使用分页控制器	96
4-17 使用 Search Bar 搜索数据	99
4-18 加入 iAd 广告	102
4-19 利用动画方式呈现按钮移动	104
4-20 旋转图片	105
4-21 将直角改成圆角	106
4-22 加上阴影	108
第 5 章 表格	110
5-1 使用表格组件	113
5-2 显示单元格指示器	116
5-3 在表格上创建两个区块	118
5-4 在表格上删除一条数据	120
5-5 在表格上新增一条数据	121
5-6 得知用户单击哪一个单元格	123
5-7 改变单元格顺序	124
5-8 如何自定义单元格样式	126
5-9 使用表格控件	129
5-10 导航控件与表格	132
5-11 使用 Collection 组件	133
5-12 表格下拉更新	137
第 6 章 动态行为	139
6-1 吸附	139
6-2 碰撞	142
6-3 引力	143
6-4 推力	144

6-5 扑向	146
第 7 章 拍照与音乐	148
7-1 让手机震动	149
7-2 检测设备上是否配备相机与闪光灯	150
7-3 打开相机拍照并保存文件	151
7-4 从相册中挑选一张照片	153
7-5 一次取得相册中的所有照片	156
7-6 播放 App 内置的音乐	157
7-7 显示目前音乐播放进度	158
7-8 播放已经存在的音乐	160
7-9 取得目前播放中的歌曲信息	162
第 8 章 手势	166
8-1 使用 Tap 手势	167
8-2 使用 Pinch 手势	168
8-3 使用 Rotation 手势	169
8-4 使用 Swipe 手势	169
8-5 使用 Pan 手势	170
8-6 使用 Long Press 手势	171
第 9 章 传感器	173
9-1 读取加速器数据	175
9-2 读取陀螺仪数据	177
9-3 读取磁力仪数据	178
9-4 读取 Device Motion 数据	179
9-5 检测设备摇晃	182
9-6 使用加速器来控制赛车的方向盘	183
9-7 使用距离传感器	186
第 10 章 绘图	189
10-1 取得绘图区域	190
10-2 在绘图区域上画线	191
10-3 在绘图区域绘制多边形	193
10-4 在绘图区域绘制矩形	195
10-5 在绘图区域绘制弧线	196
10-6 在绘图区域绘制椭圆形	197
10-7 在绘图区域绘制曲线	199
10-8 在绘图区域显示图片	200
第 11 章 文件管理	202
11-1 创建目录与文件	203
11-2 复制、移动与删除	204
11-3 检查目录或文件是否存在	205

11-4	区分目录与文件	206
11-5	列出目录下的所有文件	207
11-6	设置文件不要备份属性	208
11-7	文字类型的文件 I/O	209
11-8	数组类型的文件 I/O	210
11-9	将图片存入文件	211
11-10	delegate 的用法	212
第 12 章	结构化数据访问	214
12-1	访问 PLIST 文件	216
12-2	创建 SQLite 数据库	218
12-3	连接 SQLite 数据库	219
12-4	查询数据库中的数据	222
12-5	修改数据库中的数据	223
12-6	数据库访问图片	225
12-7	设计与规划 Core Data	226
12-8	访问 Core Data 数据	230
12-9	查询 Core Data 时附带查询条件	232
12-10	使用 Core Data 预存的 Fetch Requests	233
12-11	在 Core Data 预存的 Fetch Requests 中增加变量	234
12-12	删除 Core Data 中的数据	235
12-13	访问 Core Data 中的一对多关系	236
12-14	Core Data 访问图片	238
第 13 章	云端存储	240
13-1	使用 iCloud 读写文件	241
13-2	使用 iCloud 读写 Key-Value 数据	243
13-3	让两个 App 共享 iCloud 下的数据	244
13-4	注册 Dropbox App 及下载 SDK	244
13-5	在项目中安装与设置 Dropbox SDK	247
13-6	上传文件至 Dropbox	249
13-7	从 Dropbox 下载文件	251
13-8	取得 Dropbox 上的文件列表与文件信息	254
第 14 章	多线程	257
14-1	利用传统方式打开多线程	259
14-2	使用 NSOperationQueue	261
14-3	使用 NSOperation 类	262
14-4	使用 Main 派遣队列	264
14-5	使用 Concurrent 派遣队列	265
14-6	使用 Serial 派遣队列	266
14-7	定时器	267

第 15 章 后台运行	269
15-1 后台播放音乐	272
15-2 后台更新地理坐标	273
15-3 额外争取 10 分钟的后台运行时间	273
15-4 多线程的后台运行	275
15-5 后台获取	277
第 16 章 Internet	279
16-1 异步方式下载网页	280
16-2 同步方式下载网页或图片	281
16-3 解析 XML	282
16-4 解析 JSON	284
16-5 以 GET 方式发送数据	286
16-6 以 POST 方式发送数据	286
16-7 与社区网站连接	287
16-8 信息推送	288
16-9 本地信息推送	291
16-10 与推送信息互动	292
第 17 章 媒体获取	297
17-1 获取静态图片并预览	298
17-2 前后镜头切换	302
17-3 调整相机参数	303
17-4 录制影片	304
17-5 录制声音	308
第 18 章 通信	312
18-1 蓝牙 4.0 Peripheral	313
18-2 蓝牙 4.0 Central	316
18-3 iBeacon	320
18-4 将手机模拟成 iBeacon	322
18-5 设计 Socket Library	324
18-6 设计 Socket Library (Server)	329
18-7 设计 Socket Library (Client)	334
18-8 Socket 范例程序	336
第 19 章 CloudKit	339
19-1 判断是否登录 iCloud 并取得登录者信息	340
19-2 创建表并读写数据	342
19-3 访问图片或二进制数据	345
19-4 修改与删除数据	348
19-5 关联性设置	350
19-6 订阅与删除异动通知	353

第 20 章 HealthKit.....	356
20-1 读取生日性别与血型.....	357
20-2 写入与读取心跳数据.....	360
20-3 写入与读取睡眠数据.....	364
20-4 查询条件设置.....	366
20-5 列出最大值、最小值或平均值等统计数据.....	367
第 21 章 Extension.....	369
21-1 Today——今天.....	371
21-2 Action——动作.....	374
21-3 Share——共享.....	378
21-4 PhotoEditing——照片编辑.....	380
21-5 Document Provider——文件管理.....	383
21-6 Keyboard——键盘.....	387
第 22 章 其 他.....	391
22-1 从 View 中调用 AppDelegate 方法.....	391
22-2 取得电池状态.....	392
22-3 打开手机背的 LED.....	393
22-4 拨打电话.....	394
22-5 E-mail.....	395
22-6 App 间的数据共享.....	395
22-7 将日期时间格式化输出.....	397
22-8 使用日历类.....	399
22-9 将程序设置放在系统设置区中.....	400
22-10 让系统设置区支持多语言.....	402
22-11 让 App 支持多国/地区语言.....	405
22-12 将屏幕关闭功能取消.....	409
22-13 隐藏画面最上方的状态栏.....	410
22-14 Undo 与 Redo 功能.....	410
22-15 让照片套用滤镜特效.....	411
22-16 随机数.....	412
22-17 编写 delegate.....	413
22-18 method 延迟调用.....	415
附录 A 数据库概述.....	416
附录 B 谓词语法格式.....	423
附录 C 字符串格式表示.....	425

第1章 绪 论

1-1 移动设备里的小确幸

“小确幸”——生活中微小而确实的幸福。仔细想想，虽然生活中有许多事物很小，却可能在某个瞬间让你觉得好幸福。为了留住或分享这份幸福，文字、相机等工具被应用着，甚至为了体验更多的小确幸，有许多工具正不断地被创造出来。

以手持移动设备为例，2007年1月苹果公司的iPhone首次呈现于世人面前，影响了许多人对手机的看法，2010年1月发表的iPad也带动了新一波移动设备的崛起。无论是iPhone、iPad或其他品牌的移动设备，我们都可以看到，在这样小小的移动设备上装满了许多与用户或是外界互动的组件，像是触控屏幕、GPS、麦克风、相机、喇叭、陀螺仪、网络、电话通信等，拥有它的用户，几乎可以走到哪就工作到哪，走到哪乐趣就跟随到哪，在生活中时时刻刻创造或捕捉幸福。以拍照为例，虽然手机拍出来的照片跟专业相机的要求不太一样，但是因为方便性与功能多样性，再加上经过手机将生活中瞬间感受到的幸福与人分享的渠道变得便利了，使得手机应用更多。于是拍张照片，弹指间这张照片就分享出去，似乎已成为现代人生活中留存乃至传达小确幸的重要工具，甚至改变了人们记录生命的方式。此外，突然间想听某首歌，再也不用去买整张专辑，按个按钮，iTunes的音乐库就能协助你让想听的歌在耳边萦绕。这种走到哪里乐趣就跟到哪里的小确幸，因为iPhone和iPad的问世，让许多人觉得幸福并不遥远，传达幸福也变得容易许多。

可是小确幸的关键真的只是这冰冷的四方形设备吗？答案肯定是“不”。事实上，真正赋予iPhone或iPad生命，协助它带给人们这么多乐趣的关键，除了上述许多不同功能的硬件设备外，还有一个不容忽视，甚至可以说是最重要的关键，那就是它背后庞大的应用软件，这些应用软件有个专有名词，称为App，也就是说，若小确幸是个产出，iPad或iPhone是生产小确幸的工厂，那么App就是工厂里除了人与工厂标准之外，一起搭配生产小确幸的重要生产要素。没有这个重要生产要素的投入，iPad或iPhone对用户而言，就只是一个不具个人特色、无法创造幸福的工具而已。目前在苹果公司的App Store上供用户下载的App已经达

到 90 万个，涵盖范围包括商业、娱乐、教育、书籍、生活、旅游、游戏等。当然，最热门的还是游戏类，毕竟人们在舟车劳顿之余想要玩游戏的人要多一些。正因为这些 App 的投入，iPad 或 iPhone 才能创造专属于个人的小幸福。

然而，数量这么多的 App 是哪来的呢？全部是苹果公司提供的吗？当然不是，苹果公司只写了极少部分而已，绝大多数的 App 来自于专业的软件开发公司或是对写程序有兴趣的业余开发者。他们看到人们想创造、体验或传达小确幸的欲望，写出了可以为用户生产小确幸的工具，即各种各样的 App，同时也为自己或公司创造属于自己的小确幸——赚了金钱或是赢得名声。在 App Store 上供人下载的 App 有些是免费的，有些则是付费的，但即使需要付费，价格也不高，最便宜的 0.99 美金，贵一点的 2.99 美金、4.99 美金、14.99 美金或是更高，但一般来说大多在 9.99 美金以内，所以，如果 App 开发者的小确幸是在创造利润，那么根据苹果公司的合约，开发出一个不错的 App 并且标价 0.99 美金，一旦全世界有一万人下载，这个程序就卖了将近一万美金，其中苹果公司获得三成，而开发者可以赚得 7000 美金。

整体来看，App Store 改变了软件设计这一行的生态系统。在过去，想要卖一个程序，可能需要先成立一家公司，然后写一个很大也很复杂的系统，加上大量的广告营销，才可能卖得不错，但是现在，程序员只要专注在某个特定的功能，加上一些创意，就可以在 App Store 上架，无论是职业的或业余的程序员，实现小确幸的道路变得不再遥不可及。例如某个信息相关专业的学生，开发了一个特殊的图像处理算法，现在他可以使用 iPhone 上的相机功能配合这个算法，编写一个拍照的 App 放在 App Store 中，选择以 0.99 美金的价格来贩卖，创造自己的幸福，也为他人带来小确幸。更重要的是，这位学生需要学会很多东西吗？其实不多，调用 iPhone 上的相机拍照，然后将拍好的照片利用他发明的算法进行图像处理，最后保存即可。如果他喜欢，还可以调用网络共享的功能将处理完的照片共享到社区网站。2012 年 4 月 Facebook 花了 10 亿美金买下 Instagram 这个图像处理并结合社区网站的 App，此举让许多人大跌眼镜，其实这个就是每个人都想要追逐的 App 梦，都想要拥有的小确幸或是大确幸，不是吗？

1-2 注册与下载开发工具

在 iPhone、iPod 或是 iPad 上运行的操作系统称为 iOS，想要成为 iOS 的 App 开发者，必须先有一台可以运行 Mac 操作系统的计算机，然后在 Mac 的 App Store 上搜索 Xcode 软件。iOS 的开发工具称之为 Xcode，除了可以在 Mac App Store 上下载外，在 Apple 的开发者网站 <http://developer.apple.com> 也可以找到（抢先版的 Xcode 需要成为付费开发者才能下载）。由于 Xcode 只能在 Mac 操作系统上运行，因此添购一台 Mac 计算机是必要的投资，虽然有些非 Mac 计算机上也可以运行 Mac 操作系统，但可能会有性能上、稳定性上以及硬件支持上的问题，建议还是选择一台标准的 Mac 计算机。内存至少要安装 4GB 以上。如果不想花大钱投资，可以考虑购买 Mac Mini，通过外接屏幕、键盘与鼠标后即可开始享受 Mac 操作系统

并开始编写 App 了。

有了 Mac 计算机之后, 接下来必须在开发者网站注册一个开发者账号, 这个账号可以是原本已有的 Apple ID 或是注册一个新的, 注册是免费的, 所以不用担心 Apple 会跟你要钱。免费开发者可以开发 App, 但是不能在真正的设备(例如 iPhone)上测试或运行, 只能通过模拟器来测试, 而且有些功能在模拟器上也无法使用, 例如 iCloud, 当然免费开发者也不能将自己开发的 App 上架到 App Store 让全世界的人下载。相比较免费开发者而言, 你可以付费成为付费开发者。付费开发者共有两种方案可以选择: 普通开发者或是企业。另外还有一种教育方案, 提供给大专院校师生, 使其可以在实体机器上测试 App, 当然教育方案是免费的。每种方案提供了不同的功能, 下面这张表列出了各个方案之间的差异。

项目	普通开发者	企业	大专院校
提供 iOS Pre-release SDK	✓	✓	N/A
在 iPhone 等实体设备中运行	✓	✓	✓
程序代码技术支持	✓	✓	N/A
Ad Hoc 发布	✓	✓	N/A
In-House 发布	N/A	✓	N/A
个性化 B2B 发布	✓	N/A	N/A
App Store 发布	✓	N/A	N/A
价格	99 (美元/年)	299 (美元/年)	免费
备注	信用卡付费	需要提供 D-U-N-S 编号	只提供给合格且具有学位授予资格的高等教育机构

从上表来看, App 的发布模式总共有 4 种: Ad Hoc、In-House、个性化 B2B 与 App Store, 这些发布模式所代表的意思如下。

- Ad Hoc 发布: 可以将开发好的 App 安装在 100 台实体设备上运行或是测试。
- In-House 发布: 可以将开发好的 App 安装在企业内 500 个员工的实体设备上, 此方案适合企业内部使用。
- 个性化 B2B 发布: 为了某些目的开发给特定公司专用而非一般大众使用的 App。
- App Store 发布: 这个就是大家最常见的, 将 App 放到 App Store 上让全世界的人下载。如果开发者已设置售价, 那么每位下载者就必须先付费才能下载。

在以上这 4 种 App 发布方式中, 最常看到的还是 App Store 发布, 只要每年付出 99 美金成为付费开发者, 就可以让全世界的人下载自己编写的 App。除此之外, 普通开发者方案在付费时可以选择“个人”或是“公司”, 不同之处在于个人的话, 之后只有一个 Apple ID 账号能够进行开发, 如果是公司则不限。公司需要向 Apple 出示 D-U-N-S 编号(邓白氏环球编码), 证明申请者是个合法公司或是企业。

如果申请一切顺利，可以很快地在当初留下的 E-mail 中收到苹果公司寄来的邮件，按照邮件内容去启动付费开发者账号即可完成注册申请的过程。完成注册过程后，接下来就要在 Mac 计算机上创建开发者证书，目的是要让编写好的 App 可以在手机上测试或是放到 App Store 上让人下载。相关的步骤在本书的第 3 章会详细讲解，这里不再赘述。

1-3 使用的语言

编写 iOS App 的程序语言为 Objective-C 或是 Swift（这本书将不会讨论 Swift），如果有面向对象程序语言的基础（例如 C++或是 Java），那么学习 Objective-C 将是一件很容易的事。Objective-C 是 C 语言家族中的一员，因此大部分 C 的语法、数据类型或数据结构在 Objective-C 中都是一样的。市场上针对 Objective-C 的书很多，建议读者可以买一本放在手边参考，这里仅针对 Objective-C 的一些特性与编写 App 时经常用到的部分加以说明。

1. 加载头文件

C 或是 C++加载头文件（.h）时使用的语法是#include，但是在 Objective-C 中建议使用#import，这样的转变是为了不要重复加载相同的头文件。在过去编写.h 文件时最开头总是要加#ifdef、#define 等来确保.h 文件不会被重复加载，现在使用#import 语法后相同的头文件只会被加载一次。

```
#import <UIKit/UIKit.h>
```

2. 声明类

类声明由@interface 开始，以@end 结束。例如，假设 MyClass 为类名称，后面跟着的冒号之后为 MyClass 所继承的类，左右大括号框住的区域为成员变量声明的地方，括号外的部分为 method（方法）声明的位置。

```
@interface MyClass : NSObject
{
    // 声明成员变量的地方
    int count;
    NSString *name;
}
// 声明 method 的地方
-(void)initWithVariable:(int)pScore productName:(NSString *)pName;
+(void)classWithString:pString;

@end
```

上面两个 method 最前方的加号与减号稍后解释，如果来实现这两个 method 的话（放在.m

文件中), 其语法如下:

```
-(void)initWithVariable:(int)pScore productName:(NSString *)pName
{
    // 程序写在这里
}

+(void)classWithString:(id)pString
{
    // 程序写在这里
}
```

3. 调用 method

Objective-C method 的声明语法比较特殊, 跟其他语言有很大的不同, 以下面这个 method 为例:

```
-(void)initWithVariable:(int)pScore productName:(NSString *)pName;
```

第一个字母为减号或是加号, 减号开头的 method 为 instance method, 代表需要先产生类的实体后才能调用 (也就是 new 一个 instance); 加号开头的 method 为 class method, 不需要先产生类的实体就可以直接调用。“void”代表此 method 没有返回值, 因此 method 结束前不需要 return value。Objective-C method 的名称也很特殊, 上述 method 的名称为 “initWithVariable:productName:” 而不是 initWithVariables, 名称中的冒号代表后面接有参数, 所以 initWithVariable 后面接了参数, productName 后面也接了参数。

除了 method 的名称比较特殊之外, 调用 method 的方式也很特别, 调用时使用中括号, 如下所示:

```
[self initWithVariable:10 productName:@"Apple"];
```

在 self 处放置的是类的 instance 变量, 如果是 class method 的话, 这个位置就放类名称, 空一格后再接 method 名称。特别需要说明的是, self 在这里是 Objective-C 的保留字, 指的是自己这个类, 类似于 C++ 中的 this 保留字。使用 self 的意思是假设在 MyClass 这个类中的另一个 method 要调用 initWithVariable:productName: 时, 就会需要使用到 self 了。

类的 instance 变量声明方式一般来说就是两个步骤: 配置内存与初始化。配置内存使用的是 alloc, 初始化使用的是 init, 因此在很多地方都可以看到如下的程序代码:

```
MyClass *class = [[MyClass alloc] init];
```

除了这种写法之外, 新的写法如下:

```
MyClass *class = [MyClass new];
```

4. block 代码块

block 代码块是将程序代码用一段特殊语法封装起来。block 虽然不是 function，但是看起来很像 function，因此对不熟悉的读者而言，可暂时把它当成 function 来看，会比较容易理解。在正式说明 block 之前，先看一个简单的 Objective-C function（或方法），这个 function 的目的就是在调用时传入一个字符串，然后将此字符串输出。

```
-(void)print:(NSString *)s
{
    NSLog(@"%@", s);
}

-(void)viewDidLoad
{
    [self print:@"hello"];
}
```

把 print 函数转换成 block，注意因为 block 不是 function，正确来说它算是“一行”程序代码，因此在开头的地方不需要加“减号”，并且在结尾的地方要加上分号作为结束符。block 代码块既可以写在真正的 function 外面，也可以写在 function 里面，甚至可以写在某一行程序代码中间。

```
void(^print1)(NSString *) = ^(NSString *s) {
    NSLog(@"%@", s);
};

-(void)viewDidLoad
{
    print1(@"hello");
}
```

除了开始定义 block 代码块名称与参数地方的写法比较奇特之外，其余部分其实跟 function 并没有什么差别。以上述 print1 这个 block 为例，开始的 void 表示该 block 没有返回值，(^print1) 则为这个 block 命名，名称为 print1。(NSString *)表示 print1 有一个参数，其类型为 NSString*。等号右边^(NSString *s)是因为等号左边有(NSString *)，因此要再说明一次参数的类型与参数名称，之后以左右大括号框住程序代码块，最后记得以分号结尾。

如果是没有参数传递的 block 写法，原本在等号右边应该写成^(void)，但也可以省略成仅输入“^”符号即可。

```
void(^showSomething)(void) = ^{
    // 程序代码写在此
};
```

声明在 block 外面的变量，对 block 而言相当于全局变量（global variable），虽然 block 内部可以直接使用，但是仅仅只能读取，不能修改，如果尝试修改会得到一个编译错误。因此，如果一定要在 block 内部修改声明在外面的变量，则需要在该变量声明语法前加上__block 修饰符（“__”为连续两个下划线）。