

国内最大的Apple开发者社区CocoaChina联合策划

资深iOS应用开发专家 (现就职于美国苹果公司) 亲自撰写, 权威性毋庸置疑
资深iOS/Android应用开发专家、ChaCha API金奖得主Lordhong和资深iOS
应用开发专家沈卓立联袂推荐



李晨 著

iPad in Action



机械工业出版社
China Machine Press



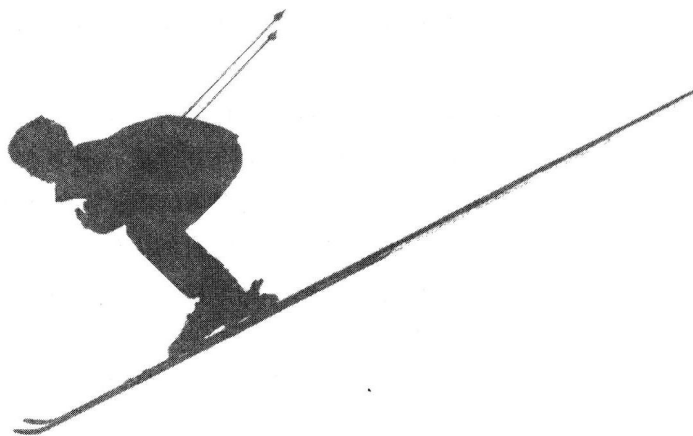
iPad

应用开发实战

李晨 著



机械工业出版社
China Machine Press



前言

欢迎加入iPad应用开发者的阵营。

如果你已经是一名iPhone应用或者Mac应用的开发者，希望转战iPad应用开发，那么本书能帮助你最快地掌握iPad特有的设计思想和API。如果你没有任何iOS或者Mac OS开发经验，希望从零开始学习iPad应用开发，那么本书对Objective-C语言的介绍以及大量的开发实例将帮助你从理论和实践两个方面迅速入门。

2007年，苹果公司发布了第一代iPhone，立即引发了智能手机产业的革命。

2008年，苹果公司发布了iPhone SDK，允许开发者为iPhone开发应用，并通过App Store进行推广和销售，这引发了移动软件开发的狂潮。

2009年，这是百花齐放的一年。大量的个人开发者、小公司、大公司在开发、销售等领域各显神通，发布了大量的iPhone应用，很多开发者也从中收获颇丰。

2010年，苹果公司发布了iPad，又一次引发了业界的革命——这一次是在平板电脑领域。iPad不仅为市场注入了新的活力，也为开发者带来了新的梦想和新的机遇。

本书就是为这些胸怀梦想、正在寻找机遇的人而写的。它从市场环境、应用内容、软件构架等多方面介绍了iPad的开发思想，从技术的角度介绍了iPad的开发细节，并且通过大量实例诠释如何将iPad应用从构想变成现实，从计划变成收入。

本书内容特色

- 本书中介绍的所有应用均使用Objective-C语言开发实现。Objective-C是iPad、iPhone、Mac应用开发中的主要语言。

- ❑ 从语法特性、常用类、常用方法等多角度介绍Objective-C语言。
- ❑ 注重iPad特性，强调其与iPhone的不同之处，这样帮助读者从应用设计上更好地贴近iPad。
- ❑ 大量的实例，涉及图像处理、娱乐、游戏、文字处理等多个领域。有简单的例子，用于解释简单概念，适合初学者；有复杂的例子，覆盖多个章节，适合进阶读者。
- ❑ 介绍了最为实用的控件和API，例如列表视图、分割视图、弹出视图、加速度传感器等，帮助读者掌握在短时间内发布应用所需要的基本技能。
- ❑ 开发环境的介绍，帮助读者熟练掌握XCode的开发、调试、快捷键功能，使开发效率成倍提高。
- ❑ 关键词中英文对照，让读者能够用母语来理解关键词和概念，也方便读者查询英文资料。
- ❑ 对iOS系统和应用生命周期的概述，让开发者能够对应应用所运行的环境以及所经历的各个阶段有更深入的了解，从而在应用设计上能够有深思熟虑的计划。
- ❑ 强调移动开发和桌面开发的不同之处，尤其是在内存管理和异步方法上，从而帮助读者开发出更适合移动设备的应用，而不是简单的桌面软件的“山寨版”。
- ❑ 对App Store市场历史以及市场环境的介绍，让开发者在设计应用时能更加贴近市场，贴近用户。
- ❑ 介绍了很多实用的营销技巧，帮助开发者更好地宣传和销售自己的应用，获得更多的收益。

适合的读者与阅读方法

- ❑ 有iOS或Mac OS编程经验的开发者：应当阅读本书的每一章，但可以跳过一些Objective-C的语法讲解部分，而专注于学习iPad特有的设计思想和API。同时可以利用大量的实例，迅速进入iPad开发的最佳状态。
- ❑ 有C++、Java、QT等面向对象编程经验的开发者：应当阅读本书的每一章，并特别注意Objective-C语法。Objective-C的语法有很多独特之处，内存管理是其中的难点和重点，需要多花时间。另外，移动应用的架构和桌面软件也很不相同，需要注意相关知识。
- ❑ 没有任何编程经验的初学者：应当仔细阅读本书每一章，从简单的例子入手，在例子的基础上进行改动和试验，了解每一处代码的用途；然后开始“临摹”创作，写一个小小的应用，掌握应用的基本架构；之后再继续学习语法、API，循序渐进，掌握iPad开发的基本技能。
- ❑ 软件策划与营销人员：可以蜻蜓点水地浏览第2~17章，了解iPad应用可能具有的功

能以及硬件性能；然后多多试玩App Store上的优秀应用，学习优秀和新颖的观点；最后应当阅读第1章和第18章，从中了解App Store的历史与现状，以及一些营销策略。

- ❑ 创业者：作为一名创业者，需要有较为全面的技能，但也许没有太多时间详细学习每一章。因此，可以略读各个章节，从中得到创意和技术上的启发。另外，应当熟悉每一个应用实例的构架和功能。

源代码下载

为了便于大家实践，本书提供了源代码供大家下载，下面两种下载途径均有效：

- ❑ 通过华章公司的官方网站（www.hzbook.com）下载，在网站上找到该书的页面即可。
- ❑ 通过我的博客（www.lichen1985.com/iPadCode.html）下载。

联系作者

如果你发现了本书的任何问题或有任何建议给我们，欢迎通过下面的邮件地址与我联系：
iPadAppPracticalProgramming@gmail.com

致谢

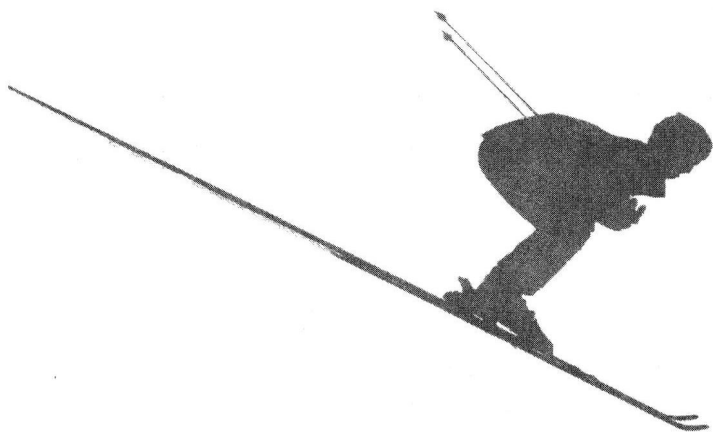
本书的编写，经历了撰写、审核、修改、复审、再修改、编辑等很多很多轮的改进。这本书的出版是很多人共同努力的结果。

我首先要感谢华章公司，是公司里每一位老师、编辑的辛勤劳动，使得稿件的质量不断提升，并最终得以出版。其中尤其要感谢杨福川编辑，他一直在坚定地支持着我写作以及相关的一切工作。

我要感谢Cocoa China论坛、密歇根理工大学iPhone开发俱乐部、苹果开发论坛（Apple Developer Forum）以及其他很多iOS开发社区。和众多开发者共同学习、讨论使我受益匪浅，并且在iOS开发的道路上走得越来越远。

我要感谢我的家人和朋友，他们在我写作最为艰苦的时候给我以鼓励，让我坚持下来，并将书稿质量不断提高。

最后，我要感谢苹果公司！这家传奇的公司在2010年发布的iPad为世界带来了全新的移动体验，也为开发者们带来了新的梦想。我还要感谢众多的用户与批评者，他们购买并使用我的应用，不断地给我提出批评和建议，使得我的应用越做越好。



目 录

前言

第1章 iPad开发入门 / 1

- 1.1 iPad概览 / 2
- 1.2 从PC到iPad / 3
- 1.3 从iPhone到iPad / 4
- 1.4 iPad开发的必备知识 / 5
- 1.5 iPad开发的必备工具 / 5
- 1.6 第一个iPad应用——Hello World / 6
 - 1.6.1 建立工程：HelloWorldMail / 7
 - 1.6.2 工程概览 / 8
 - 1.6.3 添加按钮 / 15
 - 1.6.4 邮件编辑器 / 17
- 1.7 小结 / 19

第2章 视图和视图控制器 / 20

- 2.1 视图 (UIView) / 21
 - 2.1.1 视图概览 / 21
 - 2.1.2 窗口 (UIWindow) / 22
 - 2.1.3 网页视图 (UIWebView) / 23
- 2.2 视图控制器 (UIViewController) / 23

- 2.2.1 视图控制器概览 / 23
- 2.2.2 导航控制器 (UINavigationController) / 24
- 2.2.3 标签控制器 (UITabBarController) / 25
- 2.3 弹出控制器 (UIPopoverController) / 26
- 2.4 分割视图控制器 (UISplitViewController) / 27
- 2.5 实例: iPad网页导航 / 27
 - 2.5.1 基于分割视图控制器的应用 / 27
 - 2.5.2 网页视图 / 29
 - 2.5.3 表格视图 / 31
- 2.6 小结 / 34

第3章 内存管理 / 35

- 3.1 对象所有权、引用计数与释放 / 36
- 3.2 自动释放与便捷方法 / 38
- 3.3 访问器方法与属性 / 39
- 3.4 改变引用计数的特殊情况 / 40
- 3.5 内存管理总结 / 41

第4章 键盘输入和自定义键盘 / 42

- 4.1 文本输入与系统键盘 / 43
- 4.2 自定义键盘之一: 替代系统键盘 / 47
- 4.3 自定义键盘之二: 与系统键盘共存 / 50
- 4.4 自定义键盘之三: 修改系统键盘 / 51
- 4.5 小结 / 55

第5章 绘图 / 56

- 5.1 iOS概览 / 57
- 5.2 iOS中的绘图技术 / 58
- 5.3 Quartz 2D / 59
- 5.4 贝济埃路径的使用 / 61
- 5.5 画图实战: 绘图板 / 62
- 5.6 小结 / 74

第6章 容器类的使用 / 76

- 6.1 数组 (NSArray) 与可变数组 (NSMutableArray) / 77
 - 6.1.1 数组的创建与初始化 / 78
 - 6.1.2 查询数组 / 78
 - 6.1.3 产生新数组 / 79

- 6.1.4 比较数组 / 79
- 6.1.5 保存数组 / 79
- 6.1.6 添加对象 (NSMutableArray方法) / 79
- 6.1.7 移除对象 (NSMutableArray方法) / 79
- 6.1.8 替换对象 / 80
- 6.2 字典 (NSDictionary) 与可变字典 (NSMutableDictionary) / 80
 - 6.2.1 创建字典 / 80
 - 6.2.2 获取关键字与数值 / 81
 - 6.2.3 保存字典 / 81
 - 6.2.4 添加条目 (NSMutableDictionary方法) / 81
 - 6.2.5 移除条目 (NSMutableDictionary方法) / 81
- 6.3 集合 (NSSet) 与可变集合 (NSMutableSet) / 81
 - 6.3.1 创建集合 / 82
 - 6.3.2 读取集合 / 82
 - 6.3.3 比较集合 / 82
 - 6.3.4 添加或移除对象 (NSMutableSet方法) / 82
 - 6.3.5 集合操作 (NSMutableSet方法) / 83
- 6.4 容器在拼图游戏中的使用 / 83
 - 6.4.1 拼图游戏的设计 / 83
 - 6.4.2 相关定义 / 84
 - 6.4.3 Piece类与字典 / 87
 - 6.4.4 Puzzle类与字典和数组 / 89
- 6.5 小结 / 93

第7章 文件系统 / 94

- 7.1 iOS文件系统概述 / 95
- 7.2 通过模拟器看应用的包裹 / 96
- 7.3 文件管理器 / 97
- 7.4 NSString的路径功能 / 98
- 7.5 拼图游戏中的文件管理 / 99
- 7.6 小结 / 107

第8章 绘图进阶 / 108

- 8.1 屏幕旋转 / 109
- 8.2 绘制曲线和切割图案 / 111
- 8.3 高级绘图在拼图游戏中的应用 / 114

8.4 小结 / 122

第9章 触摸事件与手势识别 / 123

9.1 触摸事件、视图结构与响应顺序 / 124

9.2 官方SDK中的手势识别器 / 126

9.3 Photo Jigsaw中对触摸的响应 / 128

9.4 自定义的手势识别器 / 135

9.5 小结 / 136

第10章 同时性编程与定时器 / 137

10.1 同时性编程概述 / 138

10.2 操作队列 / 140

10.3 拼图游戏中的同时性编程 / 142

10.4 定时器及其在拼图游戏中的使用 / 146

10.5 小结 / 148

第11章 动画 / 150

11.1 iOS动画技术概述 / 151

11.2 用图像视图来逐帧显示动画 / 153

11.3 视图的动画技术 / 154

11.4 CogRadio应用 / 156

11.5 CogRadio应用中的动画 / 160

11.6 小结 / 166

第12章 多媒体播放 / 167

12.1 音频播放 / 168

12.1.1 使用系统声音服务来播放音效 / 168

12.1.2 音频播放器 / 169

12.1.3 音乐播放器 / 169

12.2 视频播放 / 171

12.2.1 电影播放器 / 171

12.2.2 在网页视图中嵌入播放 / 174

12.3 CogRadio中的多媒体 / 175

12.4 小结 / 178

第13章 cocos2d开发入门 / 179

13.1 cocos2d简介 / 181

13.1.1 导演 / 181

13.1.2 场景和层 / 182

- 13.1.3 精灵与动作 / 183
- 13.2 创建cocos2d工程并添加场景 / 184
- 13.3 添加按钮、精灵和动作 / 185
- 13.4 粒子系统 / 187
- 13.5 小结 / 193
- 第14章 连接设备 / 195**
 - 14.1 游戏工具箱简介 / 196
 - 14.1.1 游戏中心 / 196
 - 14.1.2 点对点连接 / 198
 - 14.1.3 语音聊天 / 198
 - 14.2 实现点对点连接 / 199
 - 14.3 OpenFeint 游戏平台 / 203
 - 14.4 小结 / 209
- 第15章 加速度传感器 / 210**
 - 15.1 加速度传感器概述 / 211
 - 15.2 飞机躲子弹游戏 / 212
 - 15.3 倾斜设备来控制飞机 / 216
 - 15.4 分离重力成分 / 219
 - 15.5 小结 / 221
- 第16章 地图与连接服务器 / 222**
 - 16.1 地图编程概述 / 223
 - 16.2 实战：地图编程 / 224
 - 16.3 与服务器交互 / 227
 - 16.4 实战：与服务器交互 / 229
 - 16.5 小结 / 235
- 第17章 Objective-C与XCode进阶 / 236**
 - 17.1 NSLog扩展 / 237
 - 17.2 范畴的使用 / 239
 - 17.3 读写器的内存管理 / 240
 - 17.4 KVO / 242
 - 17.5 单例 / 243
 - 17.6 运行时的对象查询 / 245
 - 17.7 调试器的巧用 / 246
 - 17.8 XCode快捷键 / 247

17.9 小结 / 249

第18章 iPad应用的开发证书与营销 / 250

18.1 开发证书的获取 / 251

18.2 联机调试 / 254

18.3 应用名字、说明以及图片 / 258

18.4 应用的提交 / 261

18.5 应用的推广 / 262

18.6 应用升级与销售技巧 / 263

18.7 iPad开发的长期策略 / 265

18.8 小结 / 266



第1章 iPad开发入门

1.1 iPad概览

1.2 从PC到iPad

1.3 从iPhone到iPad

1.4 iPad开发的必备知识

1.5 iPad开发的必备工具

1.6 第一个iPad应用——Hello World

1.7 小结

无论你是一个熟练的iPhone开发者，还是一个从未接触过iOS (iPhone Operation System) 开发的程序员，无论你是把iPad开发当成一种职业，还是把它作为业余淘金的手段，在你准备开始写代码之前，你首先要做的就是拿起这款设备，下载几款最畅销的iPad应用，感受它们与设备的完美结合以及与用户的良好交互——是的，你的目标就是要开发出那样的应用。这本书所要做的，就是为你提供达成这个目标所需的全部基础知识。这一章首先帮助你熟悉iPad。

首先需要了解iPad和PC的关键区别。iPad是一款移动设备，它们有着与PC不同的硬件，基于不同的操作系统，运行着不同的软件，用户交互方式也有着天壤之别。这是设计iPad应用的基础。

那么iPad和iPhone相比有何差别呢？它们同为移动设备，同样运行着iOS，iPad的高分辨率和大屏幕就成了两者的决定性差异。如果说iPhone为你提供了一个可以实现梦想的世界，那么在iPad上，你应当寻找更大的梦想。

对iPad有了基本的认识以后，本章将简单介绍iPad开发的必备知识和工具，这些是赢得iPad用户的武器。有了这些武器之后，你需要一个漂亮的登场秀：遵照软件教程的传统，你将实现一个在iPad上运行的Hello World程序。这个程序能够让你通过电子邮件对朋友们说“Hello World!”。

注意 在这一章中，我们专注的是iPad本身，而非代码——因为做开发不仅仅是写代码。写代码时，需要的是理解编程语言并熟悉API，然后实现别人的软件设计；而作为一名开发者，编程技能只是基础，更重要的是理解用户需求，了解设备潜能，然后设计并实现出杀手级的应用。同样地，在本书中，我们将始终强调对iPad元素的理解，然后用代码来帮助理解和实现这些元素。

下面，可以开始你的iPad之旅了！

1.1 iPad概览

关于iPad，有一些参数是首先需要了解的，包括它的重量、大小、屏幕尺寸、容量、操作系统特性、通信接口、传感器等。有了这些知识，才能知道这款设备的潜能，以及设计的应用所能包含的功能。下面所介绍的参数^①以第一代iPad的Wi-Fi型号为例。

iPad高为24.28厘米，宽为18.97厘米，比报亭陈列的A4尺寸杂志略小，厚为1.34厘米，与一般杂志厚度相似。iPad的重量为0.68千克，约等于5个iPhone的重量。这样的尺寸和重量，

① 所有iPad参数均来自于Apple官方网站：<http://www.apple.com/ipad/specs/>。

使得iPad适合于双手持握或者平放于膝盖上操作，易于旋转方向。但由于它不适合于长时间单手持握，因此，在设计应用时，不应加入甩动之类让用户吃力的操作。同时，iPad适合于放在公文包中携带，而不是像iPhone那样能够放在口袋里（除非口袋足够大）带到野炊场所的。因此，当考虑iPad用户需求时应当想到，很多iPad用户是携带公文包的白领或中产阶级，而非每天在街边嬉戏的年轻学生。

iPad有着9.7英寸的电容式多点触摸屏，屏幕分辨率为 1024×768 像素，像素大小为132DPI (Dots Per Inch)。相对于iPhone的 480×320 像素的屏幕分辨率，iPad屏幕可以显示更多的信息和内容。比如一款含有地图的游戏，在iPhone上，用户需要不停卷动屏幕来浏览地图，而在iPad上，用户的地图翻动操作将会少得多，因此这类游戏更适合于在iPad上开发。iPad容量为16GB，但这将不会成为你的限制。它有着16GHz的苹果A4处理器，能够保证大多数应用流畅运行。但是它的DRAM (Dynamic Random-Access Memory，动态随机存取存储器) 容量仅为256MB，因此在开发中需要执行严格的内存管理。iPad下载应用的方式主要为Wi-Fi和3G，而不便于下载大容量的应用，所以应当控制所开发的应用的大小，最好不要超过100MB，否则用户会不乐意去下载你的应用。

当前，iPad运行的操作系统为iOS 3.2.1。这个版本的操作系统不支持多任务功能，因此你的应用应当能够随时保存状态，防止用户突然退出应用。即便未来iPad所运行的操作系统支持多任务功能了，仍然需要加入这样的考虑，以确保信息安全。

传感器方面，iPad上有加速度传感器和环境光传感器，其中加速度传感器是这款设备的核心设计之一，它使得在iPad上的任何一款应用都能够支持四个设备方向，也使得一些需要感应设备倾斜的应用成为可能。在设计iPad应用时，还需要考虑到对不同设备方向的支持，这也是苹果官方对所有iPad应用的建议。iPad上可供使用的输入方式有触摸、软件键盘、声音、加速度传感器，输出方式则包括屏幕和声音。这些输入/输出方式应当在应用开发当中多加使用，以丰富用户使用乐趣。

1.2 从PC到iPad

如果要从PC (Mac) 平台转向iPad开发，那么需要了解PC开发与iPad开发的几点区别。

- iPad的交互方式以触摸为主，而其他类电脑的交互方式以鼠标和键盘为主。iPad适用于查看E-mail、上网冲浪、做简单的笔记。完成这些工作不需要大量的键盘输入，人们又希望能够随时随地做这些工作，iPad的携带比任何电脑和笔记本方便很多，所以iPad非常实用。相反地，iPad不适合于文本输入工作，因为人们在软件键盘上的输入速度会比在硬件键盘上慢很多。

- ❑ iPad的用户会总是期望无论设备处于何种方向，他们都能够使用各种应用程序。他们 会从各种设备方向打开应用，然后将设备在手中反复翻转，期望应用能做出响应。相反地，用户不会去尝试翻转他们的台式机或者笔记本来让应用做出响应。
- ❑ 还需要考虑到iPad有限的计算和存储能力。iPad上的应用不应当过于消耗计算能力，那样用户会体验到很“卡”的感觉，会给你的应用评“一星”（很低）！在iPad上，应当执行更为严格的内存控制，保证不用的对象得到及时释放，否则应用会因为内存耗尽而崩溃。
- ❑ iPad和其他类电脑也有交互。iPad玩家需要用其他类电脑上的iTunes软件来向iPad同步歌曲、电影、书籍，乃至其他更多应用。有一些应用本身也通过Wi-Fi将iPad和其他类电脑连接起来，实现更多的功能。比如iPad应用Brushes，可以在iPad上面用这款应用绘制图形，然后将文件通过Wi-Fi传至其他类电脑上，再通过电脑上的应用程序来重现在iPad上一步一步完成作品的过程！

1.3 从iPhone到iPad

很多iPad开发者可能来自iPhone开发阵营。相对于其他类电脑，iPad和iPhone的差异较为微妙，如若不能理解这些微妙的差异，开发出的应用将无法吸引感兴趣的用户。

先来看看两者的相似之处：它们都有着电容式触摸屏，同样运行着iOS，但在版本和功能上会略有不同；它们都通过同一个App Store来下载应用，iPad甚至可以兼容绝大多数的iPhone应用。那么，为什么不能简单地改动一下iPhone应用使之成为iPad应用呢？为什么还要来专门研究iPad开发呢？

答案在于那块大了一倍以上的屏幕，它多了三倍以上像素数量。如果只是将iPhone应用中元素的尺寸简单扩大一倍，那么无异于浪费了iPad的高分辨率，有些元素也会显得很怪异。如果不放大iPhone应用中的元素，而是将它们重新组合在iPad屏幕上，那么你会发现，屏幕上多出了很多空白，空间被浪费了。面对iPad的高分辨率屏幕，每一个iPhone开发者都应当静下心来思考如何重新设计界面元素，重新定义和用户的交互方式，花时间重塑应用，而不要寻找捷径来牺牲用户体验。这也是苹果官方为iPad设计UISplitViewController和UIPopoverViewController[Ⓔ]的核心思想，我们会在后面的章节中讲到。

Ⓔ 控件的细节参见iOS Reference Library: <http://developer.apple.com/library/ios/#navigation/>。

1.4 iPad开发的必备知识

如果你是一名iPhone开发者，那么相信已经掌握了iPad开发的基础知识和工具，可以直接跳至第2章，去学习视图和视图控件的使用。对于初次接触iOS开发的读者，可以从这里开始搜集所需的资料 and 工具。

作为一名iPad开发者，首先需要掌握Objective-C语言^①。Objective-C是一门面向对象的语言，可以看做是C或者C++的超集。Objective-C编译器不仅能编译Objective-C源码，也可以编译C或者C++源码。一个纯Objective-C的源文件的后缀名应当为.m，而一个包含C++代码的源文件的后缀名应当为.mm。本书将在后面的实例中帮助读者逐步熟悉Objective-C的具体语法。

仅仅掌握Objective-C是不够的，因为如果要想从零开始，一行一行写代码来完成一个应用，将需要很长的时间，这样不仅效率低下，也没有必要。苹果已经为开发者提供了一个很高的起点，只需站在这个起点之上，就能事半功倍。这个起点就是Cocoa^②——iOS应用程序的框架。事实上，它也是Mac OS X应用程序的框架。Cocoa在iOS操作系统中运行在应用框架（Application Frameworks）层和核心框架（Core Frameworks）层之中。它为开发者提供了基本的应用程序框架、用户界面元素、绘画和图像显示功能、系统交互功能、多线程和内存管理功能、软件国际化功能、文本、网络等。熟悉Cocoa的这些知识，便能调用这些API来快速、高质量地完成你的应用。

如果是从零开始学Objective-C和Cocoa，本书当中的例子将是最好的参考。书中会对每一个应用的结构做出分析，也会对每一个新出现的语法做出详细的解释。

1.5 iPad开发的必备工具

要掌握的开发工具主要包括：一个集成开发环境——XCode IDE，一个iPad/iPhone界面创建软件——Interface Builder（书中简称为IB），以及一个性能分析仪器——Instruments。这三件工具也统称为XCode。这些工具可以在苹果官方的iPhone开发中心（iPhone Dev Center）的网站上获取，网址是<http://developer.apple.com/iphone/index.action>（注意，苹果可能会更新这个网站的网址，因此如果上述网址失效，请在Google上搜索iPhone Dev Center。

① 参见Introduction to The Objective-C Programming Language: <http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html>。

② 参见Cocoa Fundamentals Guide: <http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/CocoaFundamentals/Introduction/Introduction.html>。

本书中提到的其他网址也可能会失效，请使用Google搜索找到最新网址)。你需要注册一个Apple ID，用这个ID登录，然后在下载区域下载iOS SDK。iOS SDK里面包含了你所需要的全部工具。

以上只是做iPad开发所必需的工具。如果想下载最新的iOS SDK，并且希望将所开发出来的应用放在App Store上销售，就需要在上述网站中申请加入iPhone开发者项目（iPhone Developer Program，iDP）。申请获得通过后，将需要支付一定费用来获取为期一年的开发资质（个人为一年99美元，企业为一年299美元）。关于iDP申请和之后的操作，将在本书的最后一章专门叙述。

XCode IDE是苹果公司开发的一个集成开发环境，主要用于iPad、iPhone和Mac开发。它提供了项目模板，你可以直接从模板开始创建iPad应用。它提供了一个方便的界面让你设置编译目标，还提供了一键编译执行功能。它提供了一个模拟器（Simulator），这有两个好处：如果你没有iPad，依然可以在模拟器中来查看你的应用运行情况；你有iPad，通过模拟器来观察代码运行情况会更快捷。但是，模拟器也缺少真机的很多功能，比如它无法模拟重力传感器，不能连接App Store，而且模拟器的运算能力会比iPad强大，因此一些潜在问题可能反映不出来。XCode IDE还提供了良好的调试环境，让你能方便地查找代码中的漏洞。

Interface Builder能够帮助你快速并且直观地创建用户界面。它有着一个包含了大量界面元素的库，可以直接从库中拖拽界面元素到iPad界面，进行可视化编程。对于初学者来说，IB和XCode IDE的结合会令人头疼，但是一旦跟着本书完成了一两个应用之后，将会很快掌握IB用法。

Instruments可以帮助了解应用运行过程中的内存使用状况，也可以查出内存的泄露情况。前面强调过，内存管理在iPad开发中至关重要，内存泄露小则导致程序运行效率降低（对于游戏来说，是FPS（Frame per Second，每秒帧率）的降低），大则引起程序崩溃。所以，在代码中需要执行内存管理的思想，在写完代码之后，还要通过Instruments来查漏补缺，确保iPad应用能高效运行。

1.6 第一个iPad应用——Hello World

下面，将开始实现第一个iPad应用——Hello World。这个应用的主界面非常简单，只有一个按钮，上面显示“Hello World Mail”。点击这个按钮后，将弹出一个邮件发送界面，界面上预先写好发送者的邮件地址、邮件主题和邮件正文。通过这个例子，将学习到如何新建iPad应用，如何通过IB来添加界面元素，如何与用户交互，以及如何调用邮件发送界面。