

第 1 章

欢迎来到iPhone世界



你想编写iPhone、iPod touch和iPad应用程序？哦，这事也怪不得你。事实上，所有这些设备最核心的软件——iOS恐怕要算是长久以来最吸引人的新平台了。毫无疑问，它是迄今为止最有趣的移动平台，特别是现在，苹果公司还提供了一组好用的、具有良好文档的工具来支持iOS应用程序的开发。随着iOS软件开发工具包（SDK）第4版的发布，这一切变得更加美好。

1.1 关于本书

本书将带你走上创建iOS应用程序的大道。我们的目标是让你通过初步学习，理解iOS应用程序的运行和构建方式。在学习过程中，你将创建一系列小型应用程序，每个应用程序都会突出某些iOS特性，展示如何控制这些特性或与其交互。如果你扎实地掌握了本书中的基本知识，充分发挥自己的创造力，并且坚定不移，同时借助苹果公司大量翔实的文档，你就具备了创建专业级iPhone和iPad应用程序所需的一切条件。

说明 Dave、Jack和Jeff为本书创办了一个论坛。这个论坛是志趣相投的人们互相交流的理想地点，可以在这里获取问题的答案，也可以回答别人的问题。论坛地址为<http://iphonedevbook.com/forum>。一定要访问此论坛哦！

1.2 必要条件

在开始编写iOS软件之前，需要做一些准备工作。初学者需要一台运行Snow Leopard（OS X 10.6.5或更高版本）的基于Intel的Macintosh计算机。任何最近上市的基于Intel的Macintosh计算机（不管是笔记本还是台式机）应该都符合要求。

你还需要注册成为iOS开发人员。只有完成这一步，苹果公司才允许下载iOS SDK。

2 第1章 欢迎来到 iPhone 的世界

注册请访问 <http://developer.apple.com/ios/>，这会打开如图1-1所示的页面。

首先点击 **Log in** 按钮，页面将提示你输入 Apple ID。如果你还没有 Apple ID，请单击 **Create Apple ID** 按钮创建一个，然后再登录。登录之后，将进入 iOS 开发主页面。其中不仅有 SDK 的下载链接，还有各类文档、视频和示例代码等的链接，所有这些资源都对开发 iOS 应用程序有帮助。



图1-1 苹果公司的iOS开发中心网站

这个页面上一个最重要的元素是Xcode，它是苹果公司的IDE（集成开发环境），必须下载。Xcode提供了各种实用工具，用于创建和调试源代码、编译应用程序以及调优应用程序性能。学习完本书，你将会迷上Xcode！

示例所用的SDK版本和源代码

随着SDK和Xcode版本的不断发展，下载它们的机制也将会改变。有时SDK和Xcode需要分开下载，有时它们可以合在一起下载。基本出发点是：你应该下载最新发布SDK和Xcode版本（不是beta版）。

本书是针对最新的SDK版本编写的。在某些地方，我们选择使用第4版中引入的新函数或方法，它们可能与早期的SDK版本不兼容，出现这些情况时我们一定会指出。

请从本书的网站<http://iphonedevbook.com>下载最新和最优秀的源代码。

在发布新SDK版本时，我们将相应地更新代码，所以一定要定期查看该网站。

1.2.1 开发者的选择

这个可免费下载的SDK还包含一个模拟器，它支持在Mac上创建和运行iPhone与iPad程序。这对于学习编写iOS程序极其有用。但是，模拟器不支持依赖于硬件的某些特性，如iPhone的加速计或摄像功能。同时这个免费选项还不支持将应用程序下载到实际的iPhone或其他设备中。此外，它也不支持在苹果公司的App Store上分发应用程序。要实现这些功能，需要注册使用另外两个下载选项中的一个，它们不是免费的。

- 标准方案的价格为99美元/年，它提供了大量开发工具、资源和技术支持，支持通过苹果公司的App Store分发应用程序。最重要的是，标准方案支持在iOS上（而不只是在模拟器上）测试和调试代码。
- 企业方案的价格为299美元/年，面向开发专用的、内部的iOS应用程序的企业，以及为苹果公司的App Store开发应用程序且拥有参与该项目的多名开发人员的企业。

有关这两种方案的详细信息，请访问<http://developer.apple.com/programs/ios>及<http://developer.apple.com/programs/ios/enterprise>。

4 第1章 欢迎来到iPhone的世界

由于iOS是一种始终连网的移动设备，并且使用的是其他公司的无线基础设施，因此苹果公司对iOS开发人员的限制比对Mac开发人员多得多（Mac开发人员无需经过苹果公司的审查或批准就能够编写和分发程序）。尽管iPod touch和只能使用Wi-Fi的iPad不允许使用其他的基础设施，它们仍然会面临同样的限制。

苹果公司添加这些限制，主要是为了尽量避免分发恶意或效率低下的程序，因为这类程序可能降低共享网络的性能。开发iOS应用程序似乎麻烦不少，但苹果公司在简化开发过程方面付出了巨大努力。还应该提及的是，99美元的价格比微软公司的软件开发IDE——Visual Studio的价格低得多。

另外，很明显，你还需要一部iPhone手机、iPod touch或iPad。虽然大部分代码都可以通过iOS模拟器进行测试，但并非所有程序都是如此。模拟器上运行的一些应用程序需要在实际的设备上上全面测试，然后才能分发给公众。

说明 如果已决定注册标准版或企业版方案，应该立即注册。批准过程可能需要一些时间，并

且批准之后才能在实际的设备上运行应用程序。但是不必担心，前几章中的所有项目以

及本书中的大多数应用程序，都可以在iOS模拟器上运行。

1.2.2 必备知识

学习本书应该具备一定的编程知识。你应该理解面向对象编程的基础知识，例如，了解对象、循环和变量的含义，还应该熟悉Objective-C编程语言。SDK中的Cocoa Touch是本书使用的主要工具，它使用的是Objective-C 2.0，但是如果不了解Objective-C的新增特性也没有关系。我们将重点介绍要使用的Objective-C 2.0语言特性，并解释其工作原理和使用它的原因。

作为用户，你还应该熟悉iOS本身。就像在任何其他平台中编写应用程序一样，你需要熟悉iPhone、iPad或iPod touch的各种特性，并了解iOS界面以及iPhone和iPad程序的外观。

Objective-C的学习资源

如果你从未使用Objective-C编写过程序，那么以下资源有助于你了解该语言。

首先，阅读Mac编程专家Mark Dalrymple和Scott Knaster撰写的《Objective-C基础教程》^①，该书浅显易懂，是学习Objective-C基础知识的优秀图书。网址如下所示。

<http://www.apress.com/book/view/9781430218159>

你可以参考Apple公司在*Learning Objective-C: A Primer*中对该语言的介绍：

http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/Learning_Objective-C_A_Primer

还可以读读*Objective-C Programming Language*，其中对该语言的介绍非常详尽全面，是一个上佳的参考指南，从以下网站可以获取该书：

<http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC>

最后一个资源可以从iPhone、iPod touch或iPad的iBooks免费下载。有了这些资源就完美了，可以继续阅读了！苹果公司已经发布了此种形式的好几种开发资料，希望还有更多惊喜！在iBooks中搜索“apple developer publications”可以找到这些内容。

1.3 编写 iOS 应用程序有何不同

如果从未使用过Cocoa或它的前期产品NextSTEP和OpenStep，那么你可能会发现Cocoa Touch（用于编写iOS应用程序的应用程序框架）稍显另类。它与其他常用应用程序框架（如用于构建.NET或Java应用程序的框架）之间存在一些根本差异。你起初可能会有点不知所措，但不必担心，只要勤加练习，就可以掌握其中的规律。

如果你具备使用Cocoa或NextSTEP编程的经验，就会发现iOS SDK中有许多熟悉的身影。其中的许多类都是从用于Mac OS X开发的版本中原样借鉴过来的，一些类即便存在不同，也遵循相同的基本原则，并使用与旧版本类似的设计模式。但是，Cocoa和Cocoa Touch之间却存在一些差异。

无论你的知识背景如何，都需要谨记iOS开发与桌面应用程序开发之间的重要差异。

1.3.1 只有一个应用程序正在运行

在iOS上，特定时间段内仅激活一个应用程序并在屏幕上显示。从iOS 4开始，当用户按下home按钮后应用程序有时可能继续在后台运行，但是这种情况也只限于特定的场合。

当应用程序未激活或未运行于后台时，它不会引起CPU任何形式的注意，这将导致网络连接断开以及其他问题。iOS 4在后台处理能力方面已经有了大幅提升，但要使你的应用程序在这种

6 第1章 欢迎来到 iPhone 的世界

情况下仍运行良好，还需要你自身的努力。

1.3.2 只有一个窗口

在桌面及笔记本电脑操作系统中，多个程序可以同时运行，并且可以分别创建和控制多个窗口。而iOS则有所不同，它只允许应用程序操作一个“窗口”。应用程序与用户的所有交互都在这个窗口中完成，而且这个窗口的大小就是iOS屏幕的大小，是固定的。

1.3.3 受限访问

计算机上的程序可以访问其用户能够访问的任何内容，而iOS则严格限制了应用程序的访问权限。

你只能在iOS为应用程序创建的文件系统中读写文件。此区域称为应用程序的沙盒，应用程序在其中存储文档、首选项等需要存储的各种数据。

应用程序还存在其他方面的限制。举例来说，你不能访问iOS上端口号较小的网络端口，也不能执行台式计算机中需要有根用户或管理员权限才能执行的操作。

1.3.4 有限的响应时间

由于使用方式特殊，iOS及其应用程序需要具备较快的响应时间。启动应用程序时，需要先打开它，载入首选项和数据，并尽快在屏幕上显示主视图，这一切要在几秒之内完成。

只要应用程序在运行，就可以从其下方拖出一个菜单条。如果用户按home按钮，iOS就会返回主页，用户需要快速保存一切内容并退出。如果未在5秒之内保存并放弃控制，则应用程序进程将终止，无论用户是否已经完成保存。

请注意，在iOS 4中，这种情况因为一种新API的存在而有所改善。这种API允许你的应用程序在终止前申请多一些的时间来处理。

1.3.5 有限的屏幕大小

iPhone的屏幕显示效果非常出色，从推出到现在，它一直是消费设备中分辨率最高的屏幕。

但是，iPhone的显示屏幕并不大，你施展的空间要比现代计算机小很多，最新的Retina显示器（iPhone 4和第4代iPod touch）仅有640×960像素，更老的仅有320×480像素。

而且，现在的尺寸与以前相同，只不过Retina显示屏为640×960像素，所以不要以为可以放更多的控件或什么了——仅仅是分辨率比以前高了。

iPad稍大一些，是1024×768像素，但也不算很大。与此形成鲜明对比的是，在撰写本书时，苹果公司最便宜的iMac支持1920×1080像素，最便宜的笔记本电脑MacBook支持1280×800像素。而苹果公司最大的显示器，27英寸的LED Cinema Display，支持超大的2560×1440像素。

1.3.6 有限的系统资源

阅读本书的任何资深程序员可能都会对256 MB内存、8 GB存储空间的机器嗤之以鼻，因为其资源实在是非常有限，但这种机器却是真实存在的。或许，开发iOS应用程序与在内存为48 KB的机器上编写复杂的电子表格应用程序不属于同一级别，二者之间没有可比性，但由于iOS的图形属性和它的功能，其内存不足是非常容易出现的。

目前上市的iOS的物理内存要么是256 MB，要么是512 MB，当然今后内存还会不断增长。内存的一部分用于屏幕缓冲和其他一些系统进程。通常，不到一半（也可能更少）的内存将留给应用程序使用。

虽然这些内存对于这样的小型计算机可能已经足够了，但谈到iOS的内存时还有另一个因素需要考虑：现代计算机操作系统，如Mac OS X，会将一部分未使用的内存块写到磁盘的交换文件中。这样，当应用程序请求的内存超过计算机实际可用的内存时，它仍然可以运行。但是，iOS并不会将易失性内存（如应用程序数据）写到交换文件中。因此，应用程序可用的内存量将受到手机中未使用的物理内存量的限制。

Cocoa Touch提供了一种内置机制，可以将内存不足的情况通知给应用程序。出现这种情况时，应用程序必须释放不需要的内存，甚至可能被强制退出。

1.3.7 不支持垃圾收集

之前已经说过，Cocoa Touch使用的是Objective-C 2.0，但该语言中的一个关键特性在iOS中并不可用：Cocoa Touch不支持垃圾收集。

1.3.8 新功能

前面提过，Cocoa Touch缺少Cocoa的一些功能，但iOS SDK中也有一些新功能是Cocoa所没有的，或者至少不是在任何Mac上都可用的。

- iPhone SDK为应用程序提供了一种定位方法，即用Core Location确定手机的当前地理坐标。
- 大部分iOS还提供了一个内置的照相机和照片库，并且SDK允许应用程序访问这两者。
- iOS还提供了一个内置的加速计（在最新的iPhone、iPod touch中有陀螺仪），用于检测机器的握持和移动方式。

1.3.9 与众不同的方法

iOS没有键盘和鼠标，这意味着在编程时它与用户的交互方式和通用的计算机截然不同。所幸的是，大多数交互都不需要你来处理。例如，如果在应用程序中添加一个文本字段，则iOS知道在用户单击该字段时调用键盘，你不需要编写任何额外的代码。

说明 目前的设备支持通过蓝牙连接外部键盘，这提供了一种不错的键盘体验并节省了一定的

屏幕空间，但这种使用情形仍然非常少。连接鼠标仍然完全行不通。

1.4 本书内容

下面是本书其余章节的简要概述。

第2章：讲述如何使用Xcode的搭档Interface Builder创建简单的界面，并在屏幕上显示一些文本。

第3章：介绍与用户的交互，构建一个简单的应用程序，用于在运行时根据用户按下的按钮动态更新显示的文本。

第4章：以第3章为基础，介绍其他一些iOS标准用户界面控件。此外，还将介绍如何使用警告框和操作表提醒用户作出决策，或者通知用户发生了一些异常事件。

第5章：了解如何处理自动旋转、自动改变大小属性，以及允许在纵向或横向模式下使用iOS应用程序的机制。

第6章：介绍更多高级用户界面，并阐述如何创建支持多视图界面的应用程序。我们将教你更改在运行时为用户显示的视图，以创建更加复杂的用户界面。

第7章：介绍如何在标准的iOS用户界面中实现标签条和拾取器。

第8章：介绍表视图。表视图是向用户提供数据列表的主要方法，并且是基于分层导航的应用程序的基础。这一章还会介绍如何让用户搜索应用程序数据。

第9章：介绍如何实现分层列表，它是最常用的iOS应用程序界面之一，你可以通过它查看更多或更详细的数据，学习实现这种标准界面时所涉及的技术。

第10章：iPad的外形与其他iOS设备不同，它需要用不同的方法来显示GUI，并借助一些组件来实现。这一章将介绍如何使用SDK中特定于iPad的内容。

第11章：介绍如何实现应用程序设置，iOS中的这种机制允许用户设置他们的应用程序级首选项。

第12章：介绍iOS中的数据管理。将讨论如何创建用于保存应用程序数据的对象，以及如何将这些数据持久存储到iOS的文件系统中。这一章还会介绍使用Core Data的基础知识，Core Data可用于方便地保存和检索数据。

第13章：从iOS 4开始，开发人员可以使用Grand Central Dispatch这种新方法进行多线程开发，还可以在某些情形下使他们的应用程序在后台运行。这一章将介绍如何实现此目的。

第14章：绘图是人们的普遍爱好，这一章介绍如何实现一些自定义绘图，这需要使用Quartz 2D和OpenGL ES中的基本绘图函数。

第15章：iOS的多点触摸屏幕可以接受用户的各种手势输入。这一章讲述如何检测基本的手势，如双指捏合和单指滑动，还将介绍定义新手势的过程，并讨论新手势的适用情况。

第16章：iOS可以通过Core Location确定其纬度和经度。这一章将编写利用Core Location计算

设备的物理位置的代码，并在各种应用中使用该信息。

第17章：介绍如何与iOS加速计和陀螺仪交互，通过它们确定设备的持握方式及运动速度与方向。我们将讨论应用程序如何通过该信息完成一些有趣的任务。

第18章：每个iOS都有自己的摄像设备和图片库，这两者都可供应用程序使用。这一章介绍如何使用它们。

第19章：iOS设备现已遍及90多个国家。这一章介绍以何种方式编写应用程序，能方便地把应用程序的所有部分翻译为其他语言，从而发掘应用程序的潜在用户。

第20章：至此，你已经掌握了iPhone和iPad应用程序的基本构建方法。这一章将探索掌握iOS SDK的后续步骤。

1.5 本次更新的内容

自本书上一版上市以来，iOS开发领域发生了很多事情。iOS开发社区的发展尤为显著。SDK一直在发展，苹果公司对SDK进行了大量更新。

当然，我们也很忙。从获悉SDK 4已面世那一刻起，我们就投入工作了。我们更新了每个项目，以确保每个项目的代码不但能在最新版的Xcode和SDK下编译，而且还能够充分利用Cocoa Touch提供的最新和最出色的特性。我们对全书进行了大量细微调整，也添加了数量可观的重大改变，包括代码以及解释。另外，我们新增了一个有关线程和多任务执行的章和有关iPad开发的章。当然我们也对全书重新进行了屏幕截图。

1.6 准备开始吧

iOS是一款出色的、令人难以置信的计算平台，是快乐开发的新领域。编写iOS应用程序将成为一种全新的体验，这种体验与之前你使用过的任何平台都不同。所有看似熟悉的功能都具有其独特的一面，但随着深入本书中的代码，你将能把这些概念紧密联系起来并融会贯通。

应该谨记，本书中的练习并不只是一份代码清单。完成这些练习之后，你也许就能成为iOS开发专家。在继续下一个项目之前，请确保已经理解了之前的概念和原理。不要害怕修改代码。多多尝试并观察结果是在Cocoa Touch等环境中克服编码困难的最佳方法。

如果你已经安装了iOS SDK，请继续阅读本书；如果还没有，请立即安装。然后开始iOS之旅！