

iPhone & iPad

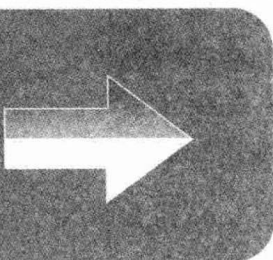
开发实战

孙东风 编著



海洋出版社

中国国内掌握iPhone 3G开发的工程师寥寥无几，大学还没有
相关专业的设置，学习iPhone&iPad开发从这里开始.....



iPhone & iPad

开发实战

孙东风 编著



海洋出版社
2010年·北京

内 容 简 介

iPhone 手机软件开发, 前景可期, 人才当立。为了帮助有志加入这一行业的初学者尽快成长, 作者结合多年商业项目经验编著此书。

主要内容: 本书分为必备知识和开发实战两大部分。必备知识包括 iPhone 平台开发基础、Objective-C 语言、Xcode 集成开发环境。开发实战给出 iPhone 个人记事本、iPhone 连连看游戏、iPhone Wap 浏览器、iPhone 手机聊天程序 4 个经典案例, 通过这些案例, 读者可以深入理解并掌握 iPhone 软件开发的全过程。

本书特点: 以作者工作过程中开发的多个商业化项目为雏形, 在此基础上进行加工, 以浅显易懂的方式进行关键知识点的解读, 帮助读者快速入门并获得触类旁通的能力。

适用范围: 本书可作为培训机构的教材, 还可作为初学者的进阶教材, 对中高层次程序开发人员也有一定的借鉴意义。

图书在版编目(CIP)数据

iPhone&iPad 开发实战/孙东风编著. —北京: 海洋出版社, 2010.8
ISBN 978-7-5027-7800-2

I. ①i… II. ①孙… III. ①移动通信—携带电话机—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字 (2010) 第 155059 号

总 策 划: 吕允英

责任编辑: 张鹤凌

责任校对: 肖新民

责任印制: 刘志恒

排 版: 北京京加诚彩印咨询有限责任公司

出版发行: 海洋出版社

地 址: 北京市海淀区大慧寺路 8 号 (705 房间)
100081

经 销: 新华书店

技术支持: (010) 62100057

发 行 部: (010) 62174379 (传真) (010) 62132549
(010) 62100075 (邮购) (010) 62173651

网 址: www.oceanpress.com.cn

承 印: 北京盛兰兄弟印刷装订有限公司

版 次: 2010 年 8 月第 1 版

2010 年 8 月第 1 次印刷

开 本: 787mm×1092mm 1/16

印 张: 20.75

字 数: 520 千字

印 数: 1~3000 册

定 价: 49.00 元

本书如有印、装质量问题可与发行部调换

前 言

时光荏苒，转眼间笔者已在无线领域摸爬滚打了 7 载有余。无线开发领域，也从当初单薄的 J2ME 平台发展到目前诸侯争霸的局面，Symbian、iPhone、Android、Ophone 各显神通，而 iPhone 依靠强大的技术创新，其智能机市场占有率直线攀升。这个世界终归会走向统一，统一这个世界的是 Symbian 抑或后起之秀 iPhone 仍然是未知数，但技术在纵深达到一定程度之后是可以一通百通的。正如于胜泳先生所言，当年能把 Visual C++ MFC 搞透的人，以后使用什么平台都有大将之风。笔者先后从事过 J2ME、Symbian、Android、iPhone 以及 Windows Mobile 平台的开发工作，这些平台无论是在界面架构还是在底层通信或者是数据持久存储方面，都有很大程度的相似。所以，如果精通了其中一个平台，即便以后从事其他平台的开发，也可以举一反三。这也是本书写作的目的，即更多地从程序架构、底层原理上深入讲解知识点细节，从而使读者可以触类旁通。

在笔者写作此书的时候，恰好 iPad 横空出世，因为 iPad 和 iPhone 是使用统一的 SDK 以及软件开发环境，本书中适用于 iPhone 开发的知识也同样适用于 iPad 开发，为了简洁起见，本书统一使用 iPhone 来代指 iPad 和 iPhone 两个平台。

在学习的过程中，建议读者把书中的实际例子进行反复实践并把实践过程中遇到的一些技术难题、心得体会及时总结下来，这无疑是个很好的习惯。在笔者写作本书的过程中，对于一些很基础的知识简单介绍，读者很容易找到相关的资料，所以本书基本上没有累赘的部分，建议读者能够逐章学习，无论是在校的大学生还是有工作经验的软件工程师，相信这些知识都能够有所帮助。本书的知识点结构如下：

章节	知 识 点	备 注
第 1 章	主要讲述了 iPhone 平台开发的基础知识，包括 iPhone 平台的简介、iPhone 开发环境的搭建以及 iPhone 应用程序框架的分析	新手必读
第 2 章	主要讲述了 iPhone 平台的原生编程语言 Objective-C 语言的知识，这些知识是进行 iPhone 应用程序开发的基础，Objective-C 语言的一些特性必须熟练掌握	进行 iPhone 平台下应用程序开发必须知道的知识点
第 3 章	主要讲述了 Xcode 集成开发环境，Xcode 是 Mac OS 下进行开发的集成环境，熟练地使用它可以使开发 iPhone 应用程序更得心应手	读者可选读
第 4 章	以 iPhone 个人记事本项目为实际例子讲解了 iPhone 核心应用界面体系架构、UITableView 视图应用、数据持久化存储、UINavigationController 导航控制器应用、多视图的管理并在最后介绍了如何使用 Doxygen 生成项目文档等	建议读者认真阅读并亲自动手实践



续表

章节	知 识 点	备 注
第 5 章	介绍了在 iPhone OS 下进行游戏开发需要学习的重要知识点, 主要包括 iPhone 复合视图、iPhone 按键处理、iPhone 全屏处理、屏幕刷新及事件触发、iPhone 音频技术等	学习完这些知识后读者应该有能力从事 iPhone OS 下游戏的开发
第 6 章	以 iPhone WAP 浏览器为实际例子重点讲述了 iPhone BSD Socket 通信机制以及 iPhone 应用程序中如何使用 TinyXML 等开源项目辅助开发	BSD Socket 通信是移动平台必须掌握的知识点
第 7 章	以 iPhone 手机聊天程序为实际例子重点讲述了 GameKit 蓝牙通信框架、CFNetwork 网络通信框架以及 iPhone 下进行联网应用程序开发的关键知识点, 重点讲述了 iPhone 客户端和 Java 服务器端以及 J2ME 客户端跨平台通信的知识点	CFNetwork 框架是 iPhone 下进行网络应用程序开发的关键, 而 GameKit 蓝牙通信框架也是 iPhone 中常用的知识点

本书的写作过程得到了很多人的帮助, 如前微软华东区技术总监于胜泳先生、掌讯通的程金良先生等, 笔者在这里一并致谢。

尽管笔者已经对本书的内容进行了反复审核, 但还是可能有所疏漏, 笔者会在自己的博客 <http://blog.csdn.net/dongfengsun> 上对本书进行维护, 也欢迎同行不吝指教。

孙东风

2010 年 4 月 28 日于上海



目 录

第 1 章 iPhone 平台开发基础..... 1	2.7 Objective-C 类别..... 65
1.1 iPhone 平台简介..... 2	2.8 Objective-C 协议..... 68
1.2 搭建 iPhone 开发环境..... 9	2.9 Objective-C 文件操作..... 70
1.3 iPhone 应用程序框架..... 10	2.9.1 Objective-C 属性列表..... 70
1.3.1 创建 iPhone 应用程序..... 10	2.9.2 Objective-C 对象编码..... 73
1.3.2 iPhone 应用程序开发的限制..... 12	2.10 Objective-C 多线程编程..... 75
1.4 “FirstApplication” 应用程序框架..... 13	2.10.1 Objective-C 多线程售票系统... 75
1.4.1 创建 iPhone 应用程序..... 13	2.10.2 Objective-C 多线程之 NSOperation..... 87
1.4.2 iPhone 应用程序界面设计..... 18	2.11 本章小结..... 88
1.4.3 iPhone 全屏自定义窗口..... 22	第 3 章 Xcode 集成开发环境..... 91
1.5 iPhone 视图控制器..... 26	3.1 Xcode 集成开发环境介绍..... 92
1.5.1 iPhone 视图控制器介绍..... 26	3.1.1 修改源文件版权信息..... 92
1.5.2 iPhone 视图控制器分类..... 27	3.1.2 使用 Xcode 组织和编辑源代码文件..... 92
1.6 打包并安装 iPhone 应用程序..... 31	3.1.3 使用 Xcode 的代码智能提示 功能..... 92
1.7 本章小结..... 33	3.1.4 使用 Xcode 集中编辑代码..... 94
第 2 章 Objective-C 语言..... 35	3.1.5 使用 Xcode 快速打开文件..... 95
2.1 Objective-C 和 C 语言..... 36	3.1.6 使用 Xcode 书签功能..... 95
2.1.1 Objective-C 语言版本“Hello World”..... 36	3.1.7 使用 Xcode 文档功能..... 96
2.1.2 “Hello World” 项目剖析..... 38	3.1.8 使用 Xcode 调试程序..... 96
2.1.3 Boolean 类型浅析..... 40	3.2 创建 Xcode 自定义模板..... 98
2.2 Objective-C 面向对象编程..... 41	3.3 Mac OS X 下 Bash 脚本编程..... 101
2.2.1 关于迂回..... 41	3.4 本章小结..... 108
2.2.2 面向对象编程中的迂回..... 45	第 4 章 iPhone 个人记事本..... 109
2.3 Objective-C 继承机制..... 49	4.1 预备知识..... 110
2.3.1 为什么使用继承..... 49	4.1.1 核心应用界面体系架构..... 110
2.3.2 Objective-C 继承语法..... 51	4.1.2 UITableView 视图应用..... 112
2.3.3 Objective-C 继承工作机制..... 53	4.1.3 数据持久化存储..... 117
2.4 Objective-C 内存管理..... 54	4.1.4 UINavigationController 导航 控制器应用..... 136
2.5 Objective-C 对象初始化..... 60	4.1.5 多视图的管理..... 156
2.5.1 为 Objective-C 对象分配内存 空间..... 60	4.1.6 使用 Doxygen 生成项目文档... 159
2.5.2 Objective-C 对象初始化..... 60	
2.6 Objective-C 属性..... 61	



4.2	设计要求	161	6.1.2	iPhone 屏幕双缓冲	227
4.3	类设计	162	6.1.3	iPhone BSD Socket 套接字	230
4.4	代码实现	163	6.1.4	iPhone HTTP 编程	240
4.4.1	程序打包及运行效果	163	6.2	设计要求	248
4.4.2	实现与代码分析	164	6.3	类设计	248
4.5	实践	164	6.4	代码实现	249
4.6	本章小结	164	6.4.1	程序打包及运行效果	249
第 5 章	iPhone 连连看游戏	165	6.4.2	实现与代码分析	250
5.1	预备知识	166	6.5	实践	269
5.1.1	iPhone 复合视图	166	6.6	本章小结	269
5.1.2	iPhone 按键处理	175	第 7 章	iPhone 手机聊天程序	271
5.1.3	iPhone 全屏处理	181	7.1	预备知识	272
5.1.4	屏幕刷新及事件触发	182	7.1.1	iPhone 蓝牙通信 GameKit 框 架	272
5.1.5	iPhone 音频技术	186	7.1.2	iPhone Socket 通信协议	291
5.1.6	iPhone 视频技术	196	7.1.3	iPhone 和 Java 通信协议	298
5.2	设计要求	201	7.2	设计要求	298
5.3	类设计	201	7.3	类设计	298
5.4	代码实现	202	7.4	代码实现	299
5.4.1	程序打包及运行效果	202	7.4.1	程序打包及运行效果	299
5.4.2	实现与代码分析	203	7.4.2	Java 服务器端实现	300
5.5	实践	220	7.4.3	Java 客户端实现	305
5.6	本章小结	220	7.4.4	iPhone 客户端实现	316
第 6 章	iPhone Wap 浏览器	221	7.5	实践	324
6.1	预备知识	222	7.6	本章小结	324
6.1.1	iPhone 静态库	222			

iPhone 平台开发基础

第

1

章



本章主要内容

- iPhone 平台简介
- iPhone 开发环境的搭建
- iPhone 应用程序框架
- iPhone 视图控制器

本章主要介绍 iPhone 的基础知识，由此勾画 iPhone 平台开发的清晰轮廓。

1.1 iPhone 平台简介

1993 年 Apple 公司制造出世界上第一款掌上电脑 (Personal Digital Assistant, PDA) (以下简称 PDA)，如图 1-1 所示。操作系统采用 Newton OS，但因为 Newton 在市场上找不到其定位而停止发展，1998 年 Apple 公司停止生产该款 PDA。

2007 年，Apple 公司推出基于 Mac OS X 10.5 精简版的手机 (被命名为 iPhone)，如图 1-2 所示。iPhone 自上市开始就在全球范围内不断地创造着销售奇迹。



图 1-1 Apple 公司生产的世界上第一台掌上电脑 (PDA)



图 1-2 2007 年 Apple 公司推出的 iPhone

基于 Mac OS X 的原因使得很多第三方软件很容易被移植到 iPhone 平台上，这一点使 iPhone 富有个性，它既可以当做 Web 服务器，还可以在 iPhone 上安装 PHP 环境，甚至可以像 Unix 那样使用 SSH 远程控制。



注意：SSH 是 Secure Shell (安全外壳协议) 的简称，它是一种在不安全网络上提供安全远程登录及其他安全网络服务的协议，最初是 Unix OS 上的一个程序，后来又迅速扩展到其他操作平台。SSH 是一个好的应用程序，在被正确使用时，它可以弥补网络中的漏洞。SSH 客户端适用于多种平台，几乎所有的 Unix 平台包括 Linux、AIX、Solaris、Irix 等都可以运行 SSH。而 WinSCP 是 Windows 环境下使用 SSH 的开源图形化 SFTP 客户端，同时也支持 SCP 协议，它的主要功能就是在本地和远程计算机之间安全地复制文件。

iPhone 的文件系统也是类 Unix 的，常用的文件夹如下。

- /Application

常用软件的安装目录。

- /Library/Ringtones

系统自带的来电铃声存放目录。

- **/Library/Wallpaper**

系统壁纸的存放目录。

- **/System/Library/Audio/UISounds**

系统声音文件的存放目录。

- **/private/var/root/Media/DCIM**

相机拍摄的照片文件的存放目录。

- **/private/var/root/Media/iTunes_Control/Music**

iTunes 上传的多媒体文件（如 MP3、MP4 等）的存放目录。

- **/private/var/root/Library/AddressBook**

系统电话本的存放目录，里面的文件可以使用 **wecontact** 电话本管理软件读取、修改以及重新上传等。

- **/private/var/root/Library/SMS**

系统短信的存放目录。

- **/private/var/run**

系统运行进程的目录。

- **/bin**

和 Unix 系统类似，是系统可执行指令的存放目录。

iPhone 平台支持丰富的应用程序开发接口，主要体现在以下几个方面。

- 丰富的标准控件库
- 多点触摸技术
- Quartz 2D 界面库
- OpenGL ES 渲染库
- 强大的音频、视频播放
- 重力传感器
- GPS
- 提供访问地址簿、照片库、照相机的 APIs
- SQLite 数据库进行数据的持久存储
- Push Notification

iPhone OS 的许多特性比如根据设备的方位自动调整视图，还有其他一些软件特性，比如所有的内置 iPhone 应用程序都共享一个联系人列表。因为很多特性都是基于用户体验的，开发者在初始设计时要考虑它们是否适用自己的应用程序，这些特性主要包括以下几个方面。

1) 使用加速计跟踪方向和运动

在 iPhone 中的加速计为开发的应用程序提供了许多有用的信息。1 个加速计会试图改变 1 个轴上的速度，iPhone 中有 3 个加速计，它们分别用来感应 3 个轴的加速度，开发者可以检测到任何方向上的变化。

开发者也许认为测量加速度的变化并不重要，但是这些信息会有一些意想不到的应用。地

心引力会把一个对象吸引到地面上,这个加速计即使在设备关机时都是存在的,跟踪这个加速计及其范围,可以相对精确地跟踪设备在一个三维空间内的位置。开发者可以把这个位置输入到自己的应用程序中。

系统使用加速计检测 iPhone 当前的位置,并通知应用程序屏幕是否旋转。如果应用程序同时支持横竖两种显示模式,那么在开始设计时就需要考虑视图控制器。UIViewController 类提供了自动旋转视图接口和调整视图位置的基础架构,以响应显示模式的改变。

开发者如果想直接获取原始的加速度数据,可以引用 UIKit 框架中的 UIAccelerometer 类,这类会根据设置的时间间隔报告当前的加速度数据,这样就可以发现设备的位置或者其他类型的瞬间速度,例如用户来回地晃动 iPhone。当然,开发者也可以在游戏或其他应用中使用这些数据。

2) 访问用户的内容

用户电话本列表是一个很重要的资源,系统中所有的应用程序都可以共享它。电话、邮件、短信程序都包含了用户电话本信息,以帮助用户进行一些方便的交流互动,比如电话、邮件或者文字短信可以直接使用电话本里的联系人信息。应用程序通过访问电话本列表做一些相同的事情,或者使用一些应用程序需要的其他信息。

3) 获取用户的当前位置信息

随着因特网的发展,人们可以在任何地方工作。收集和用户当前位置相关的信息将获得完美的用户体验。iPhone OS 核心位置框架从蜂窝基站或 Wi-Fi 热区得到定位信息转换为用户的当前位置。开发者可以使用这个框架初始化用户位置,或者提示用户位置的改变。这些信息可以为应用程序提供设备的定位或其他方面的信息。

4) 播放音频和视频

iPhone OS 通过核心音频 (Core Audio) 和 OpenAL 框架在应用程序中播放音频,并使用媒体播放框架支持视频的回放功能。核心音频提供高级的接口播放、记录和操作声音,并且解析音频流。开发者可以使用核心音频回放一个简单的音频或多声道的音频、混音并记录在音乐文件中,甚至出现 iPhone OS 的振动功能。对于游戏开发者来说,如果在代码中使用了 OpenAL,那么可以在游戏应用中回放声音。

媒体播放框架用来播放全屏的视频文件,这个框架支持播放许多格式的视频文件,并显示播放控制器,它包括是否显示控制器以及如何设置影片的长宽比例。游戏开发者可以用这个框架来播放分段的视频或者其他内容,基于媒体的应用可以采用该框架播放影片文件。

5) 使用内置的摄像头

iPhone 用户可以使用摄像头 (Camera) 来拍照,并把照片保存到统一的照片库中。尽管 iPad Touch 没有摄像头,但它也有一个照片库保存用户上传的照片。iPhone OS 提供了 UIKit 框架中的 UIImagePickerController 类来访问这些照片。

UIImagePickerController 类应用程序提供了摄像头和照片库的接口,包含摄像头和照片的应用程序可以使用这些标准的系统接口。当开发者需要访问这些接口时,访问控制会很小心地

处理用户的交互，并返回图片给应用程序。

iPhone 平台采用 Objective-C（通常简称为 Obj-C）语言进行应用程序的开发，Obj-C 是扩充 C 语言的面向对象的编程语言。

Obj-C 语言主要使用于 Mac OS X 和 GNUstep（详见 <http://www.gnustep.org/>）这两个使用 OpenStep 标准的系统中，Obj-C 可以在 Gcc（一个用于 Linux 系统下编程的编译器）运行的系统编辑和编译，因为 Gcc 包含 Obj-C 的编译器。

Obj-C 的最初版本并不支持垃圾回收机制，在当时这是争论的焦点之一，因为 Smalltalk 进行资源回收时有漫长的“死亡时间”，从而令整个系统失去作用。Obj-C 为避免此问题放弃引入这个功能，虽然某些第三方版本已经加入这个功能（比如 GNUstep），但 iPhone 中并未引入这个功能，因此需要在 iPhone 中引入“内存管理”的机制，Obj-C 语言在内存管理上采用了大多数编译器的 Reference count 技术，编译器为每个对象维护一个引用计数，在这个引用计数置“0”时释放对象占用的内存资源。如果要使用一个对象，并希望确保在使用期间该对象不被释放，那么需要程序设计人员通过函数调用来获取该对象的“所有权”，使用结束后再调用相应的函数释放该对象的“所有权”，所有权的获取和释放对应引用计数的增加和减少，为正数时代表该对象还有引用，为 0 时代表可以释放，iPhone 中获取所有权的函数包括以下几类。

- **alloc**

创建对象时调用 alloc，为对象分配内存，该对象的引用计数加 1。

- **copy**

拷贝一个对象，返回新的对象，引用计数加 1。

- **retain**

引用计数加 1，获取对象的所有权。

iPhone 中释放所有权的函数包括以下几类。

- **release**

引用计数减 1，释放所有权，若引用计数减到 0，对象会被释放。

- **autorelease**

可能在某些情况下，程序设计人员既不想获取所有权，又不希望对象被释放。例如，在一个函数中生成了一个新对象并返回，函数本身并不希望获取该新生成的对象的所有权，因为获取后再也没有机会释放，又不可能在函数内释放。这个时候可以借助 autorelease，可以理解为把所有权交给了一个外在的系统（这个系统实际上叫做 autorelease pool），由它来管理该对象的释放。Apple 官方文档并不提倡使用 autorelease 系统，因为这个系统效率不高，所以读者需要尽量自己负责对象所有权的释放。

另一个问题是 Obj-C 并不支持命名空间机制（Namespace Mechanism），取而代之的是程序设计人员必须在其类名称加上前缀。2004 年，在 Cocoa 编程环境中，所有 Mac OS X 类型和函数均有“NS”作为前缀，例如 NSObject、UIButton 等来清楚辨别它们属于 Mac OS X 核心类库。作为一门新的编程语言，Obj-C 还有众多不同于 C 语言的特性，这些特性如下。

（1）Obj-C 语言使用 *.m 后缀作为代码文件的扩展名，当然也可以使用 *.c 和 *.cpp 后缀

名，但它们会分别调用 C 编译器和 C++编译器。

(2) Obj-C 使用 `#import` 代替 `#include` 来导入头文件声明，并且编译器保证 `#import` 不会被重复导入。

(3) 尽管在 Obj-C 中可以使用 `printf()` 在控制台打印文本信息，但是更推荐使用 `NSLog()`，并且它会自动在文本后面添加 `'\n'`。

(4) 为了更好地体现其 OOP 特性，NextStep 和 Java 一样，所有的对象都继承自 `NSObject` 类，所以 `Application Kit` 里的类前缀基本都是 `NS*`，而 `Cocoa Foundation` 里的类前缀基本都为 `NSCF*`。

Obj-C 里提供了 `BOOL` 类型，这个类型是个宏定义，定义如下。

```
Typedef signed char BOOL;
```

通常返回值为 `YES` 或 `NO`，这两个值也是宏定义，定义如下。

```
#define YES (BOOL)1
```

```
#define NO (BOOL)0
```

这里和 C++ 并不一样，在 C++ 里一切非 0 值都为 `true`，0 值为 `false`。所以大多数读者容易犯下面的错误。

```
BOOL areIntsDifferent_faulty(int thing1,int thing2)
{
return (thing1-thing2);
}
```

```
if(areIntsDifferent_faulty(23,5) = YES)
{
}
```

上面的代码中，`areIntsDifferent_faulty()` 返回的是两个整数的差值，如果这个差值不为 1，那么返回值永远不会为 `YES`。

(1) Obj-C 和 C 语言一样，其 `Primitive Type` 只有 `int`、`char`、`float`、`BOOL` 并用它们表达一切。

(2) Obj-C 采用 `@interface` 格式进行接口的声明，并采用 `@implementation` 对声明进行实现。

(3) Obj-C 在 `Foundation Kit` 里提供了很多基础类库，常用的如 `NSString`、`NSLog`、`NSPoint`、`NSRect`、`NSArray` 等。

(4) Obj-C 支持 `Collection` 类库，比如 `NSArray`、`NSMutableArray` 等，这些类库提供基本的数据结构操作如插入、删除、排序等。

Obj-C 采用“迭代器” `NSEnumerator` 进行元素的遍历，用法如下。



```
NSEnumerator* enumerator = [array objectEnumerator];
id thingie;
```

```
while(thingie = [enumerator next( )bject])
{
    NSLog(@"I found %@",thingie);
}
```

Obj-C 不支持运算符重载。

Obj-C 提供 Selector 选择器支持函数指针的调用，典型用法如下。

```
SEL 变量名 = @Selector(方法名);
SEL 变量名 = NSSelectorFromString(方法名字符串);
NSString* 变量名 = NSStringFromSelector(SEL参数);
```

Obj-C 既然是面向对象的 C 语言，那么就需要支持面向对象的 3 个基本特征：继承、多态、封装，下面逐一进行介绍。

(1) 继承。继承在 Obj-C 语言里的使用非常普遍，也是进行 OOP 编程的基本特性，如果有两个类 Circle、Rectangle 定义如下。

```
@interface Circle:NSObject
{
    ShapeColor fillColor;
    ShapeRect bounds;
}
```

```
-(void) setFillColor:(ShapeColor)fillColor;
-(void) setBounds:(ShapeRect)bounds;
-(void) draw;
@end //Circle
```

```
@interface Rectangle:NSObject
{
    ShapeColor fillColor;
    ShapeRect bounds;
}
```

```
-(void) setFillColor:(ShapeColor)fillColor;
-(void) setBounds:(ShapeRect)bounds;
-(void) draw;
@end // Rectangle
```

可见，Circle、Rectangle 类的属性和行为非常类似，所以可以通过封装、继承进行代码重构，如下。

```
@interface Shape:NSObject
{
    ShapeColor fillColor;
    ShapeRect bounds;
}

@implementation Shape
-(void) setFillColor:(ShapeColor)color
{
    fillColor = color;
}
-(void) setBounds:(ShapeRect)rect
{
    bounds = rect;
}
-(void)draw
{
}
@end //Shape
```

Shape 类对 Circle、Rectangle 类的数据和操作进行了基本的抽象封装，继承 Shape 类后如下。

```
@interface Circle:Shape
@end //Circle

@interface Rectangle:Shape
@end //Rectangle

@implementation Circle
-(void) draw
{
    NSLog(@"Drawing a Circle");
}
@end //Circle

@implementation Rectangle
-(void) draw
{
    NSLog(@"Drawing a Rectangle");
}
@end // Rectangle
```



(2) 多态。Obj-C 语言使用[super Init]调用父类的方法，每一个类都有一个指向自己的 self 指针，这点和 C++ 的隐含 this 指针是类似的。

另外 Obj-C 语言并不支持多继承，笔者猜测是因为多继承的开销过大，需要解决虚基类的问题，在嵌入式平台资源有限的情况下，支持多继承并不现实。

(3) 封装。封装是面向对象的基本特性，主要依靠软件设计人员对现实模型进行抽象设计。Obj-C 用 *.h 文件封装接口的声明，以 *.m 文件实现接口。



注意：GCCE 是 Codesourcery 公司基于 ARM GNU Toolchain 开发的，更多的详细信息可以访问 Codesourcery 官方网站：<http://www.codesourcery.com/>。

1.2 搭建 iPhone 开发环境

在 iPhone 开发过程中，Apple 官方只支持 iPhone SDK 下进行开发。在进行 iPhone 开发之前，读者需要做以下准备工作。

(1) 基于 Mac OS X 的电脑，可以选择 iMac、Mac Book、Mac Book Pro、Mac Book Air 等。

(2) iPhone、iPad Touch。

(3) iPhone Developer Program，个人版 99 美元/年，企业版 299 美元/年。

(4) 注册 ADC 会员是免费的，并且可以获得很多开发资料，注册地址为 <http://developer.apple.com/iphone/program/start/register/>。

在成功注册之后，就可以下载 iPhone SDK 了，下载完毕后打开 Packages 包可以看到里面包含了 2.0、2.1、2.2.1、2.2、3.0 等版本的 SDK，双击根目录下的 iPhone SDK.mpkg 文件进行安装。

iPhone SDK 由几个组件构成 iPhone 基本的开发环境，这些组件功能如下。

● Xcode

Xcode 是 iPhone 应用程序开发最重要的工具，它提供了 iPhone 应用开发的代码编辑工具、技术文档、debugger 工具等，Xcode 由开源的 GNU 工具如 Gcc 编译器、Gdb 调试器等组成。

● Instruments

Instruments 以数据图表、数据曲线的方式显示 iPhone 应用程序的工作状态，包括内存使用、应用程序性能等，这使得软件设计人员可以清楚应用程序设计中存在效率问题的地方。Instruments 基于 Sun Microsystems 公司的开源项目 DTrace 构建而成，它在追踪应用程序中的内存泄露问题起着至关重要的作用。

● Dashcode

Dashcode 是用于创建独立的 iPhone Web 应用程序的工具，这些应用程序并不运行在传统的浏览器环境下。

● iPhone 模拟器

Xcode 会为模拟器编译 Intel x86 的二进制代码而不是运行在真机上的 ARM-based 二进制代码。

● Interface Builder

Interface Builder (IB) 是一个快速设计用户界面的工具, IB 使得读者可以使用 visual design tools 绘制用户界面并连接这些界面元素到 Xcode 中相应的对象和方法上。

搭建 iPhone 平台开发环境要求做到如下几点。

(1) 到 Apple 官方网站注册成为 ADC 会员。

(2) 到 Apple 官方网站下载 iPhone SDK, 下载完毕后按默认设置一路引导到 Custom Install on “Macintosh HD” 界面。在这个界面上可以看到 iPhone SDK 选择处于不可选状态, 把可以选择的全部选上, 这一步已经安装好了 XCode3.1.4。

 **注意:** Developer Tools Essentials 包默认的安装路径为 /Developer, 在下面安装 iPhone SDK 时也需要安装到这个路径下。

(3) 如图 1-3 所示, 到 iPhone SDK 目录下的 Packages 文件夹中寻找图中的 5 个安装包逐个进行安装, 这些包都以 iphone* 开头, 安装到 /Developer 目录下。

(4) 到 iPhone SDK 目录下的 Packages 文件夹中安装相应版本的 iPhone SDK, 这里安装了 2.0、2.1、2.2.1、2.2、3.0 等版本。

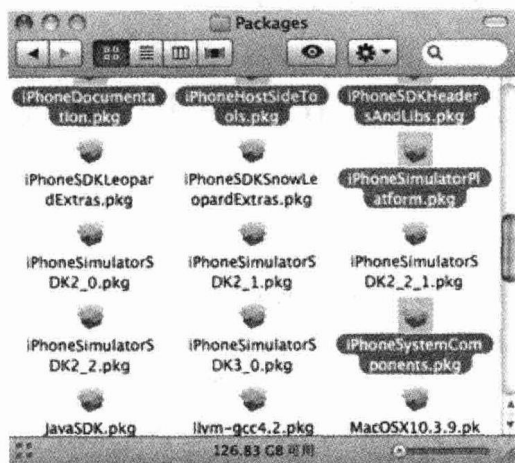


图 1-3 选择安装 5 个 iPhone* 包

1.3 iPhone 应用程序框架

iPhone 应用程序框架提供了好几个开发模板, 包括 Navigation-based Application、OpenGL ES Application、Tab Bar Application、Utility Application、View-based Application、Window-based Application, 在这些开发模板中, Window-based Application 是最为基础的, 它相当于一个空模板, 所有的 iPhone 应用程序都可以基于 Window-based Application 模板进行开发。

1.3.1 创建 iPhone 应用程序

打开 Xcode 创建一个基于 Window-based Application 的应用程序 “FirstApplication”, 如图 1-4 所示。

生成后的 “Window-based Application” 模板由一个 main.m 源文件、iPhone 默认的项目框架 (如 UIKit、Foundation、Core Graphics 等)、一些资源文件 (如 Info.plist) 等组成, 这些文件都是创建一个基本的 iPhone 应用程序所必须具备的。