

软件创富密码:

王志刚作品系列

您也可以赚取您的第一桶金

本书适合没有C语言基础以及面向对象编程知识的朋友学习Objective-C

iPhone 应用程序开发攻略之 深入浅出Objective-C 2.0

适用于iOS 4

王志刚 等编著



王志刚 作品系列

iPhone 软件创富密码:

应用程序开发攻略之深入浅出Objective-C 2.0

王志刚 等编著

電子工業出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书系统地介绍了关于使用Objective-C 2.0开发iPhone应用程序的基础知识,涵盖各种开发工具的操作技巧、框架(Framework)内部的工作原理、软件的调式技巧、后期制作等基础知识。书中花费大量篇幅介绍开发iPhone应用程序时所必需的各种常用类库,结合针对性强的应用实例的讲解,可进一步加深读者对这些常用类库的理解。

本书可作为iPhone应用程序开发的入门教材使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

软件创富密码: iPhone应用程序开发攻略之深入浅出Objective-C 2.0/王志刚等编著. —北京: 电子工业出版社, 2011.6

ISBN 978-7-121-13469-2

I. ①软… II. ①王… III. ①移动电话机—应用程序—程序设计②C语言—程序设计
IV. ①TN929.53②/TP312

中国版本图书馆CIP数据核字(2011)第084562号

策划编辑: 孙学瑛

责任编辑: 刘 舫

印 刷: 北京天宇星印刷厂

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱

邮编: 100036

开 本: 787×1092 1/16

印张: 22.25 字数: 463千字

印 次: 2011年6月第1次印刷

印 数: 4000册 定价: 69.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至zlts@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线: (010) 88258888。



今天像Android, Blackberry, Windows Phone, iPhone等智能手机广受关注, 特别是iPhone。2010年伊始, 苹果公司连续推出了iPhone的升级产品iPhone 4, 以及与iPhone使用相同OS的iPad, 2011年又推出了iPad 2, 给业界带来了强烈的冲击。

苹果公司的iPhone, iPad等产品广受消费者支持的原因, 不仅在于其前卫新颖的外观设计, 简洁直观的用户界面, 更在于其能给用户提供多种多样的应用程序的选择。苹果的App Store中已经汇集了十几万种应用程序供用户选择, 而且每天还有各种专业或业余的程序设计人员不断将自己的作品汇入其中。可以肯定地说在不远的将来, 我们还将继续见证苹果移动电子产品的成长壮大。

本书目的不是要惊叹iPhone, iPad等产品的伟大, 而是想告诉大家, 其实我们也可以在这个不断成长的大饼中, 利用自己的智慧分得一杯羹。只要您有好的创意, 即使您是没有程序设计经验的家庭主妇, 只要您做出了应用程序, 加入到App Store中后, 也可以从苹果移动电子产品的全球用户群中赚取您的利润。运气好的话, 甚至是一笔不菲的收入呢, 这方面成功的例子有很多。

本书内容

至于iPhone应用程序的开发更不是什么高不可攀的东西, 本书从易到难介绍iPhone应用程序编程的知识, 同时还会有笔者在开发iPhone应用程序过程中获得的许多心得体会。

本书主要介绍iPhone应用程序的开发语言Objective-C, 以及Cocoa与Cocoa Touch框架 (framework) 的基础知识 (Cocoa Touch框架用于iPhone应用程序, 而Cocoa框架用于



Mac应用程序)。同时还花费了一定篇幅来介绍关于iPhone应用程序开发的基本概念方面的知识,如MVC框架概念、内存管理概念等。

iPhone应用程序开发必须要在Mac电脑上进行,对于那些没有使用Mac电脑的朋友,书中也穿插了许多关于Mac电脑的使用知识,介绍了开发iPhone以及Mac应用程序时专用的、主要的几种开发工具的使用,如Xcode, Interface Builder等。涵盖从应用程序开发到最终应用程序发布为止的,几乎所有会使用到的开发工具。

本书中围绕相关知识的介绍,配备了详细的实例。具体的开发相关知识都是以实例为中心进行展开的,力争做到通俗易懂。

本书读者

开发iPhone应用程序时基本上都要使用Objective-C(部分有使用DashCode开发的Web应用程序),因此掌握Objective-C及相关框架知识是首要的前提条件。Objective-C语言的本质是C语言的扩展版本,是一种面向对象型的开发语言。有了C语言或者面向对象语言的基础知识,将有助于您掌握Objective-C,但是并不意味着没有这些,学习起来就会很困难。通常iPhone应用程序的大部分功能都是由各种框架(如iPhone应用程序的界面由AppKit框架实现)来完成的,实际代码几乎是约定俗成的,因此不用拥有对面向对象编程概念很深的理解,或者更深的C语言知识就能够着手编程了。笔者认为,本书正是适合那些没有C语言基础以及面向对象编程知识的朋友。当然对于拥有这方面知识的朋友可以将有关章节当做复习或者参考,毕竟Objective-C的面向对象的概念与其他语言如Java等还是有所区别的。

参与本书编写的人员有朱蕾、罗伟、黄建峰、朱至濂。另外,在此特别感谢我父母在本书编写过程中给予的大力支持。

作者

2011年3月



第1章 配置开发环境	2
1.1 Mac应用程序开发环境	2
1.2 iPhone应用程序的开发环境	3
1.3 查看安装目录中的内容	5
1.3.1 启动Xcode	6
1.3.2 启动Interface Builder	7
1.4 Mac中的基本操作	7
1.4.1 弹出菜单的显示方法	7
1.4.2 工具条的定制	9
第2章 开发工具	11
2.1 Xcode	11
2.1.1 综合开发环境Xcode的功能	11
2.1.2 工程窗口	12
2.1.3 文本编辑器的功能	16
2.1.4 开发者手册	18
2.2 Interface Builder	20
2.3 Dashcode	20
2.4 iPhone模拟器	21
2.5 其他工具	22
2.5.1 FileMerge	22
2.5.2 Icon Composer	23
2.5.3 Property List Editor	23
2.6 Mac或iPhone应用程序开发的流程	23
2.6.1 准备应用程序用的材料	24
2.6.2 建立步骤	25
2.6.3 gcc	26



第3章 应用程序与框架	27
3.1 关于应用程序	27
3.1.1 应用程序的种类	28
3.1.2 关于iPhone应用程序	29
3.2 框架	29
3.2.1 简单的“Hello World”程序	29
3.2.2 框架的出现	30
3.2.3 编程语言与框架的关系	30
3.3 Mac OS X以及iPhone SDK中的框架	31
3.3.1 框架的安装目录	31
3.3.2 Cocoa与Carbon	32
3.3.3 应用程序中使用的框架	32
3.4 Cocoa与Carbon的发展轨迹	34
3.4.1 Cocoa是面向对象型的框架，所以挺难掌握	35
3.4.2 Carbon是用C开发的，懂语言的人应该用Carbon	35
3.4.3 Carbon将来会消失	35
3.4.4 Carbon应用程序使用Cocoa进行改写后速度会提高	35
3.4.5 如果使用Windows版的Cocoa进行开发，在Windows上也能运行Cocoa应用程序	36
第4章 Cocoa与Objective-C	37
4.1 框架的使用	37
4.1.1 关于API	37
4.1.2 API与编程语言的关系	39
4.2 Objective-C的发展史	39
4.3 用于其他语言中的Cocoa API	41
4.3.1 通过桥调用Cocoa	41
4.3.2 是否应提倡在Objective-C以外的语言中调用Cocoa API的开发方式	42
4.4 解答与Objective-C相关的几个疑问	43
4.4.1 是否需要先掌握C语言，以及Smalltalk呢	43
4.4.2 Objective-C这样冷门的语言不好学吧	43
4.4.3 Objective-C也能广泛运用在Windows或Linux平台上吗	44
4.4.4 Objective-C将来会消失吗	44
第5章 Cocoa面向对象编程的基本概念	45
5.1 面向对象的5个基本术语	45
5.1.1 类	46
5.1.2 实例	46

5.1.3	实例变量	47
5.1.4	方法	48
5.1.5	继承	48
5.1.6	相互关系	49
5.1.7	对象	50
5.2	在Cocoa实例中理解基本术语	50
5.2.1	通过按钮配置理解基本术语	50
5.2.2	文本输入框的继承关系	51
第6章	Hello World应用程序	53
6.1	关于Cocoa版的Hello World程序	53
6.2	启动Xcode创建工程	54
6.3	创建类	56
6.3.1	创建类文件	56
6.3.2	追加代码	56
6.4	启动Interface Builder设计用户界面	58
6.4.1	xib文件	58
6.4.2	Interface Builder窗口	59
6.4.3	设计Hello World的用户界面	61
6.5	类的实例化	62
6.6	连接插座与动作	64
6.7	完成AppController的代码编写	67
6.8	建立与运行	67
6.9	开发步骤再回顾	69
第7章	MVC构架	70
7.1	Hello World应用程序开发流程中包含的信息	70
7.2	什么是MVC构架	70
7.2.1	视图层	71
7.2.2	模型层	71
7.2.3	控制层	72
7.3	使用MVC构架的实例	72
7.3.1	iTunes的模型	72
7.3.2	iTunes的视图	73
7.3.3	iTunes的控制	74
7.4	使用MVC构架的Cocoa应用程序开发特点	75
7.4.1	视图组件由Cocoa提供	75



7.4.2	保持视图与模型间的相互独立	75
7.4.3	应用程序特有的功能由控制提供	75
7.4.4	只用创建模型以及控制的类	76
第8章	插座与动作	77
8.1	视图与控制间的信息交换	77
8.2	插座与动作	78
8.2.1	什么是插座	78
8.2.2	什么是动作	78
8.3	插座以及动作的连接	79
8.3.1	插座与动作的追加	79
8.3.2	插座的连接	80
8.3.3	动作的连接	81
8.3.4	连接确认	82
第9章	C语言语法基础	85
9.1	C语言概论	85
9.1.1	C语言的发展过程	85
9.1.2	C语言的特点	85
9.1.3	C源程序的结构特点	86
9.1.4	C源程序的构成规则	89
9.1.5	C语言的字符集	89
9.1.6	C语言词汇	90
9.2	数据类型	92
9.2.1	基本数据类型	92
9.2.2	基本运算符和表达式	98
9.2.3	关系运算符和表达式	101
9.2.4	逻辑运算符和表达式	102
9.3	数组	103
9.3.1	一维数组	103
9.3.2	二维数组	104
9.3.3	字符数组	104
9.4	指针	104
9.4.1	指针的基本概念	105
9.4.2	指针变量的类型声明	105
9.4.3	指针变量的赋值	106
9.4.4	使用字符串指针变量与字符数组的区别	106
9.5	C语言的输出语句printf()函数	107
9.5.1	printf()函数调用的一般形式	107

9.5.2	格式字符串	107
9.5.3	输出的实例	109
9.6	控制语句	110
9.6.1	if语句	110
9.6.2	switch语句	113
9.6.3	while语句	115
9.6.4	do...while语句	116
9.6.5	for语句	117
9.7	函数	119
9.7.1	函数的语法	120
9.7.2	函数的声明与实体	121
9.7.3	可变参数	122
9.7.4	标准函数库	122
9.8	结构	123
9.8.1	结构的定义	123
9.8.2	结构类型变量的声明	124
9.8.3	结构变量的取值与赋值	125
9.9	在Xcode中创建学习C语言用的工程	125
第10章	Objective-C的语法	129
10.1	Objective-C的编程概论	129
10.1.1	Objective-C与C语言	129
10.1.2	Objective-C程序	129
10.1.3	类的声明与实体	130
10.1.4	创建Objective-C的测试工程	130
10.2	类的声明	135
10.2.1	导入声明文件	135
10.2.2	类的声明	136
10.2.3	实例变量的声明	137
10.2.4	实例方法的声明	137
10.3	编写类的实体	142
10.3.1	编写类的实体	142
10.3.2	编写方法代码	143
10.3.3	关于方法的声明	144
10.4	对象专用变量类型	144
10.4.1	实例对象专用的变量类型	144
10.4.2	id类型	145
10.4.3	nil类型	145
10.4.4	在条件表达式中使用对象变量	145



10.5	方法调用	146
10.5.1	实例方法的调用	146
10.5.2	类方法的调用	147
10.5.3	对象为nil的情况下	148
10.6	命名规则	148
10.6.1	类的命名规则	148
10.6.2	实例变量的命名规则	149
10.6.3	方法的命名规则	149
10.7	重新解析Cocoa的Hello World程序	150
10.7.1	AppController.h	150
10.7.2	AppController.m	151
第11章	内存管理	153
11.1	内存分配与释放	153
11.2	与内存相关的问题	154
11.2.1	内存释放的时机	154
11.2.2	内存泄漏	155
11.3	使用垃圾收集进行内存管理	155
11.3.1	垃圾回收	155
11.3.2	Objective-C与垃圾回收	157
11.3.3	垃圾回收功能有效化	158
11.3.4	垃圾回收编程的规则	159
11.4	由参照统计实现的内存管理	159
11.4.1	参照统计的基本概念	159
11.4.2	retain和release方法	160
11.4.3	对象保持到释放的流程	160
11.5	类的实例化	162
11.5.1	类的实例化与初始化	162
11.5.2	自动释放	163
11.5.3	创建实例的方法	164
11.5.4	实例的释放	165
11.6	内存管理的规则	166
11.6.1	临时对象	166
11.6.2	内存管理经验总结	168
第12章	字符串	169
12.1	字符串专用类	169
12.2	NSString与NSMutableString	170
12.3	字符编码	170

12.3.1	什么是字符编码	170
12.3.2	NSString的字符代码为Unicode	171
12.3.3	NSStringEncoding	171
12.4	字符串的生成	172
12.4.1	使用@"的形式创建新字符串	172
12.4.2	创建中文字符串	173
12.4.3	使用格式创建字符串	174
12.5	NSRange	176
12.6	字符串处理	177
12.6.1	计算字符串的长度	177
12.6.2	字符串的连接、插入和删除	178
12.6.3	字符串的比较	181
12.6.4	字符串的检索	183
12.6.5	抽取部分字符串	184
12.7	读/写文本文件	186
12.7.1	读取文本文件	186
12.7.2	输出文本文件	187
第13章	集合	189
13.1	数组	189
13.1.1	NSArray与 NSMutableArray	189
13.1.2	数组的创建	190
13.1.3	数组的长度	191
13.1.4	通过索引取得对象	192
13.1.5	使用NSEnumerator取得对象	193
13.1.6	使用高速枚举法取得对象	196
13.1.7	对象的追加与删除	197
13.2	字典	199
13.2.1	什么是字典类型	199
13.2.2	NSDictionary与 NSMutableDictionary	200
13.2.3	NSDictionary的创建	200
13.2.4	键与值的取得	202
13.2.5	键与值的追加	203
13.3	包装类	205
13.3.1	什么是包装类	205
13.3.2	NSNumber	205
13.4	对象的等价与同值	208
13.4.1	拥有相同值的对象	208
13.4.2	等价与同值的判定	208

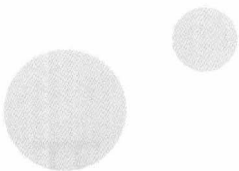


13.4.3	在集合类中的应用	210
13.5	集合类与参照统计	210
13.5.1	NSArray与参照统计	210
13.5.2	NSDictionary与参照统计	212
第14章	XML应用	213
14.1	XML概要	213
14.1.1	什么是XML	213
14.1.2	一种常用的XML实例——RSS	213
14.2	XML的解析	215
14.2.1	解析及解析器	215
14.2.2	XML解析器的种类	215
14.2.3	树结构	216
14.3	Cocoa中的XML解析器	218
14.3.1	NSXMLDocument, NSXMLNode和NSXMLElement	218
14.3.2	XML文档的解析器	219
14.3.3	Xpath	221
14.3.4	节点信息	222
14.3.5	节点的父子关系	223
14.3.6	属性值的取得	225
第15章	表格视图	227
15.1	表格视图	227
15.1.1	NSTableView与NSTableColumn	227
15.1.2	表格的构造	229
15.2	数据资源	229
15.2.1	在表格中使用数据资源	229
15.2.2	数据资源的方法	230
15.2.3	NSTableView的方法	231
15.3	实例	231
15.3.1	数据资源的设置	238
15.3.2	numberOfRowsInTableView:	238
15.3.3	tableView:objectValueForTableColumn:row:	239
15.3.4	单击Read按钮的处理代码	242
第16章	UIKit框架简介及如何创建iPhone表格视图	245
16.1	UIKit概要	246
16.1.1	UIKit框架的类一览表	246
16.1.2	UIKit框架与AppKit框架的区别	249

16.2	创建测试UIKit类的试验工程	250
16.2.1	创建第一个iPhone应用程序——Hello World工程	251
16.2.2	修改Hello World工程使之成为不使用xib文件的实验工程	256
16.3	创建最简单的表格	261
16.3.1	创建表	262
16.3.2	追加选择单元时运行的动作	265
16.4	iPhone表格视图的相关类参考	265
16.4.1	UITableView类	266
16.4.2	UITableViewCell类	272
16.4.3	UITableViewController类	279
16.4.4	UITableViewDataSource协议类	280
16.4.5	UITableViewDelegate协议类	284
第17章	常见的错误与警告	287
17.1	什么是错误与警告	287
17.2	Xcode中的错误与警告	287
17.2.1	错误与警告的显示方式	288
17.2.2	建立结果窗口	288
17.2.3	代码窗口中的显示	289
17.3	经常会出现的错误信息	290
17.3.1	‘XXX’ undeclared (first use in this function)	290
17.3.2	parse error before ‘XXX’ token	291
17.3.3	invalid preprocessing directive #XXX	291
17.3.4	No Such file or directory	292
17.3.5	Undefined symbols	292
17.4	经常会出现的警告信息	293
17.4.1	‘XXX’ may not respond to ‘YYY’	293
17.4.2	unused variable ‘XXX’	294
17.4.3	local declaration of ‘XXX’ hides instance variable	295
17.4.4	incomplete implementation of class ‘XXX’	296
17.4.5	control reaches end of non-void function	297
17.4.6	passing argument n of ‘XXX’ assignment from distinct Objective-C type	298
第18章	调试方法	299
18.1	缺陷与调试	299
18.1.1	什么是缺陷与调试	299
18.1.2	描述缺陷	300
18.1.3	缺陷的重现	300



18.2	NSLog	301
18.2.1	使用NSLog输出调试信息	301
18.2.2	NSLog的格式	302
18.3	调试工具	304
18.3.1	什么是调试工具	304
18.3.2	调试构建配置	305
18.3.3	断点与调试模式的启动	305
18.3.4	调试界面	306
18.3.5	调试的分步执行	308
18.4	例外	309
18.4.1	什么是例外	309
18.4.2	捕捉例外	309
18.4.3	发生例外时停止运行	312
18.4.4	避免使用太多的例外	313
第19章	应用程序的后期制作	314
19.1	应用程序图标	314
19.1.1	制作图标的图片	315
19.1.2	图标格式文件的制作	316
19.1.3	向工程中追加图片文件	317
19.2	应用程序的设置	317
19.2.1	什么是目标	317
19.2.2	与目标相关的应用程序设置	318
19.2.3	Info.plist	321
19.3	本地化	322
19.3.1	什么是本地化	322
19.3.2	用户界面的本地化	323
19.3.3	应用程序名称的本地化	326
19.3.4	本地化的测试	328
19.4	构建配置	329
19.4.1	什么是构建配置	329
19.4.2	将构建配置设为Release	329
19.5	完成的应用程序	330
附录A	加入iPhone Developer Program	331
索引		337



写在前面的话

iPhone应用程序绝大多数是使用Objective-C（Objective-C 2.0是其最新版本）编写的，因此在学习开发iPhone应用程序之前，需要很好地掌握Objective-C 2.0。本书主要讲解Objective-C 2.0的相关知识，力求深入浅出。

通常在Objective-C 2.0中使用被称为Cocoa的框架，因此使用Objective-C 2.0的应用程序开发，有时会被称为Cocoa编程。因为本书还没有具体使用iPhone SDK，本章的Cocoa应用程序实例都只是运行在Mac OS（10.0以后的版本）上的。一旦掌握了这些基础的Objective-C 2.0编程知识，结合iPhone SDK很容易开发出自己的iPhone应用程序，关于此，本系列的第二本书（书名为《iPhone应用程序开发攻略之iPhone特色传感器应用》）中会有详细介绍。



配置开发环境

在进行实际应用程序开发之前，首先要配置开发环境。本章介绍如何配置iPhone应用程序的开发环境。

1.1 Mac应用程序开发环境

“开发环境”有时也被称为“集成开发环境（IDE：Integrated Development Environment）”，其中包含了编程时所需要的应用程序以及文档等。使用现在流行的开发语言（如Java）进行应用程序开发时都需要另外准备专门的IDE软件，有各种商用版本或者免费版本。值得庆幸的是，在Mac OS中默认提供了名为Xcode的IDE软件，可以立即使用它进行MAC应用程序开发，当然也可以开发iPhone应用程序。

具体使用方法如下，将Mac OS X（X为具体的版本号）的安装DVD（只要购买了苹果电脑，都会附带安装DVD）放入光驱后，将会显示图1.1所示的界面，其中有名为“可选安装”的选项。



图1.1 Mac OS X的安装DVD