

Objective-C

程序设计入门与实践

刘一民 刘宪利 编著



源代码下载地址:

<http://www.tdpress.com/zyzx/tsscflwj>

Getting You the Best Book!

细致入微地讲述了Objective-C语言的基本知识和基础应用实践,帮助读者坚实地迈进iPhone/iPad开发门槛。

字里行间融入作者多年开发经验,语言风格平实活泼,范例讲解精炼到位,代码注释详略得当。

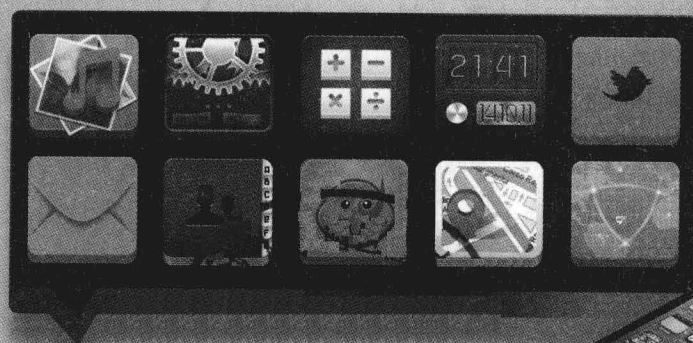


中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Objective-C

程序设计入门与实践

刘一民 刘宪利 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书详细介绍了 Objective-C 语言和苹果 iOS、Mac 平台面向对象程序编程的知识；结合独特的学习方法，在每章都合理嵌入大量的范例，让本书更适合于读者学习和课堂教学。

如果你对 iPhone/iPad 开发有兴趣，但是没有软件开发经验，本书将是你的首选，它能帮助你迈进开发的门槛，帮你成为一名初级的软件开发人员。对于用其他开发语言编程的人，本书也是一本不错的入门级别的书。如果想成为一名软件开发高手，仅仅靠本书的是不够的，建议掌握本书所介绍的知识之后，加入到苹果开发者中心 (<https://developer.apple.com.cn/>)，通过苹果官方提供的有关开发材料，来进一步学习，提高自己对 iPhone/iPad 的框架。

图书在版编目 (CIP) 数据

Objective-C 程序设计入门与实践 / 刘一民, 刘宪利
编著. —北京: 中国铁道出版社, 2013. 9
ISBN 978-7-113-16815-5

I. ①O… II. ①刘… ②刘… III. ①C 语言—程序设计
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 123865 号

书 名: Objective-C 程序设计入门与实践
作 者: 刘一民 刘宪利 编著

策 划: 荆 波
责任编辑: 张 丹
责任印制: 赵星辰

读者热线电话: 010-63560056

特邀编辑: 赵树刚

封面设计: 多宝格·付 巍

出版发行: 中国铁道出版社 (北京市西城区右安门西街 8 号 邮政编码: 100054)

印 刷: 三河市华丰印刷厂

版 次: 2013 年 9 月第 1 版 2013 年 9 月第 1 次印刷

开 本: 787 mm×1092 mm 1/16 印张: 26.5 字数: 611 千

书 号: ISBN 978-7-113-16815-5

定 价: 49.80 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社发行部联系调换。

我经常在探亲或者出差的车上，遇到那些曾经跟我类似的青年，特别是那些正要或者刚毕业的学生。在交流的时候，遇到最大的困惑，虽然他们有很强烈地加入到软件开发队伍中来的激情，但是由于在学校的学习主要是理论基础知识的夯实，造成他们所掌握的实践应用知识很有限，因此他们感觉自己没有掌握什么，没有什么东西可拿得出手，以至于他们很多人缺乏自信，特别是参加过几场人才招聘交流会后。同时，他们很多人认为从事软件行业的人都是“很厉害”的“主”。

如何加入软件开发团队？

难道加入软件开发团队真的很难吗？不难，只要你有梦想有激情，有强烈的学习欲望，有较强逻辑思维能力和一定的英语基础，热爱软件开发工作，同时能做到持之以恒，相信你不但能成为一名职业的软件开发工程师，而且还能成为一名优秀的软件开发工程师。其中如何判断自己逻辑思维能力是否强，其实很简单，只要你的数学学得不错，基本上就能证明你的逻辑思维能力不错。作为一名优秀的软件开发工程师，英语的阅读能力也是必不可少的，因为很多技术资料，特别是一些新兴的技术资料，往往都是用英语来写的，这就要求从事软件这个行业，需要有一定的英语阅读能力。在生活中，很多时候，不是某个行业把自己拒之门外，而是自己把这个行业拒之门外。

如果你对软件开发有兴趣，同时具备上述条件，如果你不想再靠父母的羽翼呵护——拥有自己的独立生活，那就加入到软件的开发团队，靠自己养活自己，乃至靠自己的能力组建自己的家庭，创建自己的事业。这个行业的收入相比较而言，还算是一个不错的；同时这个行业也是一个靠能力说话的行业。

同时，这个行业又是一个充满激情的行业，也许你就可能是中国未来的比尔·盖茨或者乔布斯。在我以前的同事中，就有两个人，一个初中毕业一个电大函授毕业，他们在这个行业都做的很不错，不比什么所谓的重点大学本科或者研究生差，他们都早已成为这个行业的技术骨干，在北京和上海等地方都置办有自己的产业。但是进入这个行业，企图靠“天上掉馅饼”，自己不努力，自己不好好自学，想“混”下去，也是很难立足的，很容易被淘汰。我提到这两个同事，是什么使他们能做的这么优秀？正是由于他们肯于上进肯于好学，他们为了提高自己的技术，狂补大量知识，特别是英语和高等数学的相关知识。

因此，不管你是将要毕业或者是已经毕业的，不管你是计算机专业毕业的还是非计算机专业毕业的，不管是大学毕业的还是中学毕业的，你都可以加入到软件开发团队中，只要肯努力肯上进，相信不久，一定会给你带来更加精彩的生活。

注意：对于高校在读的学生，建议最好有一个规划，早做一些准备，不管加入到软件行业团队还是其他行业团队，避免毕业了还要依赖父母养活自己，做“啃老族”，“断不了奶”。

为什么要学习 Objective-C?

1. 在最近几年中, Objective-C 是编程语言中应用增速最快的编程语言

2. Objective-C 作为 iPhone/iPad 框架的原生态语言

iPhone/iPad 框架选择了 Objective-C 作为 iPhone/iPad 框架的原生态语言有许多方面的原因。

- 首先,也是最主要的原因之一,它是一个面向对象的语言。iPhone/iPad 框架框架中的很多功能只能通过面向对象的技术来呈现。
- 其次,是标准 C 语言的一个超集,现存的 C 程序无须重新开发就能够使用 iPhone/iPad 软件框架,并且你可以在 Objective-C 中使用 C 的所有特性。可以选择什么时候采用面向对象的编程方式(例如定义一个新的类),什么时候使用传统的面向过程的编程方式(定义数据结构和函数而不是类)。
- 此外, Objective-C 是一个简洁的语言,它的语法简单,没有歧义,并且易于学习。因为易于混淆的术语以及抽象设计的重要性,对于初学者来说,面向对象编程的学习曲线比较陡峭。像 Objective-C 这种结构良好的语言使我们成为一名熟练的面向对象程序员更为容易。本书介绍 Objective-C 的章节也如同其语言本身一样简洁。
- 和其他的基于标准 C 语言的面向对象语言相比, Objective-C 对动态机制支持得更为彻底。编译器为运行环境保留了很多对象本身的数据信息,因此某些在编译时需要做出的选择就可以推迟到运行时来决定。这种特性使得基于 Objective-C 的程序非常灵活和强大。例如, Objective-C 的动态机制提供了两个一般面向对象语言很难提供的优点: Objective-C 支持开放式的动态绑定,从而有助于交互式用户接口架构的简单化。例如,在 Objective-C 中发送消息既无须考虑消息接收者的类也不用考虑方法的名字,从而可以允许用户在运行时再做出决定,也给了开发者在设计时极大的自由。Objective-C 的动态机制成就了各种复杂的开发工具。运行环境提供了访问运行中程序数据的接口,所以使得开发工具监控 Objective-C 程序成为可能。

3. iPhone/iPad SDK 用的是 Objective-C 开发语言

这样就可以更容易理解苹果的文档和使用 iPhone/iPad SDK 的功能。现今流行的诸如 Facebook Connect、OpenFeint 和 Cocos2D 等同样也是用 Objective-C 编写的,因此这些功能可以很容易地整合到自己的软件里面。

4. 作为编程语言来说, Objective-C 有很悠久的历史

它在 20 世纪 80 年代早期由 Stepstone 公司发明,作者是 Brad Cox 和 Tom Love。20 世纪 80 年代后期, NeXT 计算机有限责任公司获得了使用 Objective-C 来开发 NeXTStep 框架的授权,也就是后来的 Cocoa。NeXT 在多方面对 Objective-C 进行了扩展,例如协议部分。

注意:即使你已经掌握了其他的编程语言,我还是建议你学习 Objective-C。虽然我已经使用 Pascal、C++和 C#很多年,但是初次看到 Objective-C 的语法时,特别是 Objective-C 的函数定义方式总感觉怪怪的,有些不大喜欢这种已经很古老和过时的语言。但是再继续看下去,就会很容易被这种独特的语言风格所折服,同时也会感受到微软公司与苹果公司在“精致度”上的差距,回首展望,不得不佩服乔布斯的独特眼光。

本书适合哪些读者？

- 对软件开发有兴趣的人，特别是对 iPhone/iPad 开发有兴趣的人。
- 想成为一名软件开发人员，不管你是否是计算机专业毕业，不管你是什么学历，只有你有梦想，有激情。
- 在校的学生，适合把本书作为学习计算机软件开发的教程。
- 有一定的软件开发经验，想了解一些有关 iPhone/iPad 软件开发方面的知识。

你将会学到什么？

本书是一门 iPhone/iPad 开发入门级别的书，本书的最大目标，就是引导你如何入门，不管你是将要毕业或者已经毕业的，不管你是计算机专业毕业的还是非计算机专业的毕业的，不管是大学毕业还是中学毕业，只要你有梦想，有激情，通过本书你将会学到 iPhone/iPad 入门开发所需要掌握的基本知识，掌握这些知识，可以使你迈进 iPhone/iPad 开发那道门槛，当然成为一名 iPhone/iPad 软件开发高手，你还需要学习更多的知识，所以希望阅读本书完毕，继续深造自己，使自己不断“更上一层楼”！

你该如何阅读本书？（本书的章节安排）

本书的内容从形式上可以分为五部分，共 15 个章节。首先，将会介绍 iOS 平台开发的基础内容：其中包括了 iOS 平台开发技术、了解一下 iOS 的发展史、Objective-C 应用开发环境的搭建以及初次感受一下 iPhone 开发，经过这一部分使读者对有关 iPhone/iPad 开发有一个大致的了解，同时感受一下 iPhone 开发过程；其次，介绍 Objective-C 语言基础知识，这也是本书的基础点；然后，以面向对象的角度，同时结合 Objective-C 语言的特性，对类、消息和协议、对象以及基础（Foundation）框架继续讲解，来进一步了解 Objective-C，这一部分是本书的重点部分；然后，以性能的角度，从内存管理、文件和数据管理以及 UIKit 常用类及应用，来进一步深化对 Objective-C 介绍；最后，是实践环节，通过两个案例，把前面学到知识应用到实践中，使读者真正体验和了解到软件开发的过程。

在章节安排上，采取循序渐进的方式，使读者，特别是没有一点计算机软件开发经验的人，能够轻松阅读本书，尽可能减少理解上的障碍。如果你是有一定开发经验的软件人员，可以直接跳过第 4 章～第 6 章。

感恩致谢

在漫长的写作中，终于可以停下笔来，放松一下。我这个人，要是一件事情做不完，就很难放松下来。在这几个月中，写作主要靠下班之后或者周末来完成，身体有些超负荷运转，有时候想停下来，但是看着自己写出来一页又一页，自己又不免鼓励自己。在写作的过程，经常需要与审稿老师刘霞就写作的问题进行沟通，自己情绪低落的时候，不免会把自己低落情绪转加到她身上，所以这本书能坚持写完，在很大程度上，要感谢她的耐心和细心，使我在写作过程中，渡过一道又一道难关。

希望将此书献于我的宝宝和母亲！母亲远离故土，来帮我照看孩子和安排家中的生活起居，才能让我有时间静下心来写作。写累的时候，带宝宝在小区玩耍，我发现是一种有效的减缓疲劳的方式，同时又锻炼了我的身体，感谢他给我带来快乐，带来平淡中不时有惊喜的生活。

最后，感谢的就是你，我亲爱的读者，感谢你拿起这本书，你们的认可，就是我的最大的快乐。

刘一民

2013年6月

第 1 篇 认识 Objective-C 与 iPhone 开发

第 1 章 Objective-C、iOS、iPhone 和乔布斯

1.1	认识 Objective-C	1
1.1.1	发展历程及变化	1
1.1.2	语言性能与分析	2
1.2	iOS 简介	3
1.2.1	iOS 的发展历程	3
1.2.2	iOS 的设计和特性	4
1.3	iOS 架构	5
1.3.1	Cocoa Touch 层	6
1.3.2	Media (媒体) 层	8
1.3.3	Core Services 层	11
1.3.4	Core OS 层	13
1.4	iPhone 简介	14
1.4.1	iPhone 发展历程	14
1.4.2	iPhone 功能	16
1.4.3	iPhone 特色	16
1.4.4	iPhone 市场行情	18
1.5	iOS 开发者工具	18
1.5.1	Xcode	18
1.5.2	Interface Builder	20
1.5.3	Instruments	21
1.5.4	Shark	22
1.6	乔布斯简介	22
1.6.1	早年	22
1.6.2	创建苹果公司及发展	22
1.6.3	回归苹果及改革	23
1.6.4	个人成就	23

第2章 搭建 Objective-C 应用开发环境

2.1 基于 Windows 平台搭建开发环境	24
2.1.1 前期准备	24
2.1.2 安装虚拟机 VMware	25
2.1.3 安装 Mac OS	27
2.1.4 设置 Windows 和 MAC 共享目录	30
2.1.5 安装 iPhone SDK	31
2.1.6 启动 Xcode	32
2.2 进一步认识 Objective-C	32
2.2.1 类接口 (@Interface)	32
2.2.2 类实现 (@Implementation)	33
2.2.3 创建类对象	34
2.3 Xcode 的命令行工具	34
2.3.1 使用 Xcode 的命令行工具	34
2.3.2 输入和输出函数	37
2.3.3 指令符 (@)	38
2.4 进一步认识 Xcode	39
2.4.1 工程定义信息	40
2.4.2 源文件引用	42
2.4.3 目标的处理	43
2.4.4 执行环境	45

第3章 创建你的第一个 iPhone 应用程序

3.1 示例实现目标	46
3.2 应用概念概述	46
3.2.1 设计模式	47
3.2.2 委托	47
3.2.3 模型-视图-控制器模式	48
3.2.4 目标-动作	50
3.3 教你如何创建工程	50
3.3.1 使用 Xcode	51
3.3.2 应用程序引导	52
3.4 如何添加一个视图控制器	54
3.4.1 添加一个视图控制器类	54
3.4.2 添加一个视图控制器属性	55

3.4.3	创建视图控制器实例	56
3.4.4	如何建立视图	57
3.4.5	内务处理做些什么	57
3.4.6	实现源码列表	57
3.4.7	测试你的应用程序	58
3.5	查看 nib 文件	58
3.5.1	使用 Interface Builder	58
3.5.2	nib 文件包含些什么	59
3.5.3	文件的拥有者	59
3.5.4	如何使用视图插座变量	60
3.5.5	如何加载 nib 文件	61
3.5.6	测试应用程序	61
3.6	如何配置视图	62
3.6.1	添加用户接口元素	62
3.6.2	视图控制器接口声明	64
3.6.3	教你如何制定关联	65
3.6.4	测试	67
3.7	如何实现视图控制器	67
3.7.1	视图控制器中的属性	67
3.7.2	changeGreeting:方法	68
3.7.3	文本字段的委托	68
3.8	如何解决编译遇到的疑难	69
3.8.1	代码和编译器警告	69
3.8.2	检查 nib 文件中的关联	69
3.8.3	委托方法的名称	69
3.9	下一步做什么	70
3.9.1	用户接口	70
3.9.2	编程创建用户接口元素	70
3.9.3	如何把程序安装到设备	71
3.9.4	额外的功能	71

第 2 篇 Objective-C 的一点基础

第 4 章 Objective-C 的一点基础——数据类型和运算符

4.1	基本数据类型	72
4.1.1	整型	72

4.1.2	实型	75
4.1.3	字符型	78
4.1.4	字符串	80
4.1.5	枚举类型	80
4.2	其他常用类型	81
4.2.1	id 类型	81
4.2.2	布尔值 (BOOL)	83
4.2.3	typedef	85
4.2.4	nil 和 Nil	87
4.2.5	SEL 类型	87
4.3	指针和数组	89
4.3.1	数据和内存	89
4.3.2	指针	90
4.3.3	数组	94
4.3.4	数组和指针的关系	98
4.4	表达式和运算符	101
4.4.1	Objective-C 运算符	102
4.4.2	表达式和运算优先级	102
4.4.3	算术运算符	102
4.4.4	算术表达式	103
4.4.5	赋值运算符	104
4.4.6	关系运算符	105
4.4.7	逻辑运算符	105
4.4.8	增减运算符	106
4.4.9	强制类型转换运算符	107
4.4.10	位运算符	108

第 5 章 Objective-C 的一点基础——程序控制语句

5.1	条件语句	112
5.1.1	逻辑运算符	112
5.1.2	if 语句	112
5.1.3	switch 语句	115
5.1.4	条件运算符	117
5.2	循环结构	118
5.2.1	for 语句	118
5.2.2	while 语句	120

5.2.3	do-while 语句	120
5.2.4	嵌套语句	121
5.3	跳转语句	122
5.3.1	break 语句	122
5.3.2	continue 语句	122
5.3.3	return 语句	122
5.4	编译预处理	123
5.4.1	宏定义 (#define)	124
5.4.2	运算符	125
5.4.3	#import 语句	126
5.4.4	条件编译	127
5.5	几个常用特性	128
5.5.1	关联引用	128
5.5.2	快速枚举	130
5.5.3	选择器	131
5.5.4	静态类型的使用	133

第 6 章 Objective-C 的一点基础——算法

6.1	认识算法	137
6.1.1	什么是算法	138
6.1.2	算法有哪些特性	138
6.2	算法的表达方式	139
6.2.1	用自然语言表达算法	139
6.2.2	用流程图表达算法	139
6.2.3	用 N-S 流程图表达算法	141
6.2.4	用伪代码表达算法	142
6.2.5	用计算机语言表达算法	143
6.3	排序	144
6.3.1	简单排序	144
6.3.2	高级排序	149
6.3.3	其他排序	151
6.4	递归算法	153
6.4.1	认识递归算法	153
6.4.2	递归算法的简单应用	155

第3篇 充实你的 Objective-C 知识

第7章 充实你的 Objective-C 知识——类

7.1 认识根类	156
7.1.1 NSObject 简介	156
7.1.2 根类和协议	157
7.1.3 根类方法	157
7.1.4 根类接口规范	159
7.1.5 根类实例方法和类方法	159
7.2 如何设计类	159
7.2.1 设计接口 (@interface)	160
7.2.2 设计实现 (@implementation)	162
7.3 变量的种类	163
7.3.1 何为局部变量	164
7.3.2 何为全局变量	164
7.3.3 何为实例变量	164
7.3.4 何为静态 (static) 变量	167
7.3.5 变量的存储类别	168
7.4 认识属性	169
7.4.1 如何声明属性 (@property)	169
7.4.2 声明属性的特性	170
7.4.3 如何实现属性 (@synthesize)	171
7.4.4 属性类型和相关函数	173
7.4.5 认识属性的类型编码	173
7.4.6 如何实现属性重声明	174
7.4.7 如何修改父类的属性	174
7.4.8 新旧版本属性运行时的区别	175
7.5 探究方法	175
7.6 继承	176
7.6.1 如何继承父类的实例变量	177
7.6.2 如何继承父类的方法	177
7.6.3 如何重写 (override) 方法	178
7.6.4 如何重载 (overload) 方法	179
7.6.5 何时进行方法的重载	181
7.6.6 何时使用子类	183

7.6.7 类别 (Categories) 和延伸 (Extensions)	184
7.7 异常处理	187
7.7.1 如何启用异常处理	187
7.7.2 异常处理方式	188
7.7.3 如何捕捉不同类型的异常	188
7.7.4 如何抛出异常	189
7.8 如何使用类名	189
7.9 类怎样进行比较	190
7.10 类类型	190
7.10.1 如何指定静态类型	190
7.10.2 类型的自查处理机制	191

第 8 章 充实你的 Objective-C 知识——消息和协议

8.1 消息	192
8.1.1 认识消息语法	192
8.1.2 如何向 nil 发送消息	193
8.1.3 接收实例变量	194
8.1.4 如何获得方法地址	194
8.1.5 了解 objc_msgSend 函数	195
8.1.6 如何使用隐藏的参数	196
8.1.7 消息转发处理的机制	197
8.1.8 消息转发和多重继承	198
8.1.9 何为消息代理对象	199
8.1.10 消息转发和类继承	199
8.1.11 认识多态性	200
8.1.12 如何实现动态绑定	201
8.1.13 解析动态方法	201
8.1.14 如何使用动态加载	202
8.1.15 “.” 语法格式	202
8.2 协议	205
8.2.1 何为预定义接口	205
8.2.2 如何使用预定义方法	206
8.2.3 为匿名对象声明接口	206
8.2.4 非层级关系的相似 (Nonhierarchical Similarities)	207
8.2.5 何为正式协议	207
8.2.6 何为非正式协议	208

8.2.7 何为协议对象	209
8.2.8 如何采用一个协议	209
8.2.9 如何服从一个协议	210
8.2.10 类型校验处理机制	210
8.2.11 如何协议嵌套协议	211
8.2.12 引用其他协议	211

第9章 充实你的 Objective-C 知识——对象

9.1 如何创建对象	213
9.1.1 对象分配处理的机制	213
9.1.2 对象初始化	214
9.1.3 使用 dealloc 方法	219
9.1.4 何为类工厂方法	220
9.2 对象动态类型	221
9.3 探究对象的可变性	221
9.3.1 什么时候用可变或不可变对象	221
9.3.2 使用可变对象编程	222
9.4 对象间如何实现通信	225
9.4.1 面向对象程序中的通信有哪些	225
9.4.2 IBOutlet 变量如何实现通信	226
9.4.3 委托和数据源	227
9.4.4 可视化通信形式——目标-动作机制	230
9.4.5 特定通信形式——绑定	234
9.4.6 特定通信形式——通告	236
9.4.7 委托、观察者和目标的所有权	241
9.5 对象的生命周期	242
9.6 对象的所有权和销毁	245
9.6.1 对象所有权策略有哪些	245
9.6.2 保留计数的处理机制	246
9.6.3 何时使用自动释放	246
9.6.4 共享对象的有效性	248
9.6.5 如何使用存取方法	248
9.6.6 如何回收对象	249
9.6.7 通过引用返回的对象	250
9.6.8 保留循环的处理机制	250
9.6.9 何为对象的弱引用	251

9.6.10 如何有效地进行资源管理	251
--------------------------	-----

第 10 章 充实你的 Objective-C 知识——基础 (Foundation) 框架

10.1 Foundation 框架结构	253
10.2 与 Core Foundation 框架的区别	256
10.3 如何引用及查询 Foundation 框架信息	256
10.4 数字类型	259
10.4.1 数字类型类 (NSNumber)	259
10.4.2 int、NSInteger、NSUInteger 和 NSNumber 的区别及联系	260
10.5 字符串类型	260
10.5.1 不可变字符串类 (NSString)	261
10.5.2 可变字符串类 (NSMutableString)	263
10.6 数组类型	265
10.6.1 不可变数组类 (NSArray)	265
10.6.2 可变数组类 (NSMutableArray)	266
10.6.3 多维数组	268
10.7 字典 (Dictionary) 类型	271
10.7.1 不可变字典类 (NSDictionary)	272
10.7.2 可变字典类 (NSMutableDictionary)	272
10.8 日期 (Date) 类型	273
10.8.1 日期类型类 (NSDate)	274
10.8.2 NSCalendar 和 NSDateComponents	276
10.8.3 应用日期类型类	278

第 4 篇 提高你的 Objective-C 知识

第 11 章 提高你的 Objective-C 知识——内存管理

11.1 内存管理规则	280
11.2 基本的内存管理	281
11.2.1 普及内存管理基础知识	281
11.2.2 几个简单的内存管理应用	282
11.2.3 何为存取方法	282
11.2.4 如何应用存取方法	284
11.2.5 致使内存管理混乱的因素	287
11.3 何为自动释放池	288

11.3.1	非 Application Kit 程序中的自动释放池	289
11.3.2	自动释放池和线程	290
11.3.3	作用域和嵌套自动释放池的关系	290
11.3.4	何为所有权策略	291
11.3.5	如何实现垃圾回收	291
11.4	内存中复制的处理机制	292
11.4.1	使用深复制还是浅复制	292
11.4.2	何时使用 alloc 和 init 方式	293
11.4.3	何时使用 NSCopyObject	294
11.4.4	如何实现可变和不可变对象的复制	295
11.5	nib 对象的内存管理	296
11.5.1	插座 (IBoulet) 实现的内存管理机制	296
11.5.2	iNib 文件实现的内存管理机制	296

第 12 章 提高你的 Objective-C 知识——文件和数据管理

12.1	文件访问的指导原则	298
12.2	文件的管理	298
12.2.1	iPhone 常用目录	299
12.2.2	程序目录路径的获取	299
12.2.3	更新过程中的文件保存	300
12.2.4	文件的备份和恢复	301
12.3	数据的读/写	301
12.3.1	文件数据的读/写	301
12.3.2	属性列表数据的读/写	302
12.3.3	用归档器对数据进行读/写	303
12.3.4	将数据写到 Documents 目录	303
12.3.5	从 Documents 目录对数据进行读取	304
12.4	文件保存处理机制	304
12.5	SQLite 3	305
12.5.1	数据类型	306
12.5.2	常用函数及返回编码	306
12.5.3	数据库的创建、打开和关闭	309
12.5.4	表的创建	310
12.5.5	数据操作语句 (insert、update 和 delete)	310
12.5.6	数据库查询 (select)	311