

- 全面进阶首选
- 掌握Objective-C强大功能和特性
- 通过诸多实例快速学以致用



精通 Objective-C

【美】Keith Lee 著 苏宝龙 译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书



精通 Objective-C

[美] Keith Lee 著 苏宝龙 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

精通Objective-C / (美) 李 (Lee, K.) 著 ; 苏宝龙
译. — 北京 : 人民邮电出版社, 2015. 2
(图灵程序设计丛书)
ISBN 978-7-115-37884-2

I. ①精… II. ①李… ②苏… III. ①C语言—程序设
计 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第288823号

内 容 提 要

本书是 Objective-C 的高阶教程, 全面深入地介绍了 Objective-C 编程语言、Objective-C 的运行时环境和重要的 API。本书通过清晰易懂的方式介绍了 Objective-C 的关键概念, 还详细介绍了一些比较复杂的特性。书中还包括许多实用的示例(代码片段和完整程序), 让读者能快速学以致用。

本书适合有一定 Objective-C 开发经验的中高级开发者, 以及没有用过 Objective-C, 但拥有 C 语言编程经验并理解面向对象编程思想的开发者阅读。

-
- ◆ 著 [美] Keith Lee
 - 译 苏宝龙
 - 责任编辑 朱 巍
 - 责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 23.75
 - 字数: 567千字 2015年2月第1版
 - 印数: 1—3 500册 2015年2月北京第1次印刷
 - 著作权合同登记号 图字: 01-2013-8987号

定价: 79.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

版权声明

Original English language edition, entitled *Pro Objective-C* by Keith Lee, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2013 by Keith Lee. Simplified Chinese-language edition copyright © 2015 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

献给我的妻子 Dinavia，感谢你的爱与支持，以及对我的激励。也献给我那卷头发的孩子，你有着超乎年龄的智慧。

——Keith

致 谢

如果没有我妻子 Dinavia 的鼓励和理解，本书就不会存在。Dinavia 义无反顾地支持我的写作，无微不至地照顾我的生活。我永远都感谢你，Dinavia！

写一本书需要付出非常多的努力！我非常幸运，能够与 Apress 出版社的优秀团队合作。项目编辑 Douglas Pundick 和流程编辑 Mark Powers 帮助本书保持正确的轨道。技术审稿人 Felipe Laso Marsetti 仔细审读了各章的内容和代码。Kimberly Burton-Weisman 准时、高效地完成了本书的文字编辑工作。实际上，整个 Apress 编辑团队都为本书提供了许多帮助。衷心地感谢你们！

最后，我要感谢苹果公司，尤其是史蒂夫·乔布斯和史蒂夫·沃兹尼亚克，是他们为我们提供了这些优秀的技术。许多年前，在我还在上大学的时候，我购买了我的第一台计算机——Mac SE 30。从那时起，苹果公司的系统、硬件、软件和工程学就一直让我印象深刻。感谢这两位先驱的远见与坚持，使我得以写作此书！

引 言

Objective-C 编程语言变得越来越流行，其用途也变得越来越广泛。出现这种情况的原因是这种语言本身具有强大功能和易用性，以及该平台不断增加的大量新特性。许多程序员已经掌握了 Objective-C 的基础知识，并且想要进一步提升专业水平。本书就是一本带你登堂入室的进阶指南。

本书要点

本书全面深入地探讨了 Objective-C 编程语言、Objective-C 的运行时环境和重要的 API，以清晰易懂的方式介绍了 Objective-C 的关键概念，还详细介绍了一些比较复杂的特性。本书要点如下：

- Objective-C 的基本知识和重要的语言元素
- Objective-C 的运行时系统
- Foundation 框架 API
- Objective-C 语言的高级特性

本书还包括许多实用的示例（代码片段和完整程序），让你快速学以致用。

书中对每个主题的介绍清晰、准确，并适时地穿插一些有助于提高 Objective-C 编程效率的技巧。最为重要的一些特性都得到深入探讨，而且每一章都含有许多示例程序，以清晰地演示 Objective-C 的用法。

读者对象

本书主要面向有一定 Objective-C 开发经验的中高级开发者，也适合没有用过 Objective-C，但拥有 C 语言编程经验并理解面向对象编程思想的开发者阅读。

阅读前的准备工作

在为苹果公司的 OS X 和 iOS 平台编写 Objective-C 代码前，你需要拥有一台基于 Intel 的 Mac 电脑（MacBook、iMac、Mac Pro 等），并安装 OS X Mountain Lion（OS X 10.6.8）或更新版本的系统。你还在这台计算机上安装 Xcode，Xcode 是苹果公司为 iOS 和 Mac 软件开发提供的工具集。第 1 章会介绍如何获取和安装 Xcode。

本书主要内容

下面概括每章的主要内容。

第 1 章

本章介绍 Objective-C 编程语言和开发环境、如何下载和安装 Xcode，并编写第一个 Objective-C 程序。

第 2 章

本章通篇介绍类这个面向对象编程中的主要构件，还将介绍 Objective-C 中用于开发类的关键元素和独特功能。

第 3 章

在学习了如何创建类之后，本章会介绍如何使用它们。还会着重介绍创建和初始化 Objective-C 对象，以及在对象之间传递消息的概念和细节。

第 4 章

适当的内存管理方式是开发正确且高效的程序的关键。本章介绍 Objective-C 程序分配和释放计算机内存的方式、Objective-C 内存模型，以及如何编写以适当的方式执行内存管理的程序。

第 5 章

Objective-C 含有一个用于在编译前翻译源文件的预处理器。本章介绍预处理器的工作方式以及如何在 Objective-C 源文件中使用预处理器语言。

第 6 章

如前所述，ARC 是受推荐的 Objective-C 内存管理方式。本章是专家技巧章，会介绍更精细的 ARC 内存管理知识，以及对直接桥接对象应用 ARC 的方式。

第 7 章

本章开始介绍 Objective-C 运行时系统，将深入探讨 Objective-C 语言中的动态特性，包括运行时类型确定、方法决议、对象内省、动态代码加载等。

第 8 章

本章介绍 Objective-C 运行时系统的设计和架构，以及其动态特性的实现方式。本章还会介

绍在编译时和程序执行时，代码如何与运行时系统交互。

第 9 章

这个专家技巧章通过开发几个示例程序，增强你对运行时系统中的一些关键特性和 API 的理解。

第 10 章

本章介绍提供大多数 Objective-C 程序所需常用功能的 Foundation 框架类。这些类包括根类、字符串、值对象、集合、XML 数据处理和谓词。

第 11 章

本章介绍提供系统服务的 Foundation 框架类。使用这些类可以为网络、文件管理、进程间通信、系统信息检索、文本处理、线程化和并行处理提供各种操作系统服务。

第 12 章

本章介绍多个提供特定系统服务的 Foundation 框架类，这些服务实现支持事件驱动编程、对象永久化和分布式编程的功能。

第 13 章

本章介绍 Foundation 框架中的通用函数、数据类型和常数，这些 API 可以为 Objective-C 软件开发提供各种基本功能。

第 14 章

这个专家技巧章介绍错误处理。本章会介绍导致运行时错误的原因、处理错误的编程选项和用于处理错误和异常情况的 Foundation 框架 API。

第 15 章

本章介绍使用块语句编写程序的方式，块是 Objective-C 中强大的扩展功能。本章会介绍块语法和语义、块内存管理、在代码中编写块的方式和在现有 API 中使用块的方式。

第 16 章

常量是 Objective-C 中新增的功能。本章介绍常量的语法、语义和一般使用指南。

第 17 章

Objective-C 平台提供了各种语言扩展、API 和系统服务，这些元素专门用于确保你能够安全高效地进行并发编程。本章会详细介绍这些技术。

第 18 章

本章介绍键值编程（键值编码和键值观察）的基本知识。本章会介绍相关的实现细节，还会介绍键值编程的用法。

附录A

附录 A 简要总结 Objective-C 语言的基本元素，内容涵盖 Objective-C 语言对 ANSI C 的扩展。

附录B

附录 B 深入探讨 Xcode，其中包括 Xcode 的基本概念、主要功能元素及其关键特性的用法。

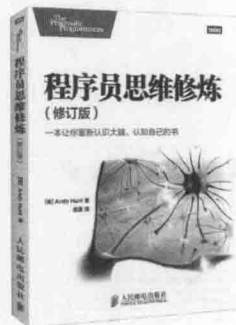
附录C

附录 C 详细介绍 Xcode 的调试器——LLDB，包括 LLDB 的设计和架构、LLDB 与 Xcode 整合的方式和在 Xcode 中使用 LLDB 调试程序的方式。



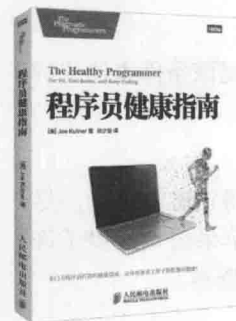
本书简明实用、见解深刻，总结了高效程序员在开发过程中的 45 个个人习惯、思想观念和方法，有助于开发人员在开发进程、编码工作、开发者态度、项目和团队管理以及持续学习这 5 个方面积极修炼。通过学习这些内容，养成这些好的习惯，你可以极大地提升自己的编程实力，更快速、更可靠地交付更高质量的软件，从而成为真正的高效程序员。

书名：高效程序员的 45 个习惯：敏捷开发修炼之道（修订版）
书号：978-7-115-37036-5
定价：45.00 元



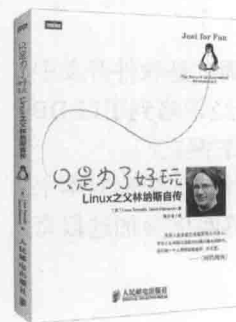
本书从认知科学、神经学、学习理论和行为理论角度，深入探讨了如何才能具备优秀的学习能力和思考能力，阐述了成为一名专家级程序员的关键要素，具体包括：大脑运行机制简介，如何正确使用和调试大脑，改进学习能力的具体技巧，如何通过自我引导积累经验，控制注意力的方法。为了让读者加深印象，作者还特别设立了一个“实践单元”，其中包括具体的练习和实验，旨在让读者真正掌握所学内容。

书名：程序员思维修炼（修订版）
书号：978-7-115-37493-6
定价：49.00 元



众所周知，程序员工作压力大、无暇照顾自己，去健身房更是奢侈行为。颈椎病、手腕疼痛、干眼症等问题困扰着几乎所有程序员和长期伏案的工作人员。在久坐不动的室内工作中，在没有大块时间去健身的客观条件下，如何保持身体健康？本书在科学研究和实践的基础上，总结了一些简单易行的健康指导，帮助读者从周身病痛、病弱且极有可能脾气暴躁的黑客，蜕变为快乐、高效的程序员。

书名：程序员健康指南
书号：978-7-115-36716-7
定价：39.00 元



2001 年 Linus 与人合著了他的自传，也是唯一一本关于他的传记。这部自传风趣幽默，充满极客风格，向读者展示了林纳斯如何用自己精彩的编程人生和对自由软件的热爱演绎现代社会中一个书呆子的胜利。

13 年后我们重新翻译出版这位最有影响力的发明家一生的传奇经历，期待与更多人一起探寻如何以“一切为了好玩”的态度对待技术、软件和人生……

书名：只是为了好玩：Linux 之父林纳斯自传
书号：978-7-115-36164-6
定价：49.00 元

目 录

第 1 章 入门.....1

1.1 简介.....1

1.1.1 苹果公司的 Objective-C 平台.....1

1.1.2 选择 Objective-C 的原因.....2

1.2 开发一个简单的 Objective-C 程序.....3

1.2.1 新建工程.....3

1.2.2 编写第一个 Objective-C 程序.....7

1.3 小结.....9

第 2 章 使用类.....10

2.1 开发第一个类.....10

2.1.1 向项目中添加一个类.....10

2.1.2 编写 Atom 类的接口.....12

2.1.3 编写实现 Atom 类的代码.....13

2.2 实例变量.....15

2.2.1 设置访问实例变量的方式.....16

2.2.2 声明实例变量.....16

2.2.3 访问实例变量.....17

2.3 属性.....17

2.3.1 属性声明.....17

2.3.2 属性定义.....18

2.3.3 由属性支持的实例变量.....19

2.3.4 访问属性.....19

2.4 方法.....20

2.4.1 语法.....20

2.4.2 调用方法.....21

2.5 协议.....21

2.6 分类.....24

2.7 小结.....27

第 3 章 对象和消息传递.....28

3.1 创建对象.....28

3.1.1 初始化对象.....29

3.1.2 延伸 Elements 工程.....30

3.2 发送消息.....35

3.3 消息转发.....36

3.3.1 转发选项.....37

3.3.2 向 Hydrogen 类添加快速转发 机制.....37

3.4 小结.....41

第 4 章 内存管理.....43

4.1 程序的内存使用情况.....43

4.2 Objective-C 的内存模型.....45

4.3 手动管理.....45

4.3.1 对象引用和对象所有权.....45

4.3.2 内存管理基本原则.....45

4.3.3 使用 MRR.....48

4.4 使用自动引用计数.....54

4.4.1 使用 ARC 的规则和约定.....55

4.4.2 ARC 的生命周期限定符.....56

4.4.3 使用 ARC.....56

4.4.4 避免循环引用.....60

4.5 小结.....60

第 5 章 预处理器.....62

5.1 概述.....62

5.2 预处理器语言.....63

5.2.1 指令.....64

5.2.2 宏.....71

5.3 小结.....72

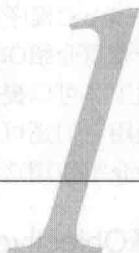
第 6 章 专家级技巧：使用 ARC	74
6.1 ARC 和对象所有权	74
6.1.1 声明对象的所有权	74
6.1.2 释放对象的所有权	75
6.2 扩展订单条目工程	76
6.3 将 ARC 与苹果公司提供的框架和服务一同使用	82
6.4 Objective-C 直接桥接	83
6.5 使用 ARC 桥接转换	85
6.6 小结	89
第 7 章 运行时系统	90
7.1 动态功能	90
7.2 对象消息	90
7.2.1 选择器	92
7.2.2 方法签名	93
7.2.3 使用对象消息	94
7.3 动态类型	100
7.4 动态绑定	101
7.5 动态方法决议	102
7.6 动态加载	104
7.7 内省	105
7.8 小结	106
第 8 章 运行时系统的结构	107
8.1 运行时系统的组成部分	107
8.1.1 编译器	107
8.1.2 运行时系统库	112
8.2 与运行时系统交互	120
8.2.1 NSObject 类的运行时方法	121
8.2.2 执行对象内省	121
8.3 小结	123
第 9 章 专家级技巧：使用运行时系统 API	125
9.1 使用可选包扩展程序	125
9.1.1 方法	125
9.1.2 步骤 1：编写基础代码	125
9.1.3 步骤 2：创建一个可选包	128
9.1.4 步骤 3：动态加载包	132
9.2 使用运行时系统 API	138
9.2.1 定义方法的实现函数	140
9.2.2 创建并注册类	140
9.2.3 创建类实例	141
9.2.4 以动态方式向类实例添加变量	141
9.3 创建动态代理	142
9.3.1 创建 Invoker 协议	143
9.3.2 编写代理类	144
9.3.3 测试 AspectProxy 程序	147
9.4 小结	149
第 10 章 Foundation 框架中的常用类	150
10.1 根类	150
10.1.1 NSObject 协议	150
10.1.2 NSObject 类	151
10.1.3 NSProxy 类	153
10.2 字符串	153
10.2.1 NSString 类	154
10.2.2 NSAttributedString 类	157
10.2.3 NSString 常量	157
10.2.4 格式化字符串	157
10.3 值对象	158
10.3.1 NSValue 类	158
10.3.2 NSNumber 类	158
10.3.3 NSDecimalNumber 类	159
10.3.4 NSNumber 常量	159
10.3.5 对日期和时间的支持	160
10.3.6 NSCache 类	160
10.4 集合	161
10.4.1 NSArray 类	161
10.4.2 NSPointerArray 类	162
10.4.3 NSDictionary 类	162
10.4.4 NSMapTable 类	163
10.4.5 NSSet 类	163
10.4.6 NSCountedSet 类	163
10.4.7 NSHashTable 类	163
10.4.8 NSPointerFunctions 类	164
10.5 XML 处理类	164
10.5.1 XML 的 DTD 处理类	164

10.5.2 基于树形结构的 XML 处理类	164	12.2 归档和序列化	192
10.5.3 事件驱动的 XML 处理类	165	12.2.1 归档	192
10.6 谓词类	165	12.2.2 属性列表序列化	193
10.7 小结	166	12.2.3 归档对象图	195
第 11 章 Foundation 框架中的系统服务	167	12.3 分布式对象	204
11.1 网络服务	167	12.4 脚本	205
11.1.1 NSHost 类	167	12.5 小结	205
11.1.2 Bonjour 网络服务	167	第 13 章 Foundation 框架函数和数据类型	206
11.2 应用服务	168	13.1 Foundation 框架函数	206
11.3 正则表达式和文本处理	168	13.1.1 断言	206
11.4 文件系统实用程序	169	13.1.2 记录日志	209
11.4.1 应用程序包	169	13.1.3 包	209
11.4.2 文件管理	170	13.1.4 十进制数值运算和字节排序	211
11.4.3 数据流的输入输出	170	13.1.5 与运行时系统进行交互	214
11.4.4 元数据查询	171	13.1.6 文件路径	214
11.5 并发机制和线程	171	13.1.7 几何图形	215
11.5.1 线程管理	171	13.2 数据类型	216
11.5.2 并行操作	172	13.3 常数	216
11.5.3 锁定	173	13.4 小结	216
11.5.4 计时器和运行循环	173	第 14 章 专家级技巧：错误处理	217
11.5.5 创建 Bonjour 网络服务客户端	174	14.1 运行时错误情况	217
11.6 URL 处理	178	14.1.1 断言	217
11.6.1 URL 加载	178	14.1.2 错误编码	218
11.6.2 缓存管理	180	14.1.3 错误对象	218
11.6.3 使用 URL 加载 API 下载资源	180	14.1.4 异常	218
11.6.4 身份认证和证书管理	185	14.2 NSError 类	218
11.6.5 Cookie 管理	186	14.2.1 使用错误对象	219
11.6.6 协议支持	186	14.2.2 处理委托方法错误	221
11.7 进程间通信	186	14.2.3 通过间接方式创建错误对象	225
11.7.1 通过信道通信	187	14.2.4 错误恢复	226
11.7.2 通过端口通信	187	14.2.5 错误回应对象	227
11.7.3 端口注册	187	14.2.6 NSError 编码	227
11.8 小结	188	14.3 异常处理	227
第 12 章 Foundation 框架的专用服务	189	14.4 NSException 类	228
12.1 通知	189	14.4.1 异常和内存管理	229
		14.4.2 执行异常处理	230

14.5	Foundation 框架标准异常名称	232	17.5	在 Objective-C 中实现并发编程	276
14.6	错误处理指导原则	233	17.6	语言特性	277
14.7	小结	234	17.7	消息传递	278
第 15 章	块	235	17.8	线程	279
15.1	块的语法	235	17.8.1	NSObject 线程	279
15.2	块就是闭包	239	17.8.2	NSThread	280
15.2.1	词汇范围	239	17.8.3	线程同步	281
15.2.2	可修改的 <code>_block</code> 变量	241	17.9	使用线程实现并发处理	284
15.3	块的内存管理	241	17.10	操作和操作队列	288
15.4	使用块	243	17.11	实现并发操作	291
15.4.1	使用块为数组排序	243	17.12	使用操作队列实现并发处理	295
15.4.2	使用块加载 URL	245	17.13	Grand Central Dispatch	298
15.4.3	使用块的并行编程方式	248	17.14	为并发编程选择合适的 API	301
15.5	小结	251	17.15	小结	301
第 16 章	Objective-C 常量	252	第 18 章	键值编程	303
16.1	常量	252	18.1	键值编码	303
16.2	NSNumber 常量	253	18.2	设计和实现 KVC	306
16.3	容器常量	254	18.2.1	键值编码 API	307
16.3.1	NSArray 常量	254	18.2.2	键值搜索模式	308
16.3.2	NSDictionary 常量	255	18.2.3	属性访问方法的命名约定	308
16.4	表达式常量	256	18.2.4	一对多属性访问方法命名约定	309
16.5	对象下标	257	18.2.5	键值检验	310
16.5.1	NSArray 对象的下标	257	18.2.6	键值编码的集合操作符	311
16.5.2	NSDictionary 对象的下标	258	18.3	键值观察	313
16.6	自定义下标	259	18.3.1	键值观察和通知	314
16.7	使用自定义下标编辑寄存器值	260	18.3.2	键值观察 API	314
16.8	小结	270	18.4	KVO 的设计和实现	316
第 17 章	并发编程	272	18.5	使用键值编程	316
17.1	并发编程的基本原则	272	18.6	小结	323
17.1.1	并发处理的优势	273	附录 A	语言元素	325
17.1.2	实现并发处理	274	附录 B	Xcode 的配置功能	337
17.2	并行处理带来的挑战	274	附录 C	使用 LLDB	353
17.3	共享数据	275			
17.4	消息传递	276			

第 1 章

入 门



欢迎Objective-C新手们随我一起踏上这次Objective-C之旅！本章首先简要介绍这门语言，然后就开始编写代码。我们首先介绍苹果公司的Objective-C开发环境，然后探讨Objective-C在应用开发方面如此流行的原因，随后会介绍苹果公司提供的集成开发环境（IDE）Xcode，并研究如何用它来提高Objective-C编程的效率和增加编程乐趣。

1.1 简介

Objective-C是苹果公司的OS X和iOS（iPod、iPhone和iPad）平台上的主要编程语言。近年来，这两大平台已经位于最流行的应用开发环境之列，而其成功的一个关键因素在于Objective-C的特性。

苹果公司在2007年发布了Objective-C 2.0。该版本新增了许多特性，其中包括自动声明和补全属性、点语法、快速枚举、异常处理、运行时性能提升和支持64位机。

Objective-C语言一直在不断发展，并通过新增的特性获得更强大的编程和表达能力。该语言最近的一些较大改进包括：Objective-C对象的自动引用计数特性、改进的数据隐藏特性、改进的枚举类型安全性，以及块、常量和和其他特性的新语言结构。

1.1.1 苹果公司的Objective-C平台

苹果公司的Objective-C开发环境由以下四部分组成：

- Objective-C编程语言
- Objective-C运行时环境
- 软件库
- 软件开发工具

本书主要介绍使用Objective-C进行面向对象的软件开发方法。第一部分介绍这门编程语言和该语言对面向对象程序设计的支持。

Objective-C程序在Objective-C运行时环境中执行，因此可动态编程。本书第二部分详细介绍Objective-C的运行时环境，以及使用其应用编程接口（API）的方法。

软件库中含有一系列多功能的框架、库和服务，使用它们可以简化应用开发工作。此外，软

件库还提供了在OS X和iOS平台上开发应用所需的许多功能。本书第三部分介绍Foundation框架，它是各种Objective-C程序都使用的基础API。

第四部分着重介绍Objective-C的高级特性，使用它们可以开发更为精巧的应用。

软件开发工具可以提供源代码编辑和编译、用户界面开发、版本控制、项目管理、测试与调试等功能。使用它们还可以简化应用开发工作，提高开发、管理和维护Objective-C软件的效率。本书通篇都会介绍使用这些工具开发程序的方法。附录B给出了一些额外的提示和建议。

1.1.2 选择Objective-C的原因

与当今的诸多编程语言相比，Objective-C具有哪些优势呢？毕竟，支持面向对象程序设计的语言相当多。Objective-C是苹果公司OS X和iOS平台上的主要编程语言，这是不是它流行起来的最大原因呢？Objective-C原本就是一款非常优秀的编程语言，它拥有许多特性，在应用开发过程中功能强大、能够处理各种情况并且易于使用。

- ❑ 面向对象的编程功能：Objective-C编程语言为面向对象的程序设计（OOP）提供了全面的支持，其中包括对象消息、封装、继承、多态和开放式递归等功能。
- ❑ 对象消息传递：该功能使对象能够通过彼此传递消息协同工作。实际上，Objective-C代码（如类、对象方法或函数）会向接收对象（接收器）发送消息，然后接收器会使用该消息调用相应的方法，并在有需要时返回结果。如果接收器没有相应的方法，也可以使用其他方式处理该消息，如将其发送给另一个对象、向其他对象广播该消息、检查该消息并应用自定义逻辑等。
- ❑ 动态的运行时环境：与许多面向对象的编程语言相比，Objective-C拥有非常多的动态特性。它将许多处理类型、消息和方法决议（method resolution）的工作转移到运行程序的时候进行，而不是在编译或链接时处理。使用这些功能能够以实时方式，同时促进程序的开发和更新，而无需重新编译和部署软件，而且随着时间的推移，这样做对现有软件的影响最小甚至没有影响。
- ❑ 内存管理：Objective-C提供了内存管理功能——自动引用计数（ARC），使用该功能既可以简化应用开发过程，又可以提高应用的性能。ARC是一种编译时技术，它整合了传统内存自动化管理机制（如垃圾回收器）的许多优点。然而，与传统技术相比，ARC可以提供更好的性能（内存管理代码会在编译时被插入到程序代码中），因而不会在执行程序时引起由内存管理原因导致的暂停。
- ❑ 内部检查和获取信息：通过Objective-C语言提供的功能，程序能够在运行时检查对象、获取信息（对象的类型、属性和该对象支持的方法），以及修改对象的结构和行为。这样就可以在执行程序时修改程序。
- ❑ 对C语言的支持：Objective-C实际上是C语言面向对象编程的扩展。所以，它是C语言的超集。这意味着Objective-C程序中使用未经修改的原始C语言代码，而且Objective-C程序也可以直接访问C语言标准函数库。

□ 苹果公司的技术：苹果公司为Objective-C应用开发提供了丰富的软件库和工具。这些开发工具拥有含基础设施的框架和库，让你可以集中精力开发应用。Xcode是苹果公司提供的集成开发环境，提供了使用Objective-C开发应用所需的所有工具。

上面仅是Objective-C越来越流行的部分原因，我可以肯定你在阅读本书时还会发现更多的原因。下面让我们使用Xcode编写一个简单的Objective-C程序，看看Objective-C的真正实力。

1.2 开发一个简单的 Objective-C 程序

精通一门编程语言的最佳方法是用它编写程序，因此下面我们就要编写一些代码。但是在此之前，需要先下载并安装Xcode。

Xcode是用于在Mac上进行Objective-C开发的完整IDE，它不仅彻底整合了iOS和OS X，而且还含有编写和编译源代码、开发精妙的用户界面、测试和调试软件、发布和管理版本、管理项目以及实现许多其他特性所需的所有工具。苹果公司的iOS Developer Program和Mac Developer Program都为会员免费提供Xcode下载。如果你不是其中任何一个的会员，也可以从Mac App Store免费下载Xcode。本书使用Xcode 4.5，它是本书写作时的最新版本。Xcode 4.5可以在安装了OS X Lion或更高版本Mac OS的任何Intel Mac上运行。

1.2.1 新建工程

下载并安装后，启动Xcode。屏幕会显示如图1-1所示的Xcode欢迎窗口。

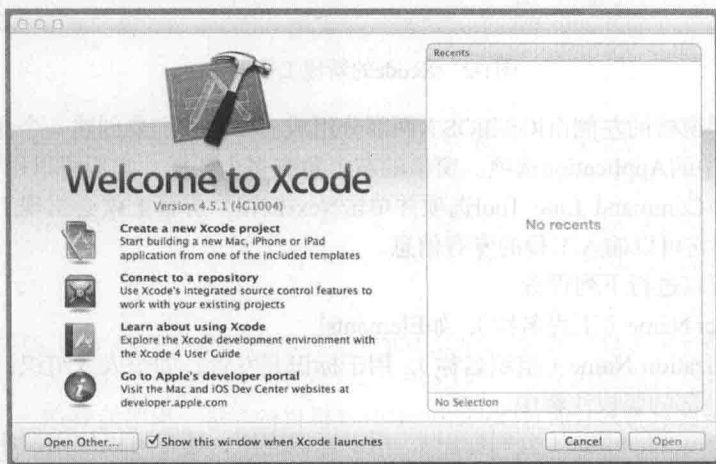


图1-1 Xcode欢迎窗口

注意 如果你的计算机上连接了iOS设备（如iPhone、iPod、iPad），系统会显示消息，询问你是否使用该设备进行开发工作。这里不是开发移动应用，因此单击Ignore按钮。