



开发人员专业技术丛书

用于Mac OS X和iPhone开发的Objective-C 2.0语言的完全指南

Programming in Objective-C 2.0

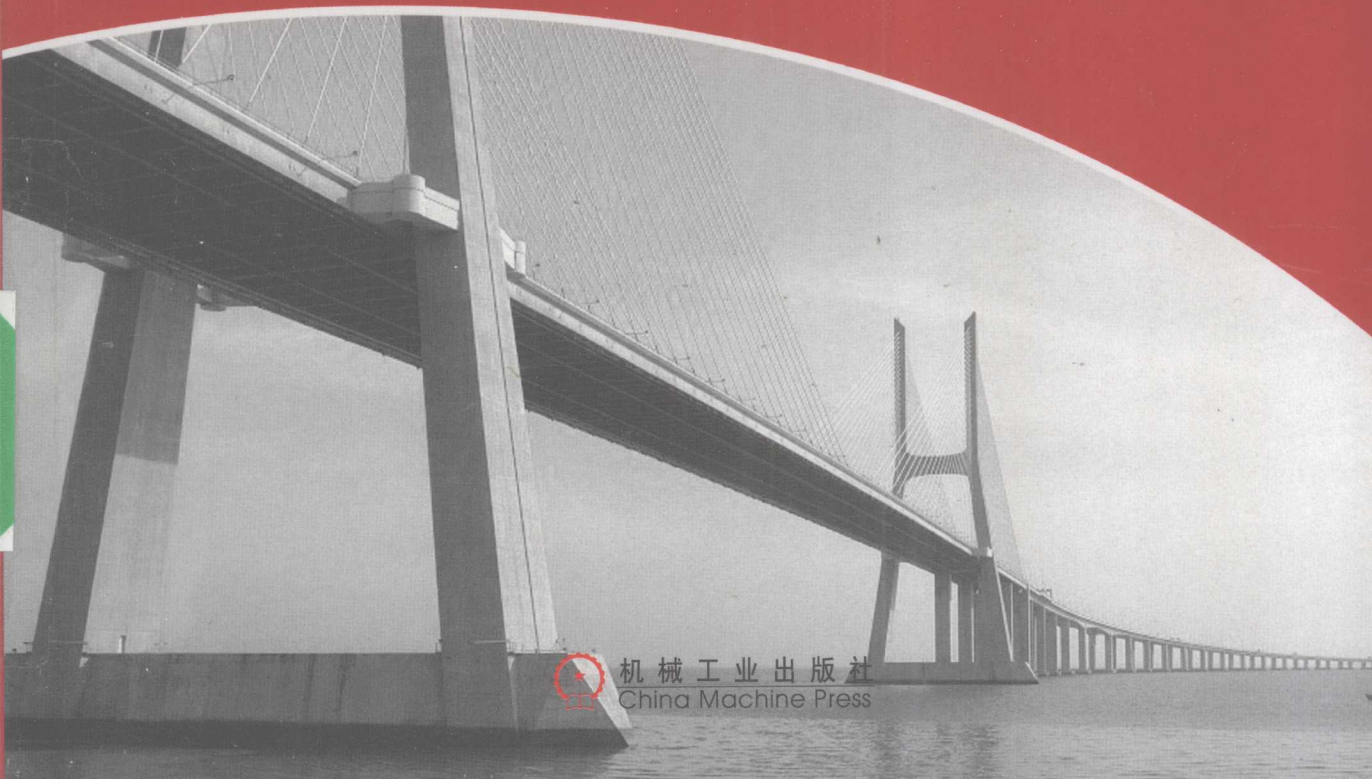
Second Edition

Objective-C 2.0 程序设计

(原书第2版)

(美) Stephen G. Kochan 著

张波 黄湘琴 等译



机械工业出版社
China Machine Press

开发人员专业技术丛书

Programming in Objective-C 2.0

Second Edition

Objective-C 2.0 程序设计

(原书第2版)

(美) Stephen G. Kochan 著

张波 黄湘琴 等译



机械工业出版社
China Machine Press

本书是Objective-C领域的经典著作，对Objective-C进行了系统而全面的阐述，权威性毋庸置疑。全书共分为四大部分：第一部分全面讲解了Objective-C语言的基础知识，包括类、对象、方法、数据类型、表达式、程序结构、继承、多态、动态类型和动态绑定、函数、数组、结构和指针等；第二部分详细阐述了Foundation框架，涵盖数字、字符串、集合、文件操作、内存管理、对象复制和归档等重要内容；第三部分简要介绍了Cocoa和iPhone SDK，第四部分是附录，主要列出了Objective-C的快速参考。

本书结构合理、内容翔实，简单易学，既适合程序设计初学者和所有程序员阅读，也可作为程序设计语言课程的入门教材。

Simplified Chinese edition copyright © 2009 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Programming in Objective-C 2.0* (ISBN 978-0-321-56615-7) by Stephen G. Kochan, Copyright © 2009.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as AddisonWesley.

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2009-3515

图书在版编目（CIP）数据

Objective-C 2.0程序设计/（美）科施恩（Kochan, S. G.）著，张波等译. —北京：机械工业出版社，2009.9

（开发人员专业技术丛书）

书名原文：Programming in Objective-C 2.0

ISBN 978-7-111-27686-9

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

中国版本图书馆CIP数据核字（2009）第117380号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：陈佳媛

北京京北印刷有限公司印刷

2009年9月第1版第1次印刷

186mm × 240mm · 29.5印张

标准书号：ISBN 978-7-111-27686-9

定价：66.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换
本社购书热线：（010）68326294

译者序

Objective-C语言是由Brad J.Cox于20世纪80年代早期设计的，它是一种简单的面向对象编程语言，它的基础是C程序设计语言。Objective-C不仅功能强大，而且简单易学。Objective-C已经成为Mac OS X系统下应用程序开发的标准程序设计语言，除此之外，它广泛用于多种支持gcc编译器的操作系统，包括LINUX、UNIX和Windows系统。

本书清晰地解释了Objective-C和OOP的基本概念，示例丰富。第二部分更进一步深化了本书的主题。对于任何想要学习编写Objective-C的人来说，不管你是否了解C和OOP，本书都是很好的读物。本书从最基本的内容开始讲起，阅读本书无需任何编程经验。如果你已经熟悉了C，那么可以跳过基础部分，直接阅读你感兴趣的部分。如果你是编程高手，可以从第二部分中受益。每章后面都有练习，因此本书也可作为教材。

本书共包括三个逻辑部分和四个附录。第一部分是“Objective-C语言”，讲述该语言的基础知识。内容包括：使用Objective-C编写第一个程序；类、对象和方法；数据类型和表达式；循环结构；选择结构；类和对象的使用；继承；多态、动态类型和动态绑定；函数、数组、结构和指针。第二部分“Foundation框架”，讲述如何使用Foundation框架的各种预定义类。该部分首先介绍Foundation框架以及名为NSObject的根对象，然后讲述了Foundation框架的一些重要特性，包括数字和字符串对象的处理、集合、文件系统处理、内存管理以及对象的复制和归档。第三部分介绍了Cocoa和iPhone SDK。最后，本书在附录A中提供了一些常用术语的定义。附录B对Objective-C语言进行了总结，用于快速参考。附录C总结了Foundation框架头文件。附录D提供了一些学习资源。

本书内容由浅入深、结构清晰、步骤明确、简单易学，特别适合于程序设计初学者学习。本书也适合于Objective-C程序员阅读和参考之用，还可作为大学程序设计语言的入门教材。相信你通过本书的学习，能够轻松驾驭Objective-C语言。编程路上好运！

本书由张波、黄湘琴主译，谢君英审校。参与本书翻译工作的还有欧阳宇、王桂英、李长青、杨中民、唐美艳、李秋霞、盛海燕、郭蓓、代菊容、赵岗善、易磊等，在此一并感谢。翻译过程中译者尽心尽力，但限于时间和水平，错误在所难免，请读者谅解并指正！

译者

2009年4月28日

作者简介

Stephen Kochan是多本畅销书的作者或合著者，其中有关于C语言的，包括《Programming in C》(Sams, 2004)、《Programming in ANSI C》(Sams, 1994)和《Topics in C Programming》(Wiley, 1991)，也有关于UNIX的，包括《Exploring the Unix System》(Sams, 1992)和《UNIX Shell Programming》(Sams, 2003)。从1984年Mac最初引入时，他就已经在Macintosh计算机上进行程序设计了，他编写的《Programming C for the Mac》是Apple Press Library的一部分。2003年，Kochan编写了《Programming in Objective-C》(Sams, 2003)，之后编写了另一本与Mac有关的书籍《Beginning AppleScript》(Wiley, 2004)。

技术审校人员简介

Michael Trent从1997年开始就使用Objective-C进行程序设计，之前就已经开始在Mac上进行程序设计。他定期为Steven Frank的www.cocoadev.com的网站供稿，为大量的书籍和杂志文章进行了技术审校，并且偶尔也涉足Mac OS X开源项目。目前，他正在使用Objective-C和Apple Computer的Cocoa框架生成Mac OS X上使用的专业视频应用程序。Michael拥有比洛特学院（位于威斯康星州比洛特）的计算机科学学士学位和音乐艺术学位。他与妻子Angela居住在加利福尼亚州的圣克拉拉。

目 录

译者序	
作者简介	
技术审校人员简介	

第1章 前言	1
1.1 本书的内容	2
1.2 本书的组织方式	2
1.3 致谢	4

第一部分 Objective-C语言

第2章 Objective-C程序设计	5
2.1 编译并运行程序	6
2.1.1 使用Xcode	6
2.1.2 使用Terminal	10
2.2 解释第一个程序	12
2.3 显示变量的值	15
2.4 小结	17
2.5 练习	17
第3章 类、对象和方法	20
3.1 到底什么是对象	20
3.2 实例和方法	20
3.3 用于处理分数的Objective-C类	22
3.4 @interface部分	25
3.4.1 选择名称	25
3.4.2 实例变量	26
3.4.3 类和实例方法	27
3.5 @implementation部分	28
3.6 Program部分	29
3.7 实例变量的访问以及数据封装	34
3.8 小结	37
3.9 练习	37

第4章 数据类型和表达式	39
4.1 数据类型和常量	39
4.1.1 int类型	39
4.1.2 float类型	40
4.1.3 double类型	40
4.1.4 char类型	41
4.1.5 限定词: long、long long、short、 unsigned及signed	42
4.1.6 id类型	43
4.2 算术表达式	44
4.2.1 运算符的优先级	44
4.2.2 整数运算和一元负号运算符	47
4.2.3 模运算符	48
4.2.4 整型值和浮点值的相互转换	49
4.2.5 类型转换运算符	51
4.3 赋值运算符	51
4.4 计算器类	52
4.5 位运算符	54
4.5.1 按位与运算符	55
4.5.2 按位或运算符	56
4.5.3 按位异或运算符	56
4.5.4 一次求反运算符	57
4.5.5 向左移位运算符	58
4.5.6 向右移位运算符	59
4.6 类型: _Bool、_Complex和_Imaginary	59
4.7 练习	59
第5章 循环结构	62
5.1 for语句	63
5.1.1 键盘输入	68
5.1.2 嵌套的for循环	69
5.1.3 for循环的变形	71

5.2 while语句	72	8.3.1 择哪个方法	145
5.3 do语句	76	8.3.2 重载dealloc方法和关键字super	147
5.4 break语句	77	8.4 通过继承扩展: 添加新的实例变量	149
5.5 continue语句	78	8.5 抽象类	151
5.6 小结	78	8.6 练习	151
5.7 练习	78	第9章 多态、动态类型和动态绑定	153
第6章 选择结构	79	9.1 多态: 相同的名称, 不同的类	153
6.1 if语句	79	9.2 动态绑定和id类型	156
6.1.1 if-else结构	83	9.3 编译时和运行时检查	158
6.1.2 复合条件测试	86	9.4 id数据类型与静态类型	159
6.1.3 嵌套的if语句	88	9.5 有关类的问题	160
6.1.4 else if结构	90	9.6 使用@try处理异常	164
6.2 switch语句	98	9.7 练习	166
6.3 Boolean变量	101	第10章 变量和数据类型	168
6.4 条件运算符	105	10.1 类的初始化	168
6.5 练习	106	10.2 作用域回顾	170
第7章 类	108	10.2.1 控制实例变量作用域的指令	170
7.1 分离接口和实现文件	108	10.2.2 外部变量	171
7.2 合成存取器方法	113	10.2.3 静态变量	172
7.3 使用点运算符访问属性	114	10.3 存储类说明符	175
7.4 具有多个参数的方法	115	10.3.1 auto	175
7.4.1 不带参数名的方法	117	10.3.2 const	175
7.4.2 关于分数的操作	117	10.3.3 volatile	175
7.5 局部变量	119	10.4 枚举数据类型	176
7.5.1 方法的参数	120	10.5 typedef语句	179
7.5.2 static关键字	120	10.6 数据类型转换	180
7.6 self关键字	122	10.6.1 转换规则	180
7.7 在方法中分配和返回对象	122	10.6.2 符号扩展	181
7.8 练习	127	10.7 练习	182
第8章 继承	129	第11章 分类和协议	183
8.1 一切从根类开始	129	11.1 分类	183
8.2 通过继承扩展——添加新方法	132	11.2 协议	188
8.2.1 Point类和内存分配	136	11.3 合成对象	191
8.2.2 @class指令	137	11.4 练习	192
8.2.3 具有对象的类	141	第12章 预处理程序	194
8.3 重载方法	143	12.1 #define语句	194

12.1.1 更高级的定义类型	195	13.7.2 goto语句	249
12.1.2 #运算符	199	13.7.3 空语句	249
12.1.3 ##运算符	200	13.7.4 逗号运算符	249
12.2 #import语句	201	13.7.5 sizeof运算符	250
12.3 条件编译	203	13.7.6 命令行参数	250
12.3.1 #ifdef、#endif、#else和#ifndef 语句	203	13.8 工作原理	252
12.3.2 #if和#elif预处理程序语句	204	事实#1: 实例变量存储在结构中	253
12.3.3 #undef语句	205	事实#2: 对象变量实际上是指针	253
12.4 练习	205	事实#3: 方法是函数, 而消息表达式是 函数调用	253
第13章 基本的C语言特性	207	事实#4: id类型是通用指针类型	254
13.1 数组	207	13.9 练习	254
13.1.1 数组元素的初始化	209		
13.1.2 字符数组	210	第二部分 Foundation框架	
13.1.3 多维数组	211	第14章 Foundation框架简介	257
13.2 函数	212	第15章 数字、字符串和集合	260
13.2.1 参数和局部变量	214	15.1 数字对象	260
13.2.2 函数的返回结果	215	15.2 字符串对象	264
13.2.3 函数、方法和数组	218	15.2.1 NSLog函数	265
13.3 结构	219	15.2.2 可变对象与不可变对象	266
13.3.1 结构的初始化	225	15.2.3 可变字符串	270
13.3.2 结构数组	225	15.2.4 所有对象到哪里去了	274
13.3.3 结构中的结构	226	15.3 数组对象	276
13.3.4 关于结构的补充细节	227	15.4 同步AddressCard方法	284
13.3.5 不要忘记面向对象编程思想	229	15.4.1 快速枚举	288
13.4 指针	230	15.4.2 数组排序	296
13.4.1 指针和结构	233	15.5 词典对象	299
13.4.2 指针、方法和函数	234	15.6 集合对象	302
13.4.3 指针和数组	236	15.7 练习	306
13.4.4 指针运算	244	第16章 使用文件	308
13.4.5 指针和内存地址	245	16.1 管理文件和目录: NSFileManager	308
13.5 联合	246	16.1.1 使用NSData类	313
13.6 它们不是对象	248	16.1.2 使用目录	314
13.7 其他语言特性	248	16.1.3 枚举目录中的内容	316
13.7.1 Compound Literal	248	16.2 使用路径: NSPathUtilities.h	318

16.2.1 常用的路径处理方法	321
16.2.2 复制文件和使用NSProcessInfo类	321
16.3 基本的文件操作: NSFileHandle	325
16.4 练习	329
第17章 内存管理	331
17.1 自动释放池	331
17.2 引用计数	332
17.2.1 引用计数和字符串	334
17.2.2 引用计数与实例变量	337
17.3 自动释放池示例	342
17.4 内存管理规则摘要	344
17.5 垃圾回收	344
17.6 练习	345
第18章 复制对象	347
18.1 copy和mutableCopy方法	347
18.2 浅复制与深复制	350
18.3 实现<NSCopying>协议	352
18.4 用赋值方法和取值方法复制对象	354
18.5 练习	356
第19章 归档	357
19.1 使用XML属性列表进行归档	357
19.2 使用NSKeyedArchiver归档	359
19.3 编码方法和解码方法	361
19.4 使用NSData创建自定义档案	367
19.5 使用归档程序复制对象	371
19.6 练习	372

第三部分 Cocoa和iPhone SDK

第20章 Cocoa简介	373
20.1 框架层	373
20.2 接触Cocoa	374
第21章 编写iPhone应用程序	375
21.1 iPhone SDK	375
21.2 第一个iPhone应用程序	375
21.2.1 创建新的iPhone应用程序项目	376
21.2.2 输入代码	378
21.2.3 设计界面	380
21.3 iPhone分数计算器	386
21.3.1 启动新的Fraction_Calculator 项目	387
21.3.2 定义视图控制器	388
21.3.3 Fraction类	393
21.3.4 处理分数的Calculator类	397
21.3.5 设计UI	398
21.4 小结	399
21.5 练习	400

第四部分 附录

附录A 术语表	403
附录B Objective-C 2.0语言概览	407
附录C 地址簿源代码	453
附录D 资源	458

第 1 章 前 言

C程序设计语言是由AT&T贝尔实验室的Dennis Ritchie于20世纪70年代早期首创的。但是，直到70年代晚期，这种程序设计语言才获得了广泛的支持并流行开来。这是因为，在此之前，C编译器还不能用于贝尔实验室以外的商业用途。最初，UNIX操作系统同样的普及速度也在某种程度上促进了C语言的快速普及，UNIX操作系统几乎完全是由C语言编写的。

Objective-C语言是由Brad J.Cox于20世纪80年代早期设计的，它以一种叫做SmallTalk-80的语言为基础。Objective-C建立在C语言之上，意味着它是在C语言基础添加了扩展而创造出来的能够创建和操作对象的一门新的程序设计语言。

1988年，NeXT Software公司获得了Objective-C语言的授权，并开发出了Objective-C的语言库和一个名为NEXTSTEP的开发环境。1992年，自由软件基金会的GNU开发环境增加了对Objective-C的支持。所有Free Software Foundation (FSF) 产品的版权归FSF所有。它根据GNU通用公共许可证发布产品。。

1994年，NeXT Computer公司和Sun公司联合发布了一个针对NEXTSTEP系统的标准规范，名为OPENSTEP。OPENSTEP在自由软件基金会的实现名称为GNUStep。有一个Linux版本，它包括Linux内核和GNUStep开发环境，这个Linux发行版被十分贴切地命名为LinuxSTEP。

1996年12月20日苹果公司宣布收购NeXT Software公司，NEXTSTEP/OPENSTEP环境将成为苹果操作系统下一个主要发行版本OS X的基础。这个开发环境的该版本被苹果公司称为Cocoa。它内置了对Objective-C语言的支持，并结合了Project Builder（或它的后继版本Xcode）和Interface Builder等开发工具，苹果公司为Mac OS X上的应用程序开发创建了一个功能强大的开发环境。

2007年，苹果公司发布了Objective-C语言的更新，并称之为Objective-C 2.0。本书即是基于该版本的。

iPhone于2007年发布时，开发人员们要求为这款革新性的设备开发应用程序。起初，苹果公司不欢迎第三方应用程序开发。苹果公司安抚那些超级崇拜iPhone的开发人员的办法是：允许他们开发基于Web的应用。这些基于Web的应用在iPhone内置的Safari Web浏览器下运行，但需要用户连接到托管该应用程序的网站。开发人员对基于Web的应用的很多固有限制非常不满，于是苹果公司不久之后就宣布，开发人员能够为iPhone开发所谓的本机应用。

本机应用是驻留在iPhone上并且在iPhone操作系统下运行的应用，其运行方式与该设备上运行的内置iPhone应用（如Contacts、iPod和Weather）相同。iPhone的操作系统实际上是某个Mac OS X版本，这意味着可以在MacBook Pro上开发并调试这些应用。实际上，苹果公司很快就提供了强大的软件开发套件（SDK），允许快速地开发iPhone应用并进行调试。iPhone模拟器使得开发人员直接在其开发系统上调试应用成为可能，无需在实际的iPhone或iPod Touch设备上下下载并测试程序。

1.1 本书的内容

在计划编写这本有关Objective-C的教程时，我做出了一项重要决定。即在讲述Objective-C的内容之前，假定读者已经知道如何编写C语言程序。我将从使用丰富的例程库（例如Foundation框架和Application Kit框架）的角度讲解这门语言。我还将讲解如何使用一些开发工具（如Mac的Xcode和Interface Builder等）。

选用这种方法有以下几个问题：首先，学习Objective-C之前必须完整地学习C语言，这种说法是错误的。C语言是一门过程性的语言，有很多特性是在使用Objective-C进行程序设计时不必要的，特别是对于初学者。事实上，采用其中的某些特性违反了坚持良好的面向对象的程序设计方法的本质。同样，在学习面向对象编程语言之前，最好不要了解过程性语言的所有细节。这会导致程序员误入歧途，并在养成良好的面向对象的程序设计风格方面，造成错误的导向和思维定式。Objective-C仅仅只是C语言的扩展，并不意味着必须首先学习C语言！

因此，我决定既不首先讲述C，也不事先假定你具备了该语言的知识。相反，我决定采用一种非常规的方式，从面向对象编程的视角出发，将Objective-C和基础的C语言作为一门单独的集成语言来讲解。顾名思义，本书的目的是教你如何使用Objective-C 2.0进行程序设计。这并不表示我会详细介绍可用于开发和调试程序的开发工具，或者讲解如何使用Cocoa开发交互式图形应用。学会如何使用Objective-C编写程序后，所有这些资料都可在其他地方获得。事实上，在具备了如何使用Objective-C进行程序设计的坚实基础后，掌握这些知识是轻而易举的。本书并不假设需要编程经验，即使有，也不会很多。如果你是一名程序设计的初学者，应该可以将Objective-C作为第一门程序设计语言。

本书以示例的方式来讲述Objective-C语言。在介绍这门语言的每个新特性时，通常会提供一个完整的小例子来阐述这一特性。正如一图胜千言一样，一个经过严格筛选的例子也有如此功效。强烈建议运行每个程序（所有这些程序都可在线获得），并比较系统上获得的结果与本书中的结果。这么做，不仅可以学会Objective-C语言及其语法，而且还能熟悉编译和运行Objective-C程序的过程。

1.2 本书的组织方式

本书分为三个逻辑部分：第一部分是“Objective-C 2.0语言”，介绍了该语言的基础知识；第二部分是“Foundation框架”，讲述了如何使用构成Foundation框架的种类丰富的预定义类。第三部分是“Cocoa程序设计和iPhone SDK”，简要介绍了Cocoa的应用程序套件框架，然后逐步演示了如何使用UIKit框架开发简单的iPhone应用，以及如何使用Xcode和Interface Builder开发并调试代码。

框架就是一组从逻辑上组合在一起的类和例程，它们使开发程序更加容易。使用Objective-C进行程序设计时需要的许多能力都来源于大量可用的框架。

第2章首先讲述了如何使用Objective-C编写第一个程序。

因为本书并非主要讲解Cocoa程序设计的，所以在第三部分之前没有过多介绍图形用户界面，甚至很少提及它。这就需要使用一种方法实现程序输入并产生输出。本书中的大多数例子

都是从键盘获得输入并在一个窗口中产生输出：如果在命令行使用gcc，那么这个窗口是Terminal窗口；如果使用Xcode，那么这个窗口是控制台窗口。

第3章介绍了面向对象程序设计的基础。本章引入了一些术语，但数量保持了最少。本章还介绍了定义类的机制，以及向实例或对象发送消息的方式。教师或者有经验的Objective-C程序员将会注意到：本书使用静态类型声明对象。我认为这种方法是学生起步的最好方式，因为编译器能捕捉更多的错误，程序有更强的自文档化（self-documenting）功能，同时还能鼓励新程序员显式声明已知的数据类型。这样，id类型的概念及其强大功能直到第9章才会完全显现出来。

第4章描述了基本的Objective-C数据类型以及如何在程序中使用它们。

第5章介绍了可用在程序中的3种循环语句：for、while和do。

决定选择是任何计算机程序设计语言的基础。第6章详细讲述了Objective-C语言的if和switch语句。

第7章更深入地研究了类和对象的使用，详细讨论了方法、方法的多个参数以及本地变量的相关内容。

第8章介绍了继承的主要概念。这一特性使得程序的开发更容易，因为我们可以利用以前编写的代码。使用继承以及子类的概念可以方便地修改和扩展现有的类定义。

第9章讨论了Objective-C语言的3个重要特性。多态、动态类型以及动态绑定是本章的关键概念。

第10章至第13章对Objective-C深入讨论，既包含对象的初始化、协议、分类、预处理程序，还包括一些基本的C语言特性，如函数、数组、结构和指针。第一次开发面向对象的程序时，通常不必（最好避免）使用这些特性。建议你首次通读本书时略过第13章，只在需要了解这门语言的特殊特性的更多内容时再返回来学习它。

第二部分从第14章开始，这部分介绍了Foundation框架以及如何访问它的文档。

第15章至第19章讲解了Foundation框架的重要特性。这包括数字和字符串对象、集合、文件系统、内存管理以及对象的复制和归档。

学习完第二部分后，你将能够使用Foundation框架开发出相当复杂的Objective-C程序。

第三部分从第20章开始，本章简要介绍了应用程序套件，它提供了在Mac上开发复杂图形应用所需的各种类。

第21章介绍了iPhone SDK和UIKit框架。本章阐述了如何以迭代的方式编写简单的iPhone（或iTouch）应用，然后列举了一个计算器应用的示例，通过它可使用iPhone进行简单的分数算术运算。

因为面向对象的用语涉及大量术语，所以附录A提供了一些常用术语的定义。

附录B对Objective-C语言进行了总结，它用于快速参考。

附录C给出了本书第二部分开发并大量使用的两个类的源代码。这些类定义了地址卡和地址簿类。使用方法可以执行简单的操作，如在地址簿中添加和删除地址卡、查找某人、列出地址簿的内容等。

学会如何编写Objective-C程序后，可以继续向几个不同的方向发展。你可能想学习有关C

语言的更多内容，或开始编写在Mac OS X上运行的Cocoa程序，或者是开发更复杂的iPhone应用。不管是哪种情况，附录D都能引导你向正确的方向迈进。

1.3 致谢

我要感谢在本书第一版的准备阶段为我提供帮助的朋友们。首先，要感谢Tony Iannino和Steven Levy对原稿的审阅，并感谢Mike Gaines对本书的贡献。

其次，还要感谢本书的技术编辑Jack Purdum（第一版）和Mike Trent。我很幸运，因为Mike审阅了本书的两个版本。Mike对我所编写的每一本书都进行了最详细的审阅，他不仅指出了书中的不足之处，而且还十分慷慨地提出了建议。正是因为Mike对第一版提供的意见，我改变了讲解内存管理的方法，并尽力确保本书中的每个示例都是“无漏洞的”。Mike还为有关iPhone程序设计的章节提供了宝贵的贡献。

在第一版中，Catherine Babin提供了封面的图片，他还提供了许多有价值的图片供我选择。朋友为我制作的封面艺术效果，使得本书具有了更强的专业性。

我非常感谢Pearson的Mark Taber，他忍受了我的推迟交稿并且非常和蔼地让我按照自己的进度工作，并且容忍我在撰写第二版时又不断地将最终完成时间推后。对于Pearson的朋友，我还要感谢我的开发编辑Michael Thurston、文字编辑Krista Hansing以及我的项目编辑Mandie Frank，他出色地管理了很多乱糟糟的事情并最终漂亮地完成了任务。

和往常一样，当我整个夏天（这项工作一直到秋天才完成）都忙于此书时，我的孩子们表现出了令人难以置信的成熟和耐心！感谢Gregory、Linda和Julia，我爱你们！

Stephen G. Kochan

2008年10月

第一部分 Objective-C语言

- 第2章 Objective-C程序设计
- 第3章 类、对象和方法
- 第4章 数据类型和表达式
- 第5章 循环结构
- 第6章 选择结构
- 第7章 类
- 第8章 继承
- 第9章 多态、动态类型和动态绑定
- 第10章 变量和数据类型
- 第11章 分类和协议
- 第12章 预处理程序
- 第13章 基本的C语言特性

第2章 Objective-C程序设计

本章我们将直接切入正题，并演示如何编写第一个Objective-C程序。你现在还不需要使用对象，对象是下一章讨论的主题。希望你能通过本章理解编写程序和编译并运行它的各个步骤。我们重点讲解了在Windows和Macintosh计算机上的这一过程。

首先，举一个十分简单的例子，在屏幕上显示短语“Programming is fun!”的程序。无需大费周折，代码清单2-1显示了用于完成此任务的Objective-C程序。

代码清单2-1

```
// First program example

#import <Foundation/Foundation.h>

int main (int argc, const char * argv[])
{
    NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];
    NSLog (@\"Programming is fun!\");

    [pool drain];
}
```

```

    return 0;
}

```

2.1 编译并运行程序

在详细解释这个程序之前，先要学习编译和运行此程序的步骤。可以使用Xcode编译并运行程序，或者使用GNU Objective-C编译器在Terminal窗口中编译并运行程序。我们将用这两种方法分别实现这一系列步骤。然后，你可以确定如何处理本书其余部分的程序。

注意 这些工具应该预安装在所有使用OS X的Mac机上。如果单独安装OS X，则务必要安装Developer Tools。

2.1.1 使用Xcode

Xcode是一款功能齐全的应用程序，使用它可轻松输入、编译、调试和执行程序。如果想在Mac机上开发应用程序，那么学习使用这个强大的工具是值得的。这里只是介绍入门知识，后面会再次回到Xcode并逐步介绍如何使用它开发图形应用程序。

首先，Xcode位于Developer文件夹内一个名称为Applications的子文件夹中。图2-1显示了该工具的图标。



图2-1 Xcode图标

启动Xcode，在File菜单下，选择New Project（参见图2-2）。此时出现一个窗口，如图2-3所示。

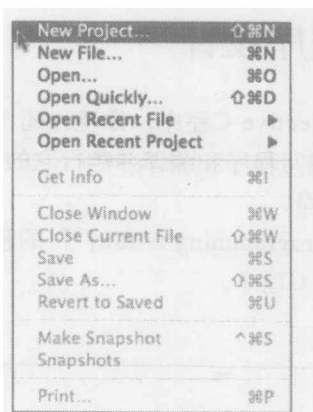


图2-2 启动一个新项目

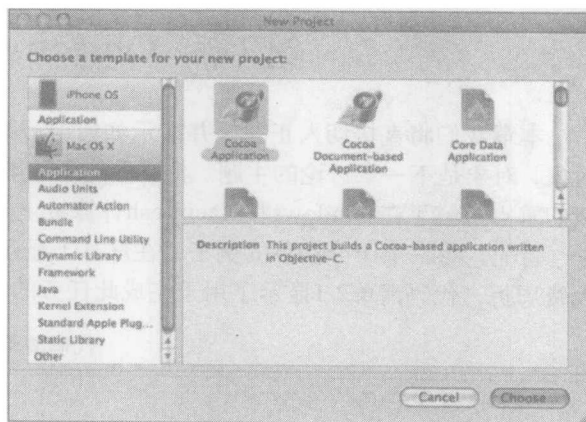


图2-3 启动一个新项目：选择应用程序类型

在左侧窗格中向下滚动，直至找到Command Line Utility。在右上窗格中，突出显示Foundation Tool，现在窗口应该如图2-4所示。

单击Choose，这会打开新窗口，如图2-5所示。

我们将第一个程序命名为prog1，所以在Save As字段中输入此名称。可以创建一个单独的文件夹来存储所有的项目。在我的系统上，我将本书中使用的项目保存在一个称为ObjC Progs

的文件夹中。

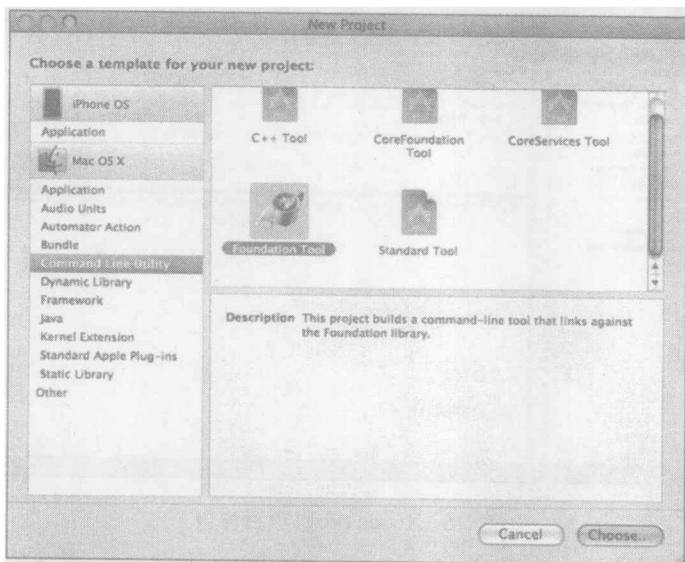


图2-4 启动一个新项目：创建Foundation Tool

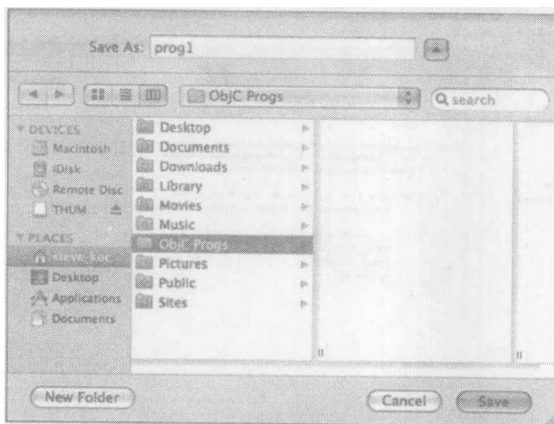


图2-5 Xcode文件列表窗口

单击Save按钮创建新项目，此时显示一个项目窗口，如图2-6所示。注意，如果你已经使用过Xcode或更改了其中的选项，那么显示的窗口可能与此处的稍有不同。

现在可以输入第一个程序了。在右上窗格中选择文件prog1.m，此时应该出现如图2-7所示的Xcode窗口。

Objective-C源文件使用.m作为文件名的最后两个字符（称为扩展名）。表2-1列出了其他常用的文件扩展名。

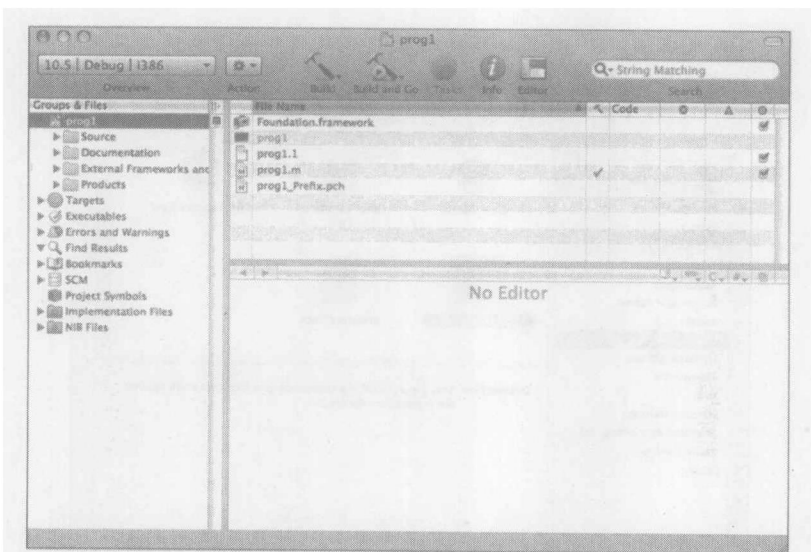


图2-6 Xcode prog1项目窗口

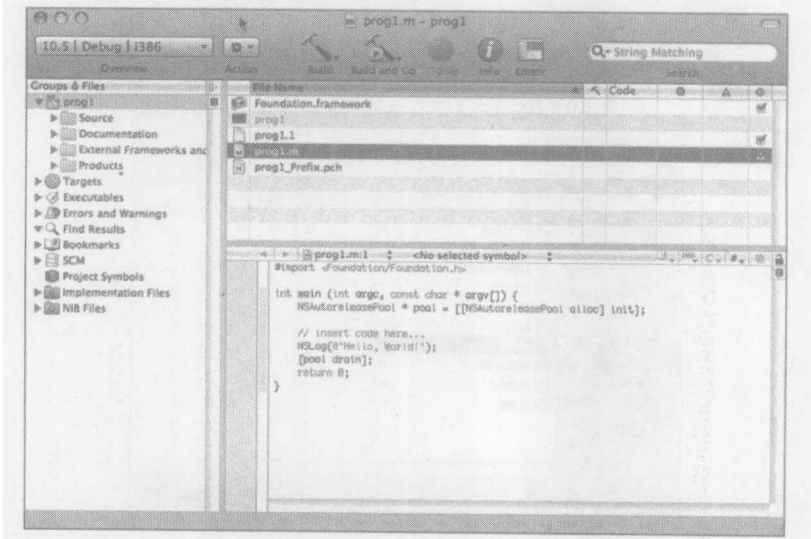


图2-7 文件prog1.m和编辑窗口

表2-1 常见的文件扩展名

扩展名	含 义	扩展名	含 义
.c	C语言源文件	.mm	Objective-C++源文件
.cc, .cpp	C++语言源文件	.pl	Perl源文件
.h	头文件	.o	Object (已编译的) 文件
.m	Objective-C源文件		