苹果开发
与应用系列

HZ BOOKS
华章科技

PEARSON

（美） Robert Clair 著
李强 等译

Objective-C 2.0 Mac和iOS开发实践指南

LEARNING OBJECTIVE-C 2.0
A HANDS-ON GUIDE TO OBJECTIVE-C FOR MAC AND IOS DEVELOPERS

机械工业出版社
China Machine Press

苹果开发
与应用系列

（美） Robert Clair 著
李强 等译

Objective-C 2.0 Mac和iOS开发实践指南

机械工业出版社
China Machine Press

本书专门为那些想要学习Objective-C以便为Mac OS X或iOS编写程序的程序员量身打造。本书分为四部分。第一部分介绍学习Objective-C编程所需的C语言基础知识。第二部分介绍Objective-C编程的核心知识，包括消息机制、类和对象、类对象等。第三部分介绍Objective-C中的高级概念，包括引用计数、垃圾收集和Objective-C 2.0的新功能块。第四部分提供了Objective-C程序员常用的信息和资料。本书大多数章末都提供了练习，读者可以通过练习巩固在书中学到的知识。

本书内容精练、可读性强、易于学习，侧重通过示例来介绍知识点和概念，是一本学习Objective-C 2.0不可多得的入门实践指南。本书适合想要学习Objective-C 2.0编程语言的初、中级程序员阅读，也可以作为社会培训机构的入门级培训教辅材料。

本书版权登记号：图字：01-2010-5851

图书在版编目（CIP）数据

Objective-C 2.0 Mac和iOS开发实践指南 /（美）克莱尔（Clair, R.）著；李强等译. —北京：机械工业出版社，2011.1

（苹果开发与应用系列）

书名原文：Learning Objective-C 2.0: a Hands-On Guide to Objective-C for Mac and iOS Developers

ISBN 978-7-111-32927-5

Ⅰ.O… Ⅱ.① 克… ② 李… Ⅲ.① 操作系统，Mac OS X—程序设计 ② 面向对象语言—程序设计 Ⅳ.① TP316.89 ② TP312

中国版本图书馆CIP数据核字（2010）第261509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：秦 健

北京市荣盛彩色印刷有限公司印刷

2011年2月第1版第1次印刷

186mm × 240mm · 18.75印张

标准书号：ISBN 978-7-111-32927-5

定价：55.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

对本书的赞誉

在本书中，作者直切主题，不仅全面介绍了Objective-C，而且从深入实际、注重实践的角度给出了节省读者时间和精力的入门简介。精练的概括、示例及具体实现的细节，能帮助读者快速、完整地理解这一语言及其核心功能和概念。

——Scott D. Yelich，*移动应用开发者*

有很多关于Objective-C的图书试图介绍面向对象编程、Objective-C计算机语言和Apple平台上的应用开发的整个内容。如此范围的主题太宽泛了，岂是单独一本篇幅有限的图书所能完全囊括的？本书专注介绍Objective-C的基础知识，它能帮助你成为一名编写Objective-C代码的合格程序员。

——Joseph E. Sacco博士，J.E. Sacco & Associates公司

本书是一本大师级的教程，它针对Objective-C语言提供了深入的、有趣的介绍，对于程序员新手和资深专业人士都有启发意义。当有刚接触Objective-C的程序员询问该从何处开始时，我会推荐这本书给他们。

——Matt Long, Cocoa Is My Girlfriend (www.cimgf.com)网站

本书作者精通Objective-C语言，并且以一种易于学习的方式来介绍它。不管你是新手还是专业程序员，都可以在本书中有所收获，而这并不需要预先有C语言的知识。

——Cory Bohon，MaclLife独立开发者和博客

我喜欢这本书，因为它的技术内容不枯燥，可读性好且有深度。

——Andy Lee，AppKiDo的作者

译 者 序

当今移动互联网正在推动互联网第三波浪潮的到来。根据摩根士丹利的调查报告，2010年全球的移动互联网用户已达到10亿人，产值近千亿美元。而中国的手机上网用户也已达到2.77亿个，由此产生的市场总值在200亿元人民币左右。移动互联网不仅将深入改变人们的生活方式，而且会造就一个又一个新的数字财富英雄。

在移动互联网浪潮中，移动应用开发将起到重要作用。App Store的成功模式给移动应用开发搭建了良好的商业模式和环境。目前有越来越多的开发者加入到移动开发的队伍中。而基于Mac OS和iPhone的移动应用开发则是其中重要的一个分支。

Objective-C最初由Smalltalk演变而来，它是基于C语言的面向对象扩展。目前，Objective-C是Mac OS的首要开发语言，也是GNUstep在Linux和其他平台上的开发语言。随着基于Mac OS和iPhone、iPad开发的兴起，Objective-C受到越来越多的关注，其流行度也在不断攀升。在TIOBE编程语言排行榜2010年10月的榜单中，Objective-C排在第8位，而与去年相比，Objective-C的份额提升最多，达到2.54%，同时入选了2010年度语言。因此，关注与学习Objective-C的人越来越多。

本书集中介绍了Objective-C语言，专门为那些想要学习Objective-C以便为Mac OS X或iOS编写程序的程序员量身打造（iOS用于iPhone、iPod Touch和iPad）。本书分为四部分。第一部分回顾了学习Objective-C编程所需的C语言基础知识，并简单介绍了Objective-C程序的结构。第二部分介绍Objective-C编程的核心知识，包括消息机制、类和对象、类对象、框架、Foundation类、控制结构、分类、特性、协议，等等。第三部分介绍了Objective-C中的高级概念，包括引用计数、垃圾收集和Objective-C 2.0的新功能块。第四部分包括5个附录，分别提供了Objective-C程序员常用的信息和资料。本书大多数章末都提供了练习，以帮助读者巩固在书中学到的知识。本书内容注重实用，侧重通过示例来介绍知识点和概念，因此本书对于初中级程序员来说，是一本不可多得的入门实践指南。

参加本书翻译的有李强、关志兴、王建勇、毛立涛、闫柳青、姜巧生、沈海峰、谢扣林、乔义峰、刘查强、王义强、刘国际、杨传辉和王建华等。读者在阅读过程中如有疑问，可通过translatebook@163.com与译者联系交流。

前　言

Objective-C是C语言的面向对象扩展。可以将其称为“带有对象的C”。如果你对学习Objective-C感兴趣，通过学习可以针对Mac OS X或iOS编写程序。但是，还存在学习Objective-C的另一个原因，那就是它是一门有趣的语言，并且相对容易学习。和现实世界的任何其他东西一样，Objective-C也有一些难点。但是，总体来说，它是比其他面向对象语言简单得多的一种语言，特别是和C++相比。Objective-C对C语言所进行的扩展，只需要一两页篇幅就能列举出来。

在Apple的世界中，Objective-C并不是单独工作的。它与称为框架的两个类库一起使用。Foundation框架包含了用于基本实体的类，例如，字符串和数组，以及包装了与操作系统进行交互的类。AppKit包含了用于窗口、视图、菜单、按钮及构建图形用户界面所需的其他各种挂件的类。将这两个框架综合起来就是Cocoa。在iOS中用一个叫做UIKit的不同框架取代了AppKit。Foundation和UIKit合称为Cocoa Touch。

Objective-C由Brad J. Cox在20世纪80年代早期创建。在1988年，由Steve Jobs在离开Apple后创建的NeXT Computer公司许可使用Objective-C，并且以其作为创建在NeXT的NeXTSTEP操作系统下运行的应用程序开发环境的基础。NeXT工程师开发了一组Objective-C库，以便构建应用程序。在NeXT于1933年减少硬件业务之后，它与Sun Microsystems合作开发了OpenStep，这是针对一种面向对象系统的开放性规范，以NeXTSTEP API为基础。Sun最终丧失了对OpenStep的兴趣。NeXT继续销售其OpenStep版本，直到Apple于1997年年初收购了NeXT。NeXTSTEP操作系统变成了Mac OS X的基础。NeXT Objective-C库变成了Cocoa的基础。

本书集中介绍了Objective-C语言。它不是教你如何编写Cocoa程序，或者使你成为Xcode的专业用户。本书只是涉及了Foundation框架的一小部分，并且只是介绍性地提及AppKit和UIKit。本书所做出的假设是，如果你首先对Cocoa所基于的语言有一个较好的理解，那么你学习Cocoa将会变得更容易。

本书的目标读者

本书是为那些想要学习Objective-C以便为Mac OS X或iOS编写程序的程序员量身打造的(iOS用于iPhone、iPod Touch和iPad)。尽管从技术上讲，也有可能使用其他的语言来编写完整的Mac OS X程序，但编写一个遵从Apple Human Interface Guidelines[⊖]并且拥有相应的Mac观感的应用程序，则需要使用Objective-C Cocoa框架。即便你使用不同的语言，例如纯C或C++，来编写一个Mac应用程序的核心，用户界面层还是应该使用Objective-C编写。在为iOS编写程

⊖ 参见http://developer.apple.com/mac/library/documentation/UserExperience/Conceptual/AppleHIGuidelines。

序时，程序员没有任何选择：iPhone应用程序的外层和用户界面必须使用Objective-C编写。

对于那些想要使用来自GNUStep项目[⊖]的软件为其他平台编写Objective-C程序的程序员来说，本书也很有用（GNUStep是OpenStep库的一个开源实现）。

你需要知道什么

本书假设读者拥有C语言的使用经验。Objective-C是C的一个扩展，本书主要关注Objective-C向C中添加了什么。对于那些具有C经验的人，以及那些适合快速学习一门新语言的人来说，本书第2章和第3章对C的基本部分进行了回顾，这些基本知识是编写一个Objective-C程序很可能要用到的。如果你没有C或任何类似C（C++、Java和C#）的计算机语言的经验，那么你需要在学习本书的同时阅读一本关于C的图书。此前接触过一种面向对象语言对学习本书有帮助，但不是必需的。随着本书的推进，将会分别介绍必需的面向对象概念。

Objective-C 2.0中的新内容

如果你已经知道一些Objective-C的知识，并且想要直接跳到2.0版新增内容的部分，那么关于这部分的知识在如下的各章中介绍：

- 快速枚举（第10章）为遍历对象的一个集合提供了一种简单（且快速）的方式。
- 声明特性（第12章）提供了一种简单的方式来指定一个对象的实例变量，并且让编译器为你创建访问这些变量的方法。
- 垃圾收集（第15章）添加了对Objective-C的自动内存管理。
- 块（第16章）允许你定义类似函数的对象，它们自身带有环境。

本书是如何组织的

本书分为四部分：第一部分是C的概览，然后介绍面向对象编程和Objective-C。本书的第二部分详细介绍了Objective-C语言，并且对Foundation框架进行了介绍。本书的第三部分介绍了在Objective-C中使用的两种内存管理形式，以及Objective-C 2.0新添加的块功能。本书的第四部分是附录，提供了Objective-C程序员常用的信息和资料。

第一部分　Objective-C简介

- 第1章概述了C的基本知识，介绍了在编写Objective-C程序时最可能用到的C知识。
- 第2章继续概述C，讨论C和Objective-C程序的内存布局，以及内存寻址和不同类型变量的生命周期。即便你熟悉C，也可能需要阅读本章。很多有经验的C程序员并不完全熟悉本章介绍的内容。
- 第3章开始介绍面向对象编程的概念，并且继续探讨这些概念是如何在Objective-C中体现的。

⊖　参见www.gnustep.org。

- 第4章详细介绍了一个简单的Objective-C程序的组成。还介绍了如何使用Xcode来创建一个项目，然后编译并运行一个Objective-C程序。可以使用这些知识来完成本书中其余章的练习。

第二部分　语言基础

对象是面向对象编程中的主要实体，它们将变量（称为实例变量）和类似函数的代码块（叫做方法）组织到一个单独的实体中。类是对象的规范。它们列出了组成一个给定的对象的实例变量和方法，并且提供了实现这些方法的代码。对象更加具体，它们是一个内存范围，类似于C结构，保存了对象的类所定义的变量。一个特定的对象称为定义了它的类的一个实例。

- 第5章开始对Objective-C语言进行全面介绍。在Objective-C中可以通过向一个对象发送一条消息来让它“做某些事情”。消息是一个方法的名称加上方法所接收的任何参数。作为对接收到的消息的响应，对象会执行相应的方法。本章介绍方法、消息，以及Objective-C消息系统是如何工作的。
- 第6章介绍如何定义类，以及如何创建和复制对象实例。这一章还介绍了继承，即通过扩展一个已有的类来定义一个类的过程，而不是从头开始创建一个类。

  在执行一个Objective-C程序的过程中，使用的每个类都由包含关于类的信息的一段内存来表示。这段内存叫做类的类对象。类也可以定义类方法，它是类执行的方法，而不是由类的实例执行的。
- 第7章介绍类对象和类方法。与某些其他面向对象语言中的类不同，Objective-C的类没有类变量，即类的所有实例所共享的变量。这一章的最后几节介绍如何通过使用静态变量来达到类变量的效果。
- 第8章介绍Apple封装动态链接库的方式。这一章介绍了框架的定义和结构，并简单介绍了在编写Mac OS X或iOS程序时将会遇到的一些常见框架。
- 第9章介绍最常用的Foundation类：用于字符串、数组、字典、集合和数字对象等的类。
- 第10章讨论当使用带有C控制结构的Objective-C构造时，需要考虑的一些问题。本章继续介绍Objective-C所添加的其他控制结构，包括Objective-C 2.0的新的快速枚举构造。这一章最后介绍了Objective-C的异常处理系统。
- 第11章介绍如何向一个已有的类添加方法而不需要子类化它，以及如何隐藏你认为私有的方法的声明。这一章最后讨论Objective-C的安全性问题。
- 第12章介绍Objective-C 2.0最新声明的特性功能。特性是一个对象的特征。特性通常由对象的实例变量来构建。设置和获取一个特性的方法叫做访问器方法。通过使用声明特性功能，你可以要求编译器合成一个特性的访问器方法，从而大大节省自己的精力。
- 第13章介绍分类对象的一种不同方法。协议是定义的一组方法，一个类可以选择实现这些方法。在很多情况下，重要的不是一个对象的类，而是对象的类是否通过实现协议中声明的方法来采用一个特定的协议（可以有多个类采用一个给定的协议）。Java关于接口的概念就是借用自Objective-C协议。

第三部分　高级概念

Objective-C提供了两种不同的系统来管理对象内存：引用计数和自动垃圾收集。

- 第14章介绍Objective-C传统的引用计数系统。引用计数也叫做保留计数或管理内存。在一个使用引用计数的程序中，每个对象保持一个计数，命名为引用计数，这是使用该对象的其他对象的数目。当这个计数减少到0的时候，就会销毁该对象。这一章介绍了正确使用引用计数所需要的规则。
- 第15章介绍Objective-C的新的自动垃圾收集系统。使用垃圾收集时，有一个单独的名为垃圾收集器的线程，它负责确定哪些对象不再需要并释放它们。垃圾收集使你不必负责大多数内存管理的杂事。
- 第16章介绍Objective-C的新的块功能。块类似于一个匿名函数，但是块带有其包围环境中的变量的值。块是Apple的GCD并发机制的核心功能。

第四部分　附录

- 附录A提供对编译器来说具有特殊含义的名称的一个表，以及Objective-C编译器指令的列表。编译器指令是以一个@字符开头的单词，它们是在各种条件下传给编译器的指令。
- 附录B给出Foundation类的一个列表，其实例具有相同的内存布局，并且可以与来自低层级的C语言Core Foundation框架的对应的对象交互使用。
- 附录C简单介绍Apple向64位计算的迁移。
- 附录D。旧的“遗留”的Objective-C运行时用于32位的OS X程序，新的“现代”运行时用于在OS X 10.5 或更高版本上运行的64位Objective-C程序以及iOS程序，附录D介绍了二者之间的区别。
- 附录E列出了包含对Objective-C开发者有用信息的图书和Web站点。

编译时和运行时

当你编写程序时，有两个时刻很重要：编译时，即当你的源代码翻译成机器语言并且连接到一起形成一个可执行的程序时；还有就是运行时（也叫做执行时），即当可执行的程序作为某台计算机上的一个进程运行时。区分Objective-C和其他常用语言（特别是C++）的特征之一是，Objective-C是一种非常动态的语言。“动态”意味着其他语言在编译时所做出的决定，在Objective-C中都推迟到运行时做出。关于这一点的最突出的例子就是Objective-C的消息系统。当一个程序计算一个消息表达式时（等同于其他语言中的方法调用）所要执行的代码的部分，是在运行时确定的。

推迟决定直到运行时，这有很多的优点，你将会在阅读本书的时候看到这些优点，但是，它也有一个显著的缺点，那就是它限制了编译器所能进行的检查的数量。当使用Objective-C编写代码时，原本在其他语言中会在编译阶段捕获的一些错误，在Objective-C中都会出现在运行时。

内存管理

正如前面所提到的，Objective-C 2.0在使用手动引用计数系统和自动垃圾收集来管理对象内存方面提供了选择。除了第15章介绍的Objective-C 2.0的垃圾收集系统以外，本书的其他部分在所有的示例中一开始就在使用引用计数。第14章详细介绍了引用计数。

这么做的主要原因是垃圾收集在iOS上不可用。如果你想要为iPhone、iPod Touch或iPad编写程序，就必须学习Objective-C的引用计数系统。

从各种Objective-C和Cocoa的邮件列表中的内容来判断，引用计数可能是人们学习Objective-C的过程中引发混淆的最大源头。但是，如果你及早地学习其规则并规范地使用它们，你将会发现，实际上使用引用计数并不难。

如果在随后的某个时刻，你想要在一个项目中使用垃圾收集，转换过程更应该是相当容易的。尽管在从引用计数迁移到垃圾收集时（第15章将介绍垃圾收集），你需要注意一些架构性的问题，但使用垃圾收集大多数时候只不过是不做那些当你使用引用计数时所要做的那些事情。

关于示例

为一本介绍性的图书创建代码示例，这提出了一个挑战：如何不迷失在那些要构建一个能工作的程序所必需的模板文件的海洋中。在很多情况下，本书采取了使用某种程度上假设的示例，并且为了帮助你关注所讨论的要点而对其进行了简化。不相关的代码部分均已省略，或者用省略号(...)替代。

例如：

```
int averageScore = ...
```

上面的代码表示`averageScore`是一个整数变量，其值由程序的其他某个部分获得。averageScore 值的来源对于该示例来说是不相关的，你只需要认为它有一个值就行了。

关于程序清单

本书中的示例采用未编号的程序清单和编号的程序清单的混合编排。

- 未编号的程序清单

主要是正文中引用的较短的代码片段，示例的前后均为正文。

- 编号的程序清单

编号的程序清单有标题，并且按照它们在一章中出现的顺序来编号（例如程序清单4.1和程序清单8.3）。这些主要是较大的示例，在该章正文后面或者在该章后面的练习中会引用到它们。

在这两种情况下，那些需要一行一行说明的示例，都给出了一个行编号，以便说明性的正文能够引用代码中特定的行。

关于练习

本书中的大多数章最后都有一组练习。当然鼓励你完成它们。很多练习要求你编写较小的

程序来验证在那一章正文中介绍的知识点。一些练习可能看上去很冗长，但是编写代码和查看结果都会提供比仅仅阅读代码更加生动的学习体验。编写较小的程序来测试你的理解，这是应该养成的宝贵习惯；对于不清楚的每个知识点，你都应该编写一个这样的程序，即便本书没有提供一个相关的练习。

练习没有建议哪个程序需要一个用户界面。所有这些都可以通过使用一个文本编辑器来编写，并且从命令行来编译和运行它们，如第2章以前的练习所示，或者使用一个简单的Xcode项目来编辑、编译和运行它们，如第4章所介绍。

致　谢

写过书的人都知道，即便是一个作者写的一本书，那也是团队努力的结果。本书也不例外，Scott D. Yelich、Andy Lee、Matt Long、Cory Bohon和Joachim Bean评阅了初稿。他们不仅发现了错误，而且还促使我更加仔细地思考一些最初我忽视了的问题。Steve Peter激发我开始编写本书，Daniel Steinberg对于初稿的形成给予了帮助。我要感谢Addison-Wesley的Romny French和我的编辑Chuck Toporek。我低估了写作各个章节所花的时间，并且初次使用Microsoft Word时备受挫折，而Chuck容忍了这一切。

当事情进展得不好的时候，每个人都需要富有同情心的倾听。在我编写本书的时候，以及准备开始写这本书之前的几十年里，我的朋友Pat O'Brien、Michael Sokoloff和Bill Schwartz每个人都乐于扮演这种倾听的角色。

有两个人值得特别提及：

Joseph E. Sacco博士阅读了本书的几个版本的草稿，测试了其中的练习，并且给予了众所周知的“很多有价值的技术讨论”，还有很多有价值的非技术讨论。

Ekko Jennings阅读了一些章节，并给予了精神上的支持和帮助，即便在本该由我做饭的时候也下厨烹饪，并且总是忍受我不停写作。谢谢！

作者简介

Robert Clair拥有Oberlin学院的物理学学士学位，以及加州大学伯克利分校的物理学硕士和博士学位。他在商业软件开发领域拥有20年以上的经验，主要从事CAD、建模和图形方面的工作。在过去的7年里，他主要从事Mac上的Objective-C工作，现在转移到iPhone上。他编写过ZeusDraw，这是一款用于Mac OS X的矢量绘图程序；还有ZeusDraw Mobile，这是用于iPhone的一款绘图程序。他还是众多的iPhone和iPad应用程序的顾问。他居住在纽约市，是Chromatic Bytes公司的负责人，这是一家独立的软件公司。

目　录

第二部分 语言基础

第三部分 高级概念

第四部分 附录