

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