

Objective-C
最新学习指南

- 最受欢迎的Objective-C培训教材全新升级版
- 权威解读Objective-C和Cocoa特性
- iPhone、iPad、Mac开发必备



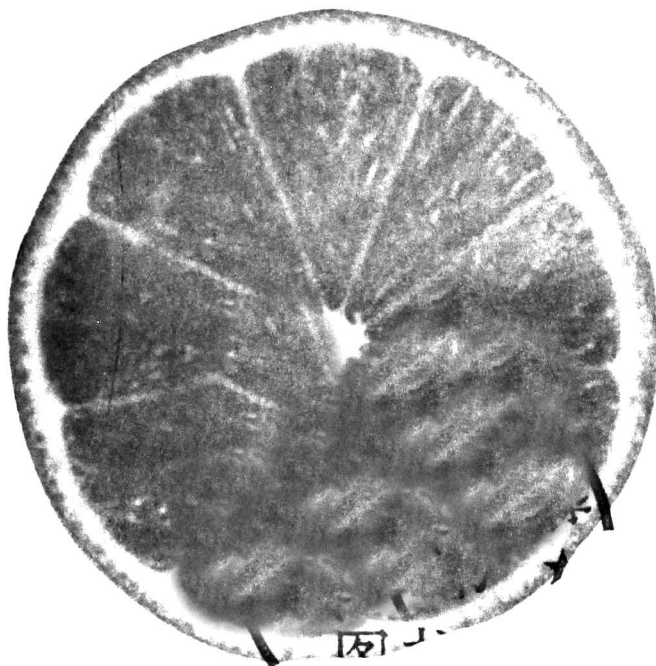
Learn Objective-C on the Mac
For OS X and iOS (2nd Edition)

Objective-C基础教程

(第2版)

Scott Knaster
[美] Waqar Malik 著
Mark Dalrymple
周庆成 译

TURING 图灵程序设计丛书 移动开发系列

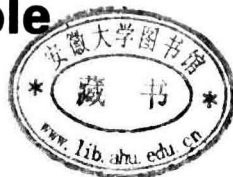


Learn Objective-C on the Mac
For OS X and iOS (2nd Edition)

Objective-C基础教程

(第2版)

Scott Knaster
[美] Waqar Malik 著
Mark Dalrymple
周庆成 译



人民邮电出版社

图书在版编目 (C I P) 数据

Objective-C基础教程 : 第2版 / (美) 克纳斯特 (Knaster, S.), (美) 马利克 (Malik, W.), (美) 达尔林普尔 (Dalrymple, M.) 著; 周庆成译. -- 北京: 人民邮电出版社, 2013.5

(图灵程序设计丛书)

书名原文: Learn Objective-C on the Mac: For OS X and iOS, Second Edition
ISBN 978-7-115-31458-1

I. ①O… II. ①克… ②马… ③达… ④周… III. ①C语言—程序设计—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第069008号

内 容 提 要

Objective-C 是扩展 C 的面向对象编程语言, 也是 iPhone 开发用到的主要语言。本书结合理论知识与示例程序, 全面而系统地介绍了 Objective-C 编程的相关内容, 包括 Objective-C 在 C 的基础上引入的特性、Cocoa 工具包的功能及框架, 以及继承、复合、源文件组织等众多重要的面向对象编程技术。附录中还介绍了如何从其他语言过渡到 Objective-C。

本书适合各类开发人员阅读。

图灵程序设计丛书

Objective-C基础教程 (第2版)

- ◆ 著 [美] Scott Knaster Waqar Malik Mark Dalrymple
- 译 周庆成
- 责任编辑 卢秀丽
- 执行编辑 张 霞
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
- 邮编 100061 电子邮件 315@ptpress.com.cn
- 网址 <http://www.ptpress.com.cn>
- 三河市海波印务有限公司印刷
- ◆ 开本: 800×1000 1/16
- 印张: 20.5
- 字数: 483千字 2013年5月第1版
- 印数: 1-5 000册 2013年5月河北第1次印刷

著作权合同登记号 图字: 01-2012-5557 号

ISBN 978-7-115-31458-1

定价: 59.00元

读者服务热线: (010)51095186

印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Learn Objective-C on the Mac: For OS X and iOS, Second Edition* by Scott Knaster, Waqar Malik, Mark Dalrymple, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2012 by Scott Knaster, Waqar Malik, Mark Dalrymple. Simplified Chinese-language edition copyright © 2013 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。

译者序

Objective-C是一种通用、高级、面向对象的编程语言。它扩展了标准的ANSI C编程语言，将Smalltalk式的消息传递机制加入到ANSI C中。它是苹果的OS X和iOS操作系统及其相关API、Cocoa和Cocoa Touch的主要编程语言。

Objective-C最初源于NeXTSTEP操作系统，之后在OS X和iOS中继承了下来。目前主要支持的编译器有GCC和Clang，其中Clang被应用于Xcode 4.0中。

20世纪80年代初，Brad Cox与Tom Love以SmallTalk-80语言为基础发明了Objective-C。Objective-C在C语言的基础上添加了扩展，成为了能够创建和操作对象的一门新的程序设计语言。

1988年，苹果前CEO乔布斯的NeXT Computer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。1994年，NeXT Computer公司和Sun Microsystem联合发布了一个针对NEXTSTEP系统的标准典范，名为OPENSTEP，在自由软件基金会的实现名称为GNUstep。1996年12月20日，苹果公司宣布收购NeXT Software公司，NEXTSTEP/OPENSTEP环境成为苹果操作系统主要发行版本OS X的基础。这个版本的开发环境被苹果公司称为Cocoa。在2006年7月苹果全球开发者会议中，Apple发布了Objective-C 2.0，增加了垃圾收集（只支持OS X系统）、属性、快速枚举等语法功能，改进了运行时性能，并添加了对64位系统的支持。2007年10月发布的Mac OS X v10.5中包含了Objective-C 2.0的编译器。

Objective-C作为面向对象的语言，其出现时间比C++还要早。随着OS X系统与iOS平台的不断发展，越来越多的移动开发者开始学习这门语言，因而其市场份额不断增大，排名也一度超越C++和C#等主流语言。在2011年与2012年，Objective-C连续两次赢得了TIOBE的年度编程语言大奖。在未来的日子里，Objective-C依然还有很大的上升空间。

本书全面介绍了Objective-C语言的基础知识。正文共有20章及一个附录，内容涵盖了面向对象编程的基础知识，Objective-C特有的继承、复合、内存管理、ARC自动引用计数、对象初始化、协议、键/值编码等特性，以及如何使用Xcode配合Cocoa或Cocoa Touch进行应用开发。附录中详细讲述了使用其他语言的编程人员在学习Objective-C时应注意的事项。本书内容精彩丰富，采用了示例程序与理论知识相结合的方式，相关的示例程序代码文件可以在Apress官网或图灵社区上免费下载得到。

本书内容由浅入深、结构清晰、步骤明确、简单易学、寓教于乐，特别适合没有面向对象编程基础的初学者学习，也同样适合于Objective-C程序员阅读和参考之用，而即使是其他语言的开

发人员，通过阅读本书也可以很快掌握Objective-C。相信你通过本书的学习，能够轻松驾驭Objective-C语言。祝你开发的应用程序能早日在App Store上架。

在翻译本书的过程中，我得到了许多人的帮助。首先感谢图灵社区的朱巍老师给予我这一次机会，感谢编辑与校对本书的所有编辑，你们帮我解决了很多问题，才保证了这本书的质量。其次感谢购买本书的读者，你们的支持使这本书的出版更有意义。最后衷心感谢上一版的译者们，你们在国内iOS开发刚起步时克服各种困难翻译出版了本书的第一版，为国内的移动开发者们提供了便利。在翻译本书第二版时，译者采纳了读者针对上一版的错误提出的建议，并尽可能地加以改善，但仍难免有疏漏之处，希望读者能一如既往地为本书勘误。

前 言

每当采用新平台时，程序员都会面临一个艰巨的挑战，那就是熟悉编程语言、开发工具、设计模式和新环境的标准软件库。

一般来说，程序员写代码时还面临着尽快交付软件的压力，因此很容易继续采用原有系统上所用的方法。这经常导致代码无法真正适应新平台，或者重复了已有的功能，还会给后期维护带来不少麻烦。

在理想状态下，新程序员可以向有新平台开发经验的同事请教，请他们提供指导，指引方向。遗憾的是，iOS平台发展得太快，很少能有这样的指导者。

如果你身边没有指导者，那么有没有别的方法呢？

本书的三位作者是苹果开发者技术服务组织的元老，他们都为新近使用苹果技术的软件工程师解答过无数的问题，并且一直在向他们传授良好的开发习惯。这些经验都体现在本书之中，可为读者答疑解惑，让大家对苹果开发平台不仅知其然，也知其所以然。

例如，第3章在介绍基础概念时采用了循序渐进、有条不紊的方式，而不是一古脑儿地列出陌生的类、方法和技术，让你通过练习自己去理清消化。

本书是你学习苹果公司的iOS以及OS X开发平台核心语言的最佳指导。

John C. Randolph

致 谢

这本书是我们写的，但如果只有我们三位作者，本书根本不可能付梓出版。诚然，我们输入了文字，编写了代码，但若没有出色的出版团队，本书可能不过是一篇又臭又长的博文而已。

非常感谢Brent Dubi，他不厌其烦地给身处各地的我们发邮件，打电话，协调工作。感谢Nick Waynik为我们提供技术支持，这项任务真的很艰巨，因为苹果公司为Xcode和Cocoa添加了很多强大复杂的功能。还要感谢Gwenan Spearing，他帮我们斟字酌句，润色语言，使每句话都清晰简明。此外，还要感谢我们能干的文字编辑Heather Lang，他曾编辑本书的第一版，这次又挺身而出，帮助我们写出既符合语法又生动形象的文字。这些人都极大地提高了本书的质量，也使本书的内容更加精彩。

在此，Waqar还要感谢他的孩子Adam和Mishal，以及他美丽的妻子Irrum，给予他充足的时间写作本书。

准备工作

如果想要下载源代码或提交勘误，请访问Apress网站上的本书页面：www.apress.com/9781430241881。

为了更好地利用本书，建议你阅读时身边放台电脑，再准备一杯喜欢的饮料也是蛮不错的，不过要小心离电脑远一点哦，打翻了就悲剧了，因为修理费用会让你郁闷一整天的。

差不多就是这些了，愿本书能为你带来无限乐趣。

目 录

第 1 章 启程	1	3.4.4 扩展 Shapes-Object 程序	45
1.1 预备知识	1	3.5 小结	47
1.2 历史	1	第 4 章 继承	48
1.3 内容简介	2	4.1 为何使用继承	48
1.4 准备工作	3	4.2 继承的语法格式	51
1.5 小结	5	4.3 继承的工作机制	53
第 2 章 对 C 的扩展	6	4.3.1 方法调度	54
2.1 最简单的 Objective-C 程序	6	4.3.2 实例变量	55
2.2 解构 Hello Objective-C 程序	10	4.4 重写方法	57
2.2.1 #import 语句	10	4.5 小结	59
2.2.2 框架	11	第 5 章 复合	61
2.2.3 NSLog()和@"字符串"	11	5.1 什么是复合	61
2.3 布尔类型	14	5.2 自定义 NSLog()	62
2.4 小结	18	5.3 存取方法	65
第 3 章 面向对象编程的基础知识	19	5.3.1 设置 engine 属性的存取方法	66
3.1 间接	19	5.3.2 设置 tires 属性的存取方法	67
3.1.1 变量与间接	20	5.3.3 Car 类代码的其他变化	68
3.1.2 使用文件名的间接	22	5.4 扩展 CarParts 程序	69
3.2 在面向对象编程中使用间接	28	5.5 复合还是继承	70
3.2.1 过程式编程	28	5.6 小结	71
3.2.2 实现面向对象编程	34	第 6 章 源文件组织	72
3.3 有关术语	37	6.1 拆分接口和实现	72
3.4 Objective-C 语言中的 OOP	38	6.2 拆分 Car 程序	75
3.4.1 @interface 部分	38	6.3 使用跨文件依赖关系	77
3.4.2 @implementation 部分	41	6.3.1 重新编译须知	78
3.4.3 实例化对象	43	6.3.2 让汽车跑一会儿	79

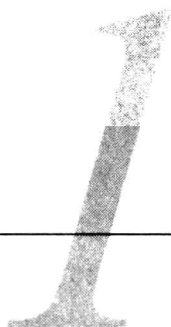
6.3.3 导入和继承	81	8.4.6 字符串内是否还包含别的字 符串	113
6.4 小结	82	8.4.7 可变性	114
第 7 章 深入了解 Xcode	84	8.5 集合大家族	115
7.1 窗口布局一览	84	8.5.1 NSArray	115
7.2 改变公司名称	85	8.5.2 可变数组	119
7.3 使用编辑器的技巧	86	8.5.3 枚举	120
7.4 在 Xcode 的帮助下编写代码	87	8.5.4 快速枚举	121
7.4.1 首行缩进 (美观排版)	88	8.5.5 NSDictionary	122
7.4.2 代码自动完成	88	8.5.6 请不要乱来	124
7.4.3 括号配对	90	8.6 其他数值	124
7.4.4 批量编辑	91	8.6.1 NSNumber	124
7.4.5 代码导航	94	8.6.2 NSValue	125
7.4.6 集中精力	96	8.6.3 NSNull	126
7.4.7 使用导航条	97	8.7 示例: 查找文件	126
7.4.8 获取信息	98	8.8 小结	130
7.5 调试	101	第 9 章 内存管理	131
7.5.1 暴力测试	101	9.1 对象生命周期	131
7.5.2 Xcode 的调试器	101	9.1.1 引用计数	132
7.5.3 精巧的调试符号	101	9.1.2 对象所有权	134
7.5.4 开始调试	101	9.1.3 访问方法中的保留和释放	134
7.5.5 检查程序	104	9.1.4 自动释放	136
7.6 备忘表	105	9.1.5 所有对象放入池中	136
7.7 小结	106	9.1.6 自动释放池的销毁时间	137
第 8 章 Foundation Kit 介绍	107	9.1.7 自动释放池的工作流程	138
8.1 稳固的 Foundation	107	9.2 Cocoa 的内存管理规则	140
8.2 使用项目样本代码	107	9.2.1 临时对象	141
8.3 一些有用的数据类型	108	9.2.2 拥有对象	141
8.3.1 范围	108	9.2.3 垃圾回收	143
8.3.2 几何数据类型	109	9.2.4 自动引用计数	144
8.4 字符串	109	9.3 异常	154
8.4.1 创建字符串	110	9.3.1 与异常有关的关键字	155
8.4.2 类方法	110	9.3.2 捕捉不同类型的异常	156
8.4.3 关于大小	111	9.3.3 抛出异常	156
8.4.4 字符串比较	111	9.3.4 异常也需要内存管理	157
8.4.5 不区分大小写的比较	112		

9.3.5 异常和自动释放池	158	12.1.1 开始创建类别	192
9.4 小结	159	12.1.2 @interface 部分	193
第 10 章 对象初始化	160	12.1.3 @implementation 部分	194
10.1 分配对象	160	12.1.4 类别的缺陷	195
10.1.1 初始化对象	160	12.1.5 类别的优势	196
10.1.2 编写初始化方法	161	12.1.6 类扩展	196
10.1.3 初始化时要做些什么	162	12.2 利用类别分散实现代码	197
10.2 便利初始化函数	163	12.3 通过类别创建前向引用	200
10.3 更多部件改进	164	12.4 非正式协议和委托类别	201
10.3.1 Tire 类的初始化	165	12.4.1 iTunesFinder 项目	202
10.3.2 更新 main() 函数	166	12.4.2 委托和类别	204
10.3.3 清理 Car 类	168	12.4.3 响应选择器	205
10.4 Car 类的内存清理 (垃圾回收方式 和 ARC 方式)	171	12.4.4 选择器的其他应用	206
10.5 指定初始化函数	172	12.5 小结	206
10.5.1 子类化问题	173	第 13 章 协议	207
10.5.2 Tire 类的初始化函数改进 后的版本	175	13.1 正式协议	207
10.5.3 添加 AllWeatherRadial 类 的初始化函数	175	13.1.1 声明协议	207
10.6 初始化函数规则	176	13.1.2 采用协议	208
10.7 小结	176	13.1.3 实现协议	209
第 11 章 属性	177	13.2 复制	209
11.1 使用属性值	177	13.2.1 复制 Engine	210
11.1.1 简化接口代码	178	13.2.2 复制 Tire	211
11.1.2 简化实现代码	179	13.2.3 复制 Car	212
11.1.3 点表达式的妙用	182	13.2.4 协议和数据类型	215
11.2 属性扩展	183	13.3 Objective-C 2.0 的新特性	215
11.2.1 名称的使用	186	13.4 委托方法	216
11.2.2 只读属性	188	13.5 小结	218
11.2.3 自己动手有时更好	189	第 14 章 代码块和并发性	219
11.2.4 特性不是万能的	189	14.1 代码块	219
11.3 小结	189	14.1.1 代码块和函数指针	219
第 12 章 类别	191	14.1.2 Objective-C 变量	223
12.1 创建类别	191	14.2 并发性	224
		14.2.1 同步	224
		14.2.2 队列也要内存管理	227
		14.2.3 操作队列	229

14.3 小结	231	18.4.1 休息一下	279
第 15 章 AppKit 简介	232	18.4.2 快速运算	282
15.1 构建项目	232	18.5 批处理	284
15.2 创建委托文件的@interface 部分	234	18.6 nil 仍然可用	285
15.3 Interface Builder	235	18.7 处理未定义的键	286
15.4 设计用户界面	236	18.8 小结	287
15.5 创建连接	239	第 19 章 使用静态分析器	288
15.5.1 连接输出口 (IBOutlet)	239	19.1 静态工作	288
15.5.2 连接操作 (IBAction)	240	19.1.1 开始分析	288
15.6 应用程序委托的实现	242	19.1.2 协助分析器	292
15.7 小结	244	19.1.3 了解更多	293
第 16 章 UIKit 简介	245	19.2 小结	295
16.1 视图控制器	249	第 20 章 NSPredicate	296
16.2 小结	263	20.1 创建谓词	296
第 17 章 文件加载与保存	264	20.2 计算谓词	297
17.1 属性列表	264	20.3 数组过滤器	298
17.1.1 NSDate	264	20.4 格式说明符	299
17.1.2 NSData	265	20.5 运算符	300
17.1.3 写入和读取属性列表	266	20.5.1 比较和逻辑运算符	300
17.1.4 修改对象类型	267	20.5.2 数组运算符	301
17.2 编码对象	268	20.6 有 SELF 就足够了	302
17.3 小结	273	20.7 字符串运算符	304
第 18 章 键/值编码	274	20.8 LIKE 运算符	304
18.1 入门项目	274	20.9 结语	305
18.2 KVC 简介	276	附录 从其他语言转向 Objective-C	306
18.3 键路径	277	索引	314
18.4 整体操作	278		

第 1 章

启 程



欢迎阅读本书！本书旨在教你学会Objective-C编程语言的基础知识。Objective-C语言是C语言的一个扩展集，几乎OS X或iOS平台上的所有应用程序都是用该语言开发的。

本书除了介绍Objective-C语言，还会介绍苹果公司为其提供的工具包Cocoa（针对OS X系统）和Cocoa Touch（针对iOS系统）。它们都是用Objective-C语言编写的，里面分别包含了OS X和iOS系统的所有用户界面元素和其他所有相关内容。学会了Objective-C之后，你就可以用Cocoa来开发功能完备的项目，还可以深入阅读其他相关书籍，比如*Learn Cocoa on the Mac*（Apress，2010）和*Beginning iOS 5 Development*^①（Apress，2011）。

本章会介绍一些在学习Objective-C语言之前需要了解的基本知识，还将介绍Objective-C语言的一些历史，并简要介绍其他各章的内容。

1.1 预备知识

读者在阅读本书之前，应对C语言或类似的编程语言（比如C++或Java）有一定的了解。无论是哪一种语言，都应该熟悉它的基本原理，理解什么是变量、方法和函数，知道怎样使用条件和循环语句来控制程序流方向。本书将重点介绍Objective-C在其基础语言C上添加的新特性，以及苹果公司Cocoa工具包的一些优秀功能。

对于不具备C语言基础的Objective-C学习者，可以先看看本书的附录或阅读*Learn C on the Mac*（Apress，2009）后，再来学习本书的内容。

1.2 历史

Cocoa和Objective-C是苹果公司OS X和iOS操作系统的核心。虽然OS X（尤其是iOS）出现的时间相对较晚，但Objective-C和Cocoa的推出则已有时日。早在20世纪80年代初，Brad Cox为了融合流行的、可移植的C语言和优雅的Smalltalk语言的优势，就设计出了Objective-C语言。1985年，史蒂夫·乔布斯创立了NeXT公司，致力于创建功能强大且经济实惠的工作站。NeXT

^① 中文版《iOS 5 基础教程》已由人民邮电出版社出版。——译者注

公司选择Unix作为操作系统并创建了NextSTEP（使用Objective-C语言开发的一款功能强大的用户界面工具包）。虽说它很有特点，并拥有了少量忠实的拥趸，但是在商业上却并没有获得成功。

苹果公司在1996年收购了NeXT（或者也可以说是NeXT收购了苹果公司^①）之后，NextSTEP更名为Cocoa，并得到了Macintosh开发人员的广泛认可。苹果公司的开发工具（包括Cocoa）都是免费提供的，只要具备一定的编程经验和基本的Objective-C知识，以及强烈的求知欲，任何程序员都可以使用这些工具。

有人可能会问：“既然Objective-C和Cocoa都是在20世纪80年代（那还是Alf和A-Team流行的时代，更不用提爷爷辈的Unix了）发明的，难道它们现在还没有过时吗？”当然没有！Objective-C和Cocoa是由一群优秀的编程人员耗费数年时间完成的，而且他们从未停止过更新与改进。经过多年发展，Objective-C和Cocoa已经演化成了一个美观精致且功能强大的工具集。近几年，iOS已经成为了最热门的开发平台，而Objective-C则是为其开发优秀应用程序的不二之选。因此，从NeXT最早采用至今已有二十多年，Objective-C的魅力依然不减当年。

1.3 内容简介

Objective-C是以C语言为基础的一个扩展集，它添加了一些微妙但意义重大的新特性。如果你接触过C++或Java编程语言，那么一定会惊叹Objective-C代码竟然如此简洁。本书其他章节将会详细介绍Objective-C在C语言基础上所添加的新特性。

- 第2章主要介绍Objective-C语言引入的基本特性。
- 第3章介绍面向对象编程的基础知识。
- 第4章介绍如何创建继承其父类特性的子类。
- 第5章讨论相关对象之间协同工作的技巧。
- 第6章演示创建程序源文件的实际策略。
- 第7章介绍Xcode的使用诀窍和强大功能，以帮助你提高编程效率。
- 第8章暂时告别Objective-C，转而介绍Cocoa的重要框架Foundation Kit，为你展现Cocoa的优秀特性。
- 第9章详细介绍Cocoa应用程序中的内存管理操作。
- 第10章讨论对象初始化的神奇。
- 第11章介绍了Objective-C语言中点表示法的实际作用，以及轻松访问对象的方法。
- 第12章详细描述了Objective-C语言中的一个非常出色的特性：类别（Category）。你可以通过它为现有的类（即便不是你所写的）添加自己的方法。
- 第13章介绍了Objective-C的一种继承方式——协议（Protocol），它允许类文件实现打包的特性集。

^① 当时苹果公司已濒临绝境，在收购NeXT后，乔布斯成为了苹果的CEO，开始大刀阔斧地进行改革。——编者注

- 第14章展示了如何使用Objective-C的最新特性“程序块”（Block, 它能够包含数据和代码）来增强函数的功能。
- 第15章介绍如何用AppKit框架来开发精致的OS X应用程序。
- 第16章与第15章类似, 只不过它介绍的是iOS应用程序的基础框架UIKit。
- 第17章将展示如何保存和检索数据。
- 第18章讲解了如何使用“键-值编码”方法来间接处理数据。
- 第19章介绍了如何利用Xcode中强大的工具来查找程序员平常会出现的错误。
- 最后, 第20章介绍如何分解数据。

如果你之前使用的是Java或C++等其他编程语言, 或是Windows或Linux等其他平台, 那么可以先阅读一下本书的附录, 其中指出了学习Objective-C所需要克服的一些思维障碍。

1.4 准备工作

Xcode是苹果提供的用来创建iOS和OS X应用程序的开发环境。Mac电脑上并没有预装Xcode, 不过只要你的苹果电脑运行的是OS X 10.7以上的系统, 就可以轻松地免费下载并安装。

踏上OS X和iOS开发漫长奇妙之旅的第一步就是确保在电脑上安装了Xcode。没有安装的读者可以从Mac App Store上下载安装。方法是点击Dock栏上的App Store图标(如图1-1所示), 也可以在用户的应用程序文件夹中打开App Store。



图1-1 Dock栏上的App Store图标

然后在Mac App Store应用程序右上角的搜索框中输入Xcode进行搜索(如图1-2所示)。

或者你也可以点击顶部的“类别”按钮并选择“软件开发工具”选项, 此时你将在顶端某个位置看到Xcode(如图1-3所示)。点击Xcode图标便会跳转到它的下载页面了(如图1-4所示)。

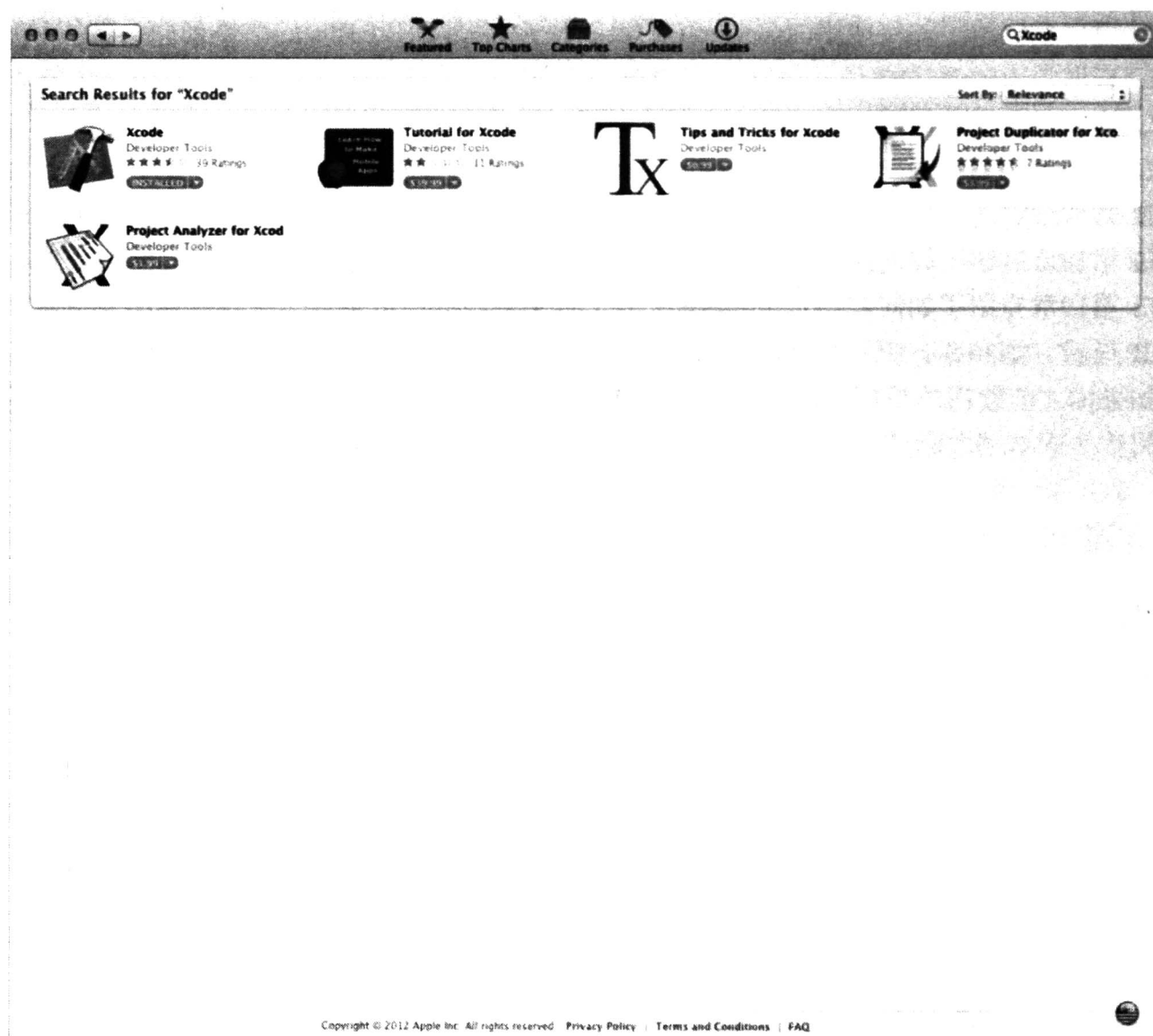


图1-2 在Mac App Store程序中搜索Xcode

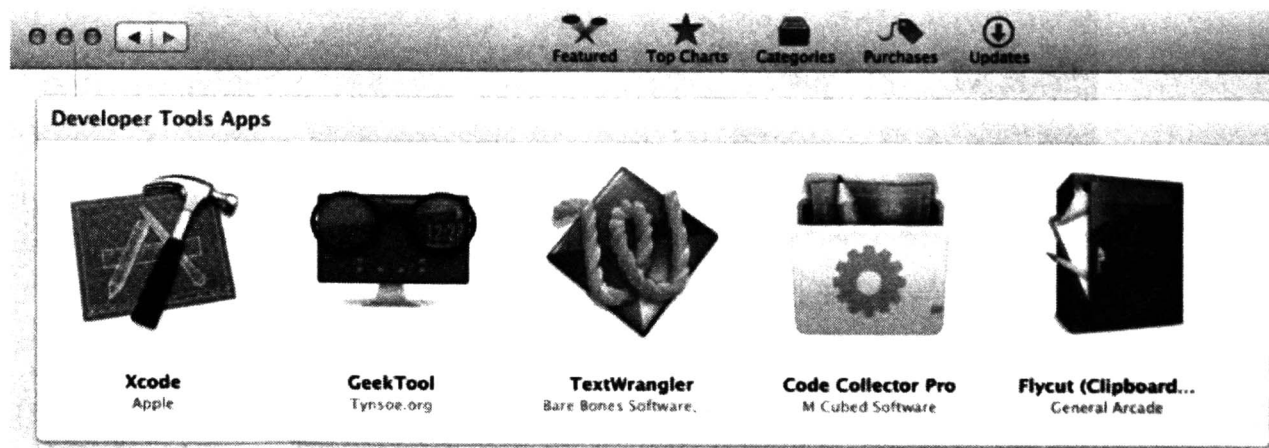


图1-3 软件开发工具



图1-4 Mac App Store中Xcode的下载页面

请点击“免费”按钮，然后点击“安装App”按钮。接下来App Store便会把Xcode^①安装到你的应用程序文件夹中。

现在你可以开始Objective-C学习之旅了。祝你好运！我们会与你一起开始这场旅途，至少会陪你踏上第一段旅途。

1.5 小结

OS X和iOS程序都是用Objective-C语言编写的，它所使用的技术可以追溯到20世纪80年代，如今这些技术已经演化成一个功能强大的工具集。本书假设你对C语言或其他一般编程语言有了一定的了解。

希望你能从本书中获得乐趣！

^① 作者在写这本书时用的是4.2之后的版本，而译者在翻译此书时，Xcode最新的版本是4.4.1，二者差别不大，读者无需担心兼容问题。——译者注