

OBJECTIVE-C PROGRAMMING



OBJECTIVE-C 编程

THE BIG NERD RANCH GUIDE

[美] Aaron Hillegass 著
夏伟频 译 李骏 审校



华中科技大学出版社
<http://www.hustp.com>

本书仅提供部分阅读，如需完整版，请联系QQ: 2011705918

提供各种IT类, 其它类书籍pdf下载，如有需要，请QQ:2011705918

注：链接至淘宝，整理那么多资料也不容易，请多多见谅！非诚勿扰！

[点击购买完整版](#)

提供各种IT类书籍pdf下载，如有需要，请 **QQ: 2011705918**

注：链接至淘宝，不喜者勿入！整理那么多资料也不容易，请多多见谅！非诚勿扰！

更多资源请点击

想开发iOS和Mac应用吗？本书是很好的入门教材，可以帮助读者从应用的使用者变成应用的开发者。

本书是畅销书作者Aaron Hillegass编写的唯一一本入门级的编程图书，内容涵盖C、Objective-C和最基础的编程知识。只有先学习这些知识，才能充分地应用Apple的编程技术。

本书内容兼容Xcode 4.2、iOS 5和Mac OS X 10.7 (Lion)，内容简练易懂。作者尽可能地帮助读者理解“自己在做什么”，以及“为什么要这样做”，培养程序员式的思维方式。

本书涵盖的知识点：

- 变量、循环和函数等基础知识
- 对象、类、方法和消息
- 指针、地址和内存管理
- 使用Apple提供的开发文档及工具
- Foundation框架提供的类
- ARC与retain循环
- 属性 (property)
- Block对象 (block)
- 范畴 (Category)
- 委托 (delegate)、目标-动作 (target-action) 和通告 (notification) 等设计模式

OBJECTIVE-C 编程



Aaron Hillegass 曾就职于NeXT公司和Apple公司，他有近20年的Objective-C编程、Cocoa编程和iOS编程教学经验。

Aaron是《Cocoa编程》的作者，并与他人合著了《iOS编程》。这两本畅销书获得了来自全球各地读者的好评。

2001年，Aaron创办了Big Nerd Ranch公司，开始提供强化式编程课程，为学员提供集中的、没有干扰的学习环境。

Big Nerd Ranch 是一家软件开发技术培训公司。从2001年开始，Big Nerd Ranch公司通过公开的新手培训课程、封闭的公司培训以及不断推出的编程书籍，帮助很多学员学习并掌握了编程语言。Big Nerd Ranch还为多家客户提供咨询服务，帮助它们调整移动平台战略，开发新颖实用的移动应用和桌面应用。

策划编辑：徐定翔
责任编辑：陈元玉



ISBN 978-7-5609-8323-3



定价：58.00元

上架建议：编程语言/iOS开发

Objective-C 编程

[美] Aaron Hillegass 著

夏伟频 译

李 骏 审校

华中科技大学出版社

中国·武汉

内 容 简 介

本书主要介绍 Objective-C 编程语言和基本的 iOS/ Mac 开发知识。首先讲解变量、条件语句、循环结构等基础编程概念;接着用浅显易懂的语言描述 Objective-C 和 Foundation 的知识,包括 Objective-C 基本语法、Foundation 常用类、内存管理等;最后手把手地教读者编写完整的、基于事件驱动的 iOS/Mac 应用。作者还穿插介绍了 Objective-C 的高级内容,包括属性、范畴和 Block 对象等知识。全书篇幅精炼、内容清晰,适合无编程经验的读者入门学习。

Authorized translation from the English language edition, entitled *Objective-C Programming: The Big Nerd Ranch Guide* 1st Edition, 9780321706287 by Aaron Hillegass, published by The Big Nerd Ranch (Aaron Hillegass), Copyright © 2011 The Big Nerd Ranch (Aaron Hillegass)

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Huazhong University of Science and Technology Press, Copyright © 2012 Huazhong University of Science and Technology Press.

湖北省版权局著作权合同登记 图字:17-2012-144 号

图书在版编目(CIP)数据

Objective-C 编程/(美) Aaron Hillegass 著;夏伟频 译;李骏 审校. —武汉:华中科技大学出版社,2012.9

ISBN 978-7-5609-8323-3

I. O… II. ①A… ②夏… ③李… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2012)第 200684 号

Objective-C 编程

[美] Aaron Hillegass 著

夏伟频 译 李 骏 审校

策划编辑:徐定翔

责任校对:朱 霞

责任编辑:陈元玉

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321915

录 排:华中科技大学惠友文印中心

印 刷:湖北新华印务有限公司

开 本:787mm×960mm 1/16

印 张:17.75

字 数:485 千字

版 次:2012 年 9 月第 1 版第 1 次印刷

定 价:58.00 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

翻译说明

本书的术语、词汇翻译优先使用 Apple 的正式译法，参考了 Apple 中国官方网站、iOS 设备中文版使用手册、中文开发文档、中文软件系统界面的译法。有些翻译无固定译法可循，特做如下说明。

- 本书将 iOS 4.0 加入的语言特性 Blocks 翻译为“Block 对象”，因为 Blocks 是简称，其正式的名称为 Block Object。
- Category 翻译为“范畴”。“范畴”是 Apple 的官方开发文档选用的译法。
- Override 有多种译法，本书翻译为“覆盖”。

iOS SDK 升级频繁，请读者留意本书中文版的网站以获取最新的安装流程介绍，以及其他更新信息。此外，该网站也会列出详细的中英文术语对照表，并提供部分译法的出处。欢迎访问：<http://www.iosprogrammingbook.com>。

致谢

Acknowledgments

很荣幸能与这么多优秀的人一起完成本书的写作。其中几位花费了大量的时间和精力，目的是想为读者奉上优秀的作品。借此机会向他们表示感谢。

- Mikey Ward 编写了本书的多个章节，包括：第一个 iOS 应用、第一个 Cocoa 应用和 Block 对象。真应该将他的名字也登在封面上。
- Objective-C 培训课程的其他教员一直在向我们提供建议和修改意见。他们是 Scott Ritchie、Mark Fenoglio、Brian Hardy、Christian Keur 和 Alex Silverman。
- 本书编辑 Susan Loper，将原本晦涩难懂的文字润色为通顺流畅的好文。
- 几位技术审校者帮助我们找到并修正了书中的多处瑕疵，他们是 James Majors、Mark Dalrymple、Scott Steinman、Bart Hoffman、Bolot Kerimbaev 和 Nate Chandler。
- Ellie Volckhausen 设计了封面。
- 来自 IntelligentEnglish.com 的 Chris Loper，设计并制作了本书的 EPUB 版本和 Kindle 版本。
- 来自 Pearson Technology Group 的团队，耐心地指导我们走完了整个出版流程。

目录

Table of Contents

第 1 部分 入门	1
第 1 章 读者与本书	3
1.1 C 与 Objective-C	3
1.2 主要内容及阅读方式	4
1.3 程序员要走的道路	4
第 2 章 第一个程序	7
2.1 安装 Apple 开发工具软件	7
2.2 Xcode 入门	7
2.3 从哪里开始编写代码	9
2.4 如何运行程序	13
2.5 程序是如何产生的	14
2.6 不要半途而废	15
第 2 部分 如何编程	17
第 3 章 变量与类型	19
3.1 类型	19
3.2 使用变量的程序	20
3.3 练习	22
第 4 章 if 和 else	23
4.1 布尔变量	24
4.2 else if	25
4.3 深入学习：条件运算符（三元运算符）	25
4.4 练习	26
第 5 章 函数	27
5.1 应该在何时使用函数	27
5.2 如何编写并使用函数	27
5.3 函数间是如何协同工作的	30
5.4 局部变量、帧和栈	32
5.5 递归	34

5.6 通过调试器查看帧	35
5.7 return.....	37
5.8 全局变量与静态变量	39
5.9 练习	40
第 6 章 数字	41
6.1 printf().....	41
6.2 整数	42
6.3 浮点数	46
6.4 练习	47
第 7 章 循环	49
7.1 while 循环	49
7.2 for 循环	50
7.3 break	51
7.4 continue	52
7.5 do-while 循环	53
7.6 练习	54
第 8 章 地址与指针	55
8.1 获取地址	55
8.2 用指针保存地址	56
8.3 通过地址访问数据	57
8.4 不同类型的数据所占用的字节大小	57
8.5 NULL	58
8.6 指针声明的代码规范	59
8.7 练习	59
第 9 章 通过引用传递.....	61
9.1 编写通过引用传递参数的函数	62
9.2 不要对 NULL 取值	64
第 10 章 结构	65
10.1 练习	66
第 11 章 堆	69
第 3 部分 Objective-C 与 Foundation	73
第 12 章 对象	75
12.1 创建并使用对象	75
12.2 消息详解	77

12.3	内存中的对象	79
12.4	id	79
12.5	练习	80
第 13 章	再谈消息	81
13.1	消息的嵌套发送	81
13.2	多个实参	82
13.3	向 nil 发送消息	82
13.4	练习	83
第 14 章	NSString	85
14.1	练习	86
第 15 章	NSArray	87
15.1	NSMutableArray	89
15.2	练习	90
第 16 章	开发文档	93
16.1	参考页	94
16.2	快速帮助	96
16.3	其他文档功能和开发资源	98
第 17 章	第一个自定义类	101
17.1	存取方法	103
17.2	用点号调用存取方法	104
17.3	属性	105
17.4	self	106
17.5	多个实现文件	106
17.6	练习	106
第 18 章	继承	109
18.1	覆盖方法	112
18.2	super	113
18.3	练习	113
第 19 章	对象实例变量	115
19.1	对象所有权与 ARC	117
19.2	练习	123
第 20 章	避免内存泄露	125
20.1	Retain 循环	127
20.2	弱引用	129

20.3	弱引用的自动置零特性	130
20.4	深入学习：手动引用计数和 ARC 历史	131
第 21 章	Collection 类	135
21.1	NSArray/NSMutableArray	135
21.2	NSSet/NSMutableSet	138
21.3	NSDictionary/NSMutableDictionary	140
21.4	C 语言基本类型	142
21.5	Collection 对象与 nil	142
21.6	练习	143
第 22 章	常量	145
22.1	预处理指令	145
22.2	#include 与#import	146
22.3	#define	146
22.4	全局变量	147
22.5	比较#define 与全局变量	149
第 23 章	通过 NSString 和 NSData 将数据写入文件	151
23.1	将 NSString 对象写入文件	151
23.2	NSError	152
23.3	通过 NSString 读取文件	153
23.4	将 NSData 对象所保存的数据写入文件	154
23.5	从文件读取数据并存入 NSData 对象	155
第 24 章	回调	157
24.1	目标-动作对	157
24.2	辅助对象	160
24.3	通告	163
24.4	如何选择	164
24.5	回调与对象所有权	164
第 25 章	协议	167
第 26 章	Property List 格式	171
26.1	练习	173
第 4 部分	由事件驱动的应用	175
第 27 章	第一个 iOS 应用	177
27.1	创建 iTahDoodle	177
27.2	BNRAppDelegate	179

27.3	添加 C 语言辅助函数	180
27.4	iTahDoodle 中的对象	181
27.5	Model-View-Controller	182
27.6	应用委托对象	183
27.7	设置视图	184
27.8	为 UITableView 对象提供数据	186
27.9	添加新任务	189
27.10	深入学习: main() 的作用	190
第 28 章	第一个 Cocoa 应用	191
28.1	编辑 BNRDocument.h	192
28.2	Interface Builder 初探	193
28.3	编辑 BNRDocument.xib	194
28.4	创建关联	198
28.5	再探 MVC	202
28.6	编辑 BNRDocument.m	202
28.7	练习	204
第 5 部分	Objective-C 高级主题	205
第 29 章	init	207
29.1	编写 init 方法	207
29.2	实现基本的 init 方法	208
29.3	在 init 中使用存取方法	209
29.4	带实参的 init 方法	210
29.5	禁用 init 方法	215
第 30 章	属性	217
30.1	属性的特性	218
30.2	KVC	221
第 31 章	范畴	225
第 32 章	Block 对象	227
32.1	定义 Block 对象	227
32.2	使用 Block 对象	228
32.3	typedef	233
32.4	返回值	233
32.5	内存管理	234
32.6	基于 Block 的编程是大势所趋	235

32.7 练习	235
第 6 部分 C 语言高级主题	237
第 33 章 位运算	239
33.1 按位或	240
33.2 按位与	241
33.3 其他位运算符	242
33.4 用 enum 定义位掩码	245
33.5 占用更多字节的整数类型	245
33.6 练习	245
第 34 章 C 字符串	247
34.1 char	247
34.2 char *	248
34.3 string literal	250
34.4 C 字符串和 NSString 对象的相互转换	251
34.5 练习	252
第 35 章 C 数组	253
第 36 章 命令行参数	257
第 37 章 switch 语句	261
结束语	263
索引	265

第 1 部分 入门

Part I Getting Started

第1章

读者与本书

You and This Book

想开发 iOS 或 Mac OS X 应用，却苦于编程经验寥寥？甚至一行代码也没写过？也许已经有人向读者推荐过其他书籍（《iOS 编程》和《Cocoa 编程》）。但是，这几本书是为有编程经验的读者准备的。那么是否还有其他选择？

在类似的书籍中，也许读者应该选读本书。为什么？因为本书的作者长期从事 iOS 和 Mac 应用开发方面的教学工作，所以很清楚读者在初学阶段要了解哪些知识。此外，作者还做了大量的工作，去芜存菁，使本书的内容更精炼。

本书的“教学方法”也有些特别。作者不是简单地向读者传授 Objective-C 的语法知识，而是向读者揭示编程是怎么回事，并介绍资深程序员在编程时是如何思考的。

有鉴于此，本书开篇就会涉及多个有分量的概念，因此读者读起来不会很轻松。此外，几乎每个概念都附有实验示例。这种“先学习概念，然后马上动手练习的组合”是学习编程的最佳方法。

1.1 C 与 Objective-C

C and Objective-C

运行程序时，计算机会将相应的文件从文件系统拷入内存（RAM），并执行其中的指令。因为这些指令难以理解，所以人们通过某种编程语言来编写计算机程序。最底层的编程语言是汇编语言（assembly code）。使用汇编语言时，需要描述 CPU（计算机的大脑）必须执行的每一个步骤。然后通过某种汇编程序（assembler）将这些代码转成机器码（machine code，计算机的“母语”）。

汇编语言冗长难懂，而且不同的 CPU 所用的汇编语言也不同（例如，最新的 iMac 的“大脑”和旧的 PowerBook 的“大脑”就有很大的不同）。也就是说，为了能在不同类型的计算机上运行某个程序，就要重写汇编代码。

为了能够方便地移植代码以适应不同类型的计算机，工程师们开发出了高级语言（high-level language）。使用高级语言时，程序员可以不用考虑 CPU 类型，使用统一的指令即可。编译器（compiler，也是一种程序）会将用高级语言写成的代码转化成高度优化的、针对某种 CPU 的机器码。C 语言就是高级语言。程序员可以用 C 语言编写程序，然后通过 C 编译器将 C 代码转成机器码。

C 语言诞生于 20 世纪 70 年代初期，由 AT&T 的工程师开发。早期的 Unix 操作系统（Mac OS X 和 Linux 都源自 Unix）都是用 C 语言编写的，并配合少量的汇编代码以处理底层操作。而 Windows 操作系统大部分也是用 C 语言编写的。

Objective-C 语言以 C 语言为基础，加入了对面向对象编程（object-oriented programming）的支持。在 Apple 的 iOS 和 Mac OS X 操作系统上运行的原生应用，都要用 Objective-C 语言编写。

1.2 主要内容及阅读方式

How this book works

通过阅读本书，读者将对 C 语言和 Objective-C 语言有一个足够的了解，以便能够开始学习 Mac 应用或 iOS 应用的开发。

为什么要先学习 C 语言？这是因为要成为一名好的 Objective-C 程序员，就要对 C 语言有深入的理解。另外，很多看似复杂的 Objective-C 问题，其实都源自 C 语言中的简单概念。本书会先通过 C 语言介绍某个概念，然后引导读者掌握 Objective-C 中的等价概念。

本书的初衷是要读者在阅读的同时上机实验。先读懂概念，然后上机实验，验证概念。这些实验是必需的，读者要完成这些实验才能真正读懂本书。学习编程的最佳途径是输入代码、查错、修正代码并最终熟悉某种语言的语法模式。仅仅靠阅读代码并在理论上理解概念是不够的，也无助于编程技能的提升。

如果读者想要做更多的练习，则可以参考每章结尾的“练习”部分。这些练习可以帮助读者磨炼技巧，并对之前所学的知识更有自信。强烈建议读者尽可能多地完成这些练习。

此外，部分章节的结尾处还有“深入学习”部分，会针对相应章节的内容作更深入的介绍。读者可以根据自身情况和兴趣选读这部分内容。希望这部分内容能对读者有所帮助。

Big Nerd Ranch 架设了一个论坛（英文），供读者讨论本书和书中的练习。网址是：
<http://forums.bignerdranch.com/>。

如果读者会盲打，那么无论是阅读本书还是编写程序都会轻松很多。盲打除了能提高输入速度，还能让读者在输入内容时看着屏幕或书，而不是键盘。这样，要发现输入错误也会容易很多。盲打是一项可受用终身的技能。

1.3 程序员要走的道路

How the life of a programmer works

读者会阅读本书，说明您已经决意成为一名程序员。那么读者须要知道等待自己的是一条怎样的道路。

程序员所要走的道路是没有终点的，要不停地学习新知识，这样才能适应不断变化的 IT 行业。这里的“学习新知识”其实是“和自己不懂的东西作斗争”的婉转说法。因

为现在的软件都太复杂了，所以即使是程序员熟悉的技术，有时也许会要花一整天的时间才能弄清楚某个问题出在哪里。

编程是困难的。很多专业程序员习惯了不间断地解决难题，日复一日，不让挫败感击垮自己。这也是读者应该学会的一项技能。如果读者对“程序员要走的是一条怎样的道路”感兴趣，并想了解更多关于现代软件项目的信息，本书作者力荐由 Scott Rosenberg 撰写的《Dreaming in Code》一书。

下面进入正题，编写第一个程序。

第2章

第一个程序

Your First Program

第1章介绍了本书的内容是如何组织的，下面介绍如何为 Mac、iPhone 和 iPad 编写程序。为此，读者将要做以下工作。

- 安装 Apple 的 Developer Tools（开发工具软件）。
- 通过这些工具软件创建一个简单的项目。
- 学习如何使用这些工具软件，使项目能够正常运作。

学习完本章后，读者将会完成自己的第一个 Mac 程序。

2.1 安装 Apple 开发工具软件

Installing Apple's developer tools

要编写 Mac OS X（运行于 Macintosh 计算机的操作系统）或 iOS（运行于 iPhone 和 iPad 的操作系统）应用，要使用 Apple 提供的开发工具。读者可以通过 <http://developer.apple.com/> 下载这些工具，或者在 Mac App Store 上购买。

安装完成后，可以在硬盘的根目录找到 /Developer 目录。读者要使用该目录中的工具软件来开发 Mac OS X 桌面应用和 iOS 移动设备应用。

本书只着重介绍一个开发工具：Xcode。读者可以在 /Developer/Applications 目录下找到该软件（建议将 Xcode 拖曳到 Dock 上，方便将来使用）。注：如果读者是通过 App Store 下载并安装 Xcode 的，那么可以在应用程序（/Applications）目录下找到该软件。

2.2 Xcode 入门

Getting started with Xcode

Xcode 是 Apple 为开发者提供的集成开发环境（Integrated Development Environment）。Xcode 可以完成几乎所有的开发任务，例如编写、构建和运行应用。

术语注释：凡是能在计算机上执行的都称为程序（program）。有些程序有图形化的用户界面，可以将这些程序称为应用（application）。

有些程序没有图形化的用户界面，并且会在后台长时间地运行，这些程序称为守护进程（daemon）。虽然守护进程一词听上去很可怕（英文单词 daemon 有恶魔的意思），但这些程序是很有用的。读者此刻使用的 Mac 计算机可能正在运行着大约 60 个守护进程，这些守护进程随时准备处理任务。以其中的一个守护进程 pboard 为例，当读者执行复制和粘贴操作时，pboard 守护进程会保存读者拷贝的数据。

还有一些没有图形化用户界面的程序，只能在终端（terminal）下运行，这类程序称为命令行工具（command-line tools）。本书为了能将精力集中在编程的核心问题上，列举的大部分例子都是命令行工具。这样，就不必为这些程序创建和管理用户界面。

下面介绍如何用 Xcode 创建一个简单的命令行工具，以便读者能对整个开发流程有一个简单的了解。

编写程序时，要创建并编辑一组文件。Xcode 通过项目（project）来管理这些文件。启动 Xcode，从 File 菜单选择 New，然后选择 New Project...

Xcode 提供了一系列的模板，以帮助开发者创建项目。开发者只要根据开发的程序类型，选择相应的模板即可。选择窗口左侧 Mac OS X 栏下的 Application，然后选择右侧的 Command Line Tool，如图 2-1 所示。

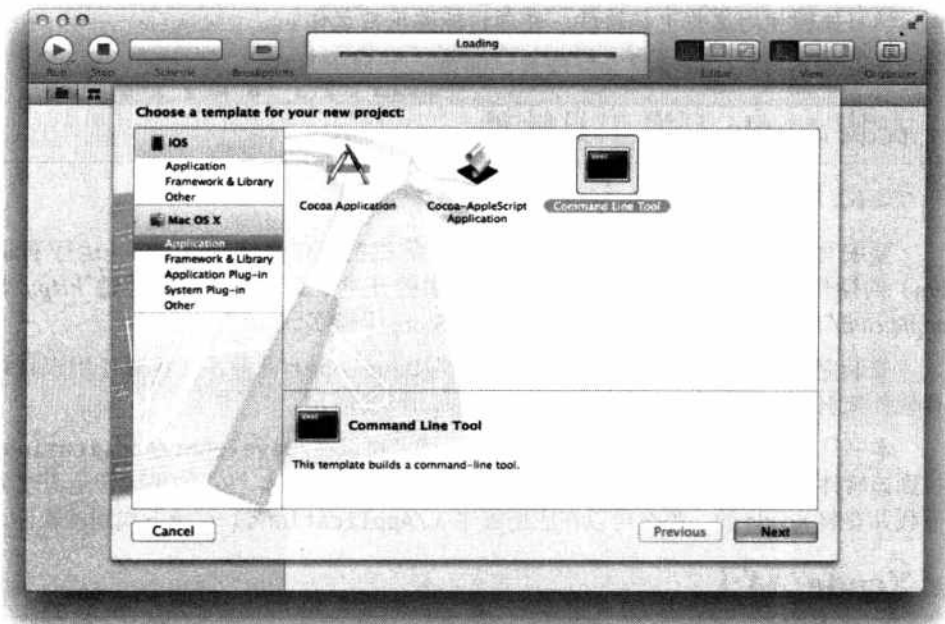


图 2-1 选择模板

单击 Next 按钮。

将新项目命名为 AGoodStart。虽然公司标识（company identifier）对本书中的例子程序无关紧要，但是这里必须要填写 company identifier 后才能继续。读者可以使用 BigNerdRanch 或别的名称。稍后会用 C 语言编写 AGoodStart 程序，因此要在弹出式菜单 Type 中选择 C，如图 2.2 所示。最后，确定勾选了 Use Automatic Reference Counting 选择框。

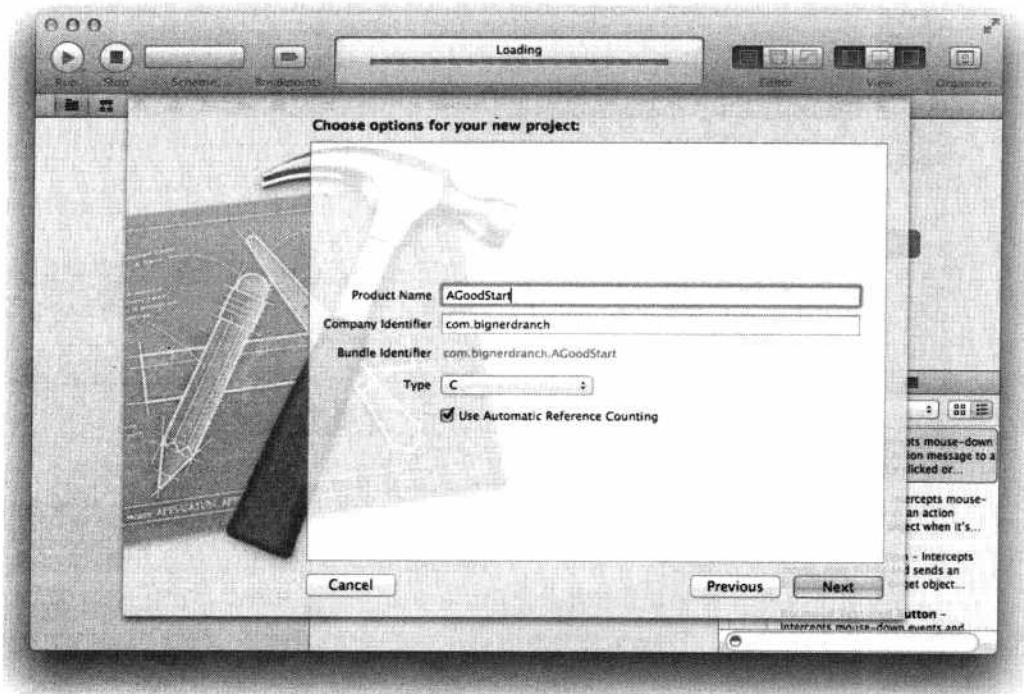


图 2.2 设置选项

单击 **Next** 按钮。

接着选择存放项目的目录。因为本例不使用版本控制（version control）功能，所以要取消选项 **Create local git repository for this project**。最后单击 **Create** 按钮。

在接下来的几章中，读者会多次创建同一类型的项目。当读者看到“创建一个名称为‘这里是程序名’的 C 命令行工具”时，请重复以上步骤。

（为什么选择 C 语言？别忘了，Objective-C 是基于 C 语言的。在 Objective-C 中，有部分知识点要在理解某些 C 语言特性后才能弄明白。）

2.3 从哪里开始编写代码

Where do I start writing code?

创建项目后，Xcode 会打开一个窗口，供读者查看并维护 **AGoodStart** 项目，如图 2-3 所示。

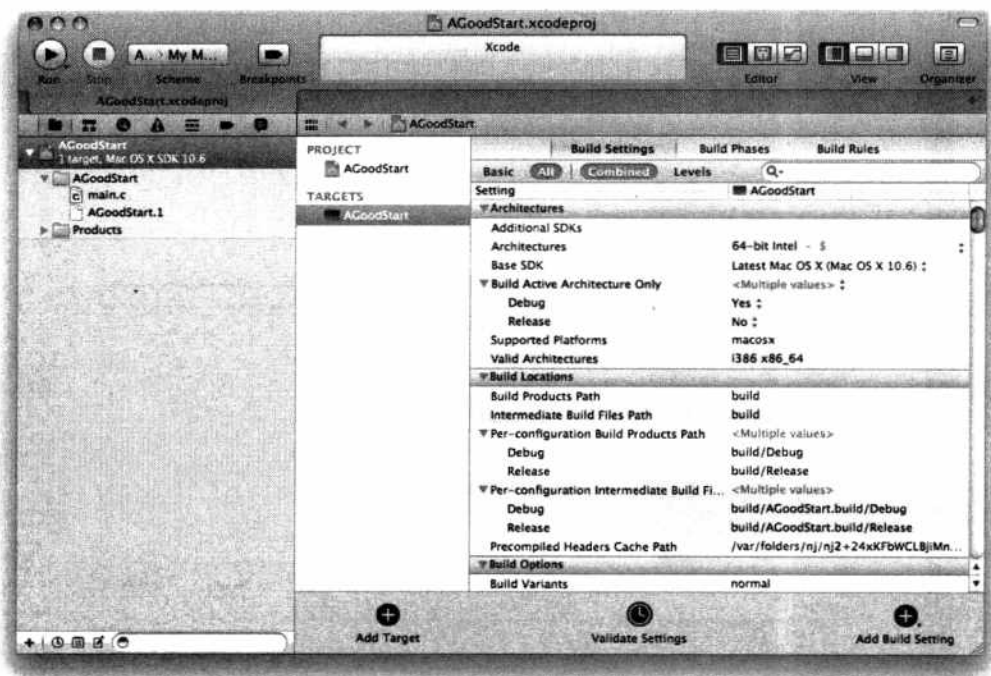


图 2-3 新创建的 AGoodStart 项目

Xcode 所显示的这个窗口包含很多细节信息，例如，生成的应用可以在哪些版本的 Mac OS X 上运行、编译代码时会使用的配置信息及项目当前的本地化设置。下面暂时忽略这些细节，先简单介绍如何开始编写代码。

在左侧面板顶部找到名为 `main.c` 的文件，单击选中该文件（如果读者找不到 `main.c`，则可以试试单击 `AGoodStart` 目录左侧的三角按钮，展开其下的内容），如图 2-4 所示。