

NITE 国家信息技术紧缺人才培养工程指定教材

教材+教案+授课资源+考试系统+题库+教学辅助案例

一站式IT就业应用系列教程

Objective-C 程序设计 案例教程

黑马程序员/ 编著



添加QQ或微信号208695827，获取教学答案、源码，抢“助学金红包”。

本书涵盖了Objective-C的基础知识和面向对象编程，提供了7个阶段案例，10套配套习题。

提供免费教学资源，包括精美教学PPT、1000道测试题、长达28小时的教学视频等。

清华大学出版社

Objective-C 程序设计 案例教程

黑马程序员/编著



清华大学出版社
北京

内 容 简 介

本书是经典的 Objective-C 指导教材,在《Objective-C 入门教程》的基础上进行了全面升级,不仅细化、补充了很多新知识,而且针对主要章节都设计了阶段案例和配套习题,帮助读者巩固知识,加深学习印象。

全书共分为 10 章。第 1 章主要带读者认识 Objective-C 这门语言,并尝试编写第一个 Objective-C 程序,激发读者对 Objective-C 的学习兴趣;第 2 章和第 3 章讲解了面向对象的编程思想;第 4 章针对 Objective-C 特有的语法进行讲解;第 5 和第 6 章讲解的是 Objective-C 中的内存管理机制,包括 MRC 和 ARC;第 7~9 章讲解了 Objective-C 中的分类、延展、block 代码块和协议;第 10 章针对 Foundation 框架进行讲解,让读者为后面 iOS 开发奠定扎实的基础。

本教材附有配套视频、源代码、习题、教学课件等资源,为了帮助初学者更好地学习本教材中的内容,还提供了在线答疑,希望得到更多读者的关注。

本书既可作为高等院校本科、专科计算机相关的程序设计课程教材,也可以作为 iOS 技术基础的培训教材,是一本适合广大移动开发编程初学者的入门级教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Objective-C 程序设计案例教程/黑马程序员编著. —北京:清华大学出版社,2017
ISBN 978-7-302-47610-8

I. ①O… II. ①黑… III. ①C 语言—程序设计 教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 154210 号

责任编辑:袁勤勇

封面设计:马 丹

责任校对:白 蕾

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm

印 张:16.75

字 数:402 千字

版 次:2017 年 10 月第 1 版

印 次:2017 年 10 月第 1 次印刷

印 数:1~2000

定 价:45.00 元

产品编号:075646-01

序 言

传智播客和“黑马程序员”

江苏传智播客教育科技股份有限公司(简称传智播客)是一家专门致力于培养高素质软件开发人才的科技公司,“黑马程序员”是传智播客旗下的高端 IT 教育品牌。

“黑马程序员”的学员多为大学毕业后想从事 IT 行业,但各方面条件还不成熟的年轻人。“黑马程序员”的学员筛选制度非常严格,除包括严格的技术测试、自学能力测试外,还包括性格测试、压力测试、品德测试等。百里挑一的残酷筛选制度确保了学员质量,降低了企业的用人风险。

自“黑马程序员”成立以来,教学研发团队一直致力于打造精品课程资源,不断在产、学、研三个层面创新自己的执教理念与教学方针,并集中“黑马程序员”的优势力量,有针对性地出版了计算机系列教材五十多种,制作了教学视频数十套,发表各类技术文章数百篇。

“黑马程序员”不仅斥资研发 IT 系列教材,还为高校师生提供以下配套学习资源与服务。

为大学生提供的配套服务:

1. 专注的辅助学习平台“博学谷”(http://yx.boxuegu.com),专业老师在线为学生答疑解惑。

2. 针对高校学生在学习过程中存在的压力等问题,我们还面向大学生量身打造了“播妞”。“播妞”不仅致力推行快乐学习,还会有定期的助学红包雨。读者可添加“播妞”微信/QQ: 208695827。

3. 高校学生也可扫描右方二维码,加入“播妞”粉丝团,获取最新学习资源,与“播妞”一起快乐学习。

为 IT 教师提供的配套服务:

针对高校教学,“黑马程序员”为 IT 系列教材精心设计了“教案+授课资源+考试系统+题库+教学辅助案例”的系列教学资源,高校老师请关注码大牛老师微信/QQ: 2011168841,获取教材配套资源,也可以扫描右方二维码,加入专为 IT 教师打造的师资服务平台——“教学好助手”,获取“黑马程序员”最新教师教学辅助资源相关动态。



前言

Objective-C,通常写作 ObjC 或 OC,较少写作 Objective C 或 Obj-C,它在 C 语言的基础上进行扩充,是支持面向对象的一门语言。Objective-C 的流行归功于 iPhone 的成功,它可以编写的应用程序如下:

- iOS 操作系统
- iOS 应用程序
- Mac OS X 操作系统
- Mac OS X 上的应用程序

为什么写本书

2014 年,我们出版了一本《Objective-C 入门教程》,该书站在初学者的角度,以形象的比喻、实用的案例、通俗易懂的语言,详细讲解了 Objective-C 语言,受到广大读者的一致好评。

本书作为《Objective-C 入门教程》的升级版,秉承之前教材的优点,不仅对开发环境和开发工具有所更新,而且在知识内容上也有所更新:

- 新增 block 代码块。作为 Objective-C 最新最强大的工具,本书通过对 block 代码块的定义和使用进行了详细讲解。block 是苹果公司推荐的类型,效率高,可以帮助我们组织独立的代码段,并提高复用性和可读性。
- 内存管理。通过 MRC 到 ARC 的转变过程,深刻帮助读者理解 Objective-C 的内存管理机制,轻松搞定任何与内存管理相关的问题。

如何使用本书

本书以全新的 OS X 10.11 为平台,以 Xcode 8.1 为开发工具,全面介绍了 Objective-C 的基本知识。考虑到本书使用对象的不同,给予以下建议:

课堂教学,建议采用案例驱动的方式来讲授,让学生在动手完成“案例”的过程中,培养学生分析问题、解决问题的能力,使学生可以直观、深刻地学会 Objective-C 开发技能。

建议自主学习者勤思考、勤练习、勤检测。任何有疑惑的地方都可以向“问答精灵”咨询,每个知识点对应的案例都要独立完成,最后通过每章配套的习题进行自我检测。

本教材共有 10 章,具体内容如下。

- 第 1 章首先介绍了 Objective-C 语言的特点、发展历史以及未来的趋势,然后讲解了开发环境的搭建,包括系统环境和开发工具,并带领大家开发一个 Objective-C 程

序,对程序的代码及编译进行讲解,最后讲解帮助文档的使用。

- 第2章讲解了 Objective-C 面向对象的概念,包括类和对象、类的设计与定义、对象的创建和访问、方法的使用等内容,帮助大家学会使用面向对象的编程思想编写简单的程序。除此之外,本章还介绍 self、匿名对象和异常,并通过一个阶段案例,综合考查读者对项目需求的分析能力以及对面向对象基本知识的掌握程度。
- 第3章主要介绍面向对象的三大特征——封装、继承和多态,介绍访问修饰符的特点、里氏替换原则以及 description 方法的原理。通过本章的学习,希望读者能够理解并且掌握封装、继承和多态这三个特征,并且可以灵活运用,为后面的学习奠定扎实的基础。
- 第4章主要介绍 Objective-C 的特有语法,包括点语法、@property、@synthesize 和 @property 的增强使用,还介绍静态类型、动态类型和构造方法的内容。希望读者认真学习本章内容,仔细体会 Objective-C 特有语法的特点和用法。
- 第5章主要介绍内存管理(MRC 机制)的知识,包括引用计数器的概念、单个对象和多个对象的内存管理,以及@property 参数的使用。大家应该掌握 MRC 内存管理的原理,便于更好地理解 ARC 机制的知识。
- 第6章主要介绍内存管理(ARC 机制)的知识,包括概念、单个对象的内存管理、@property 参数、循环引用,以及 ARC 和 MRC 的相互兼容问题。大家应该掌握 ARC 内存管理的原理,及时管理好程序的内存。
- 第7章主要讲解分类的相关知识,包括分类的简单实用、非正式协议和延展。希望读者在以后的学习中能够区分分类的不同使用情景,并灵活加以运用。
- 第8章主要介绍 block 代码块,包括 block 语法、block 的简化、block 内部访问外部变量、block 作为函数的参数和返回值。block 代码块这个知识点不难,并且是 iOS 开发经常用到的知识点,希望读者好好学习并掌握。
- 第9章主要讲解 Objective-C 中的协议,包括协议的声明、协议的遵守、协议的继承等。虽然协议的相关内容不多,但使用频率非常高,希望大家深刻理解协议的好处,并在后面的开发中灵活运用。
- 第10章主要讲解 Foundation 框架中的常见对象及其这些对象的常见操作,包括字符串、数组、字典、文件操作、日期处理等相关对象。通过本章的学习,希望读者独立使用 Foundation 框架开发应用程序。

在学习过程中,读者一定要亲自实践案例中的代码。如果不能完全理解书中所讲的知识,读者可以登录博学谷平台,通过平台中的教学视频进行深入学习。学习完一个知识点后,要及时在博学谷平台上进行测试,以巩固学习内容。另外,如果读者在理解知识点的过程中遇到困难,建议不要纠结于某个小点,可以先往后学习,通常来讲,看到后面对该知识点的讲解或者其他小节的内容后,前面看不懂的知识点一般就能理解了。如果读者在动手练习的过程中遇到问题,建议多思考,厘清思路,认真分析问题发生的原因,并在问题解决后多总结。

致谢

本教材的编写和整理工作由传智播客教育科技有限公司完成,主要参与人员有吕春林、高美云、王晓娟、刘传梅、尹桥印、潘星、薛蒙蒙、郑瑶瑶、齐瑞华、郭敬楠等,全体人员在近一年的编写过程中付出了很多辛勤的汗水,在此一并表示衷心的感谢。

意见反馈

尽管我们尽了最大的努力,但教材中难免会有不妥之处,欢迎各界专家和读者朋友来信、来函提出宝贵意见,我们将不胜感激。阅读本书时,如发现任何问题或有不认同之处,可以通过电子邮件(itcast_book@vip.sina.com)与我们取得联系。

黑马程序员

2017年5月于北京

目 录

第 1 章 Objective-C 入门	1
1.1 Objective-C 概述	1
1.2 搭建开发环境	2
1.2.1 开发环境要求	2
1.2.2 系统环境搭建	2
1.2.3 Xcode 开发工具搭建	3
1.3 开发第一个 Objective-C 程序	4
1.3.1 使用 Xcode 开发 Objective-C 程序	4
1.3.2 Objective-C 程序代码分析	7
1.3.3 Objective-C 源码、编译代码和 可执行文件	8
1.4 帮助文档的使用	9
1.4.1 帮助文档的下载安装	9
1.4.2 帮助文档的使用	9
1.5 本章小结	12
1.6 本章习题	12
第 2 章 面向对象编程(上)	13
2.1 面向对象概述	13
2.2 类和对象	14
2.2.1 类和对象之间的关系	14
2.2.2 类的设计	15
2.2.3 类的定义	16
2.2.4 使用 Xcode 创建一个类	17
2.2.5 类的对象的创建	22
2.2.6 访问对象的属性	24
2.3 方法	25
2.3.1 方法的声明实现和调用	25
2.3.2 方法实现中直接访问属性	27

专属于老师及学生的在线教育平台
yx.boxuegu.com

让 IT 教学更简单

教师获取教材配套资源

教案

授课资源

考试系统

在线题库

教学辅助
案例

添加微信/QQ

2011168841

让 IT 学习更有效

学生获取课后作业习题答案及配套源码

添加播妞微信/QQ

208695827

学习问答精灵: ask.boxuegu.com

更多学习视频: dvd.boxuegu.com



专属大学生的圈子

2.3.3	对象作为方法的参数	28
2.3.4	对象作为方法的返回值	31
2.4	self 的使用	32
2.4.1	在对象方法中使用 self	32
2.4.2	在类方法中使用 self	35
2.5	匿名对象	37
2.6	异常	37
2.7	阶段案例——猜拳游戏	38
2.8	本章小结	43
2.9	本章习题	44

第 3 章 面向对象编程(下)

46

3.1	面向对象的三大特征	46
3.2	封装	47
3.2.1	setter 的封装	47
3.2.2	getter 的封装	49
3.3	static 关键字	50
3.4	继承	51
3.4.1	继承的简单使用	51
3.4.2	继承的特点	53
3.4.3	方法的重写	56
3.4.4	super 关键字	57
3.5	访问修饰符	59
3.6	里氏替换原则	59
3.7	多态	60
3.8	description 方法	62
3.9	阶段案例——计算器	63
3.10	本章小结	67
3.11	本章习题	67

第 4 章 Objective-C 特有语法

69

4.1	点语法	69
4.2	@property	72
4.3	@synthesize	73
4.4	@property 增强	75
4.5	静态类型与动态类型	76
4.6	动态类型检测	78
4.7	构造方法	80
4.7.1	构造方法概述	80

4.7.2 重写 init 构造方法	81
4.7.3 自定义构造方法	82
4.8 阶段案例——购票系统	84
4.9 本章小结	92
4.10 本章习题	93
第 5 章 内存管理	99
5.1 内存管理概述	99
5.1.1 引用计数器	99
5.1.2 第一个 MRC 程序	101
5.2 单个对象的内存管理	105
5.3 多个对象的内存管理(setter 方法内存管理)	106
5.3.1 setter 方法内存管理(一)	109
5.3.2 setter 方法内存管理(二)	111
5.3.3 setter 方法内存管理(三)	115
5.4 @property 参数概述	117
5.4.1 @property 参数之 atomic 与 nonatomic	118
5.4.2 @property 参数之 retain 与 assign	118
5.4.3 @property 参数之 readwrite 与 readonly	121
5.4.4 @property 参数之 getter 与 setter	121
5.5 自动释放池	126
5.6 阶段案例——微博(MRC)	127
5.7 本章小结	133
5.8 本章习题	133
第 6 章 ARC 机制	138
6.1 ARC 机制概述	138
6.2 ARC 机制下单个对象的内存管理	139
6.3 @property 中的 strong 和 weak	140
6.4 ARC 机制下的循环引用	141
6.5 ARC 与 MRC 的相互兼容	145
6.6 阶段案例——微博(ARC)	147
6.7 本章小结	152
6.8 本章习题	153
第 7 章 分类和延展	156
7.1 分类概述	156
7.1.1 分类的简单使用	156
7.1.2 分类的注意细节	159

7.2	非正式协议	161
7.3	延展	163
7.4	本章小结	165
7.5	本章习题	165
第 8 章	block	169
8.1	block 概述	169
8.1.1	block 语法	169
8.1.2	block 的基本使用	170
8.1.3	使用 typedef 简化复杂的 block 定义	172
8.2	block 内部访问外部变量	173
8.3	block 作为函数的参数	174
8.4	block 作为函数的返回值	176
8.5	本章小结	177
8.6	本章习题	177
第 9 章	协议	182
9.1	协议概述	182
9.1.1	协议的基本使用	182
9.1.2	@required 和@optional	185
9.2	协议之间的继承	185
9.3	协议的类型限制	186
9.4	阶段案例——给男生找女友	187
9.5	本章小结	190
9.6	本章习题	190
第 10 章	Foundation 框架	194
10.1	NSString 的基本使用	194
10.1.1	NSString 的创建	194
10.1.2	NSString 最常见的操作	195
10.2	NSString 的其他操作	198
10.2.1	字符串的读写	198
10.2.2	字符串的比较	203
10.2.3	字符串的开始和结束判断	204
10.2.4	字符串的搜索	205
10.2.5	字符串的截取	207
10.2.6	字符串的替换	208
10.2.7	字符串的类型转换	209
10.3	NSMutableString	210

10.3.1	NSString 概述	210
10.3.2	NSString 的使用	210
10.4	NSArray	212
10.4.1	NSArray 数组的创建	212
10.4.2	NSArray 数组的基本使用	213
10.4.3	NSArray 数组的遍历	215
10.5	NSMutableArray 的使用	217
10.6	NSDictionary	219
10.6.1	NSDictionary 概述	219
10.6.2	NSDictionary 字典的创建	219
10.6.3	NSDictionary 的使用	220
10.6.4	NSDictionary 字典的遍历	221
10.7	NSMutableDictionary 的使用	222
10.8	NSFileManager	224
10.8.1	使用 NSFileManager 实现判断操作	224
10.8.2	使用 NSFileManager 获取信息	226
10.8.3	使用 NSFileManager 创建文件或文件夹	229
10.9	Foundation 框架的结构体	232
10.9.1	CGPoint	232
10.9.2	CSize	233
10.9.3	CRect	234
10.10	NSValue	235
10.11	日期处理	236
10.11.1	字符串与时间的相互转换	236
10.11.2	日期计算	237
10.11.3	得到日期的各个部分	239
10.12	阶段案例——通讯录	240
10.13	本章小结	248
10.14	本章习题	248

第 1 章

Objective-C 入门

学习目标

- 了解 Objective-C 的发展历史及其开发环境,能够体会到 Objective-C 这门语言的优点。
- 了解 Objective-C 项目代码的编译过程。
- 能够独立搭建开发环境并开发一个 Objective-C 程序,对 Objective-C 开发有进一步的认识和体验。
- 会使用帮助文档查看 API(Application Programming Interface,应用程序编程接口),具备解决问题的能力。

自 iPhone 发布以来,一直受到广大用户的大力追捧。iPhone 是一个激动人心的产品,基于 iPhone 的应用程序更是炙手可热。如果你考虑向全世界的人们展示一下你非凡的创意,或者想通过你的创意得到一些意想不到的收益,学习开发 iPhone 应用程序就是你要做的第一件事情。

Objective-C 语言是(简称 OC 语言)目前开发 iOS 应用程序的主要语言,在学习开发 iOS 应用之前,有必要先对 Objective-C 语言进行了解。接下来,本章将针对 Objective-C 语言的发展历史和开发环境进行详细讲解。

1.1 Objective-C 概述

2010 年之前,很多人不知道 Objective-C 语言,自从 iPhone4 发布后,Objective-C 语言才被大家认知,并且越来越多的程序员开始使用 Objective-C 语言。其实,Objective-C 这门语言历史悠久,早在 20 世纪 80 年代初期,Brad Cox 结合 C 语言和 Smalltalk 的优势,就设计出了 Objective-C 语言。

1985 年,乔布斯创建了 NeXT 公司,致力于设计经济实惠且功能强大的工作站,并选择 UNIX 作为其工作站的操作系统。后来,NeXT 公司设计出 NextStep,它是使用 Objective-C 语言编写的一套功能强大的界面工具包。

1995 年,NeXT 公司获得 Objective-C 的全部商业版权。

1996 年,苹果公司收购了 NeXT,NextStep 得到了 Mac 开发人员的广泛认可。之后,NextStep 更名为 Cocoa,使得 Objective-C 成为开发 Mac 平台的主力语言。

Objective-C 可以视为一门支持面向对象的 C 语言,它在 C 语言的基础上增加了一部分面向对象的语法,将 C 语言复杂和繁琐的语法封装得更为简单。它完全兼容 C 语言,也就是说可以在 OC 语言中写任意的 C 语言。

2014 年苹果公司在 WWDC(Worldwide Developers Conference,全球开发者大会)大会

上正式推出了 Swift 语言,可能有的读者会觉得 Swift 一定会取代 Objective-C,成为 iOS 开发的主力语言。但是就目前来看,要实现这一目标,还有一段道路要走,我们可以从下列三个原因来分析:

原因 1: 很多企业已经使用 Objective-C 语言开发好了自己的应用程序,这些企业是不可能使用 Swift 语言重新开发一遍的。

原因 2: Swift 刚发布不久,还有很多 bug 没有完善,企业不会那么积极地去小白鼠的。

原因 3: Swift 本身还没有成型,在后期发布的版本中,每次都有大量的变动,甚至包括语法的变动。

就目前状况而言,我们处于 iOS 开发新旧交替的阶段,旧的 Objective-C 语言要学,新的 Swift 语言也要学习。如果大家学会 Objective-C 后,再去学习 Swift,将会感觉非常轻松,因为大多数编程语言都是相通的,它们的编程思想都是一致的。

1.2 搭建开发环境

每一门编程语言都离不开硬件和软件的支撑,Objective-C 作为开发 iOS 应用的核心语言,它需要在 Mac OS X 系统下进行,其开发工具必须使用 Xcode。接下来,本节针对 Objective-C 开发环境的搭建进行详细讲解。

1.2.1 开发环境要求

1. 硬件要求

- (1) 苹果电脑(MacBook Pro、MacBook Air、iMac、Mac mini、Mac Pro)。
- (2) iPhone、iPodTouch 或 iPad 设备,主要用来测试编写好的程序。如果没有此条件,可以使用 Xcode 模拟器运行。

2. 软件要求

- (1) 苹果系统 Mac OS X(电脑系统必须升级到 OS X 10.9.4 及以上)。
- (2) Xcode 开发工具(Xcode6.0 及以上)。

1.2.2 系统环境搭建

苹果电脑自带的免费系统是 Mac OS X 系统,读者可以单击屏幕左上角的苹果图标,在弹出的列表中选择【关于本机】查看系统的版本,如图 1-1 所示。

(1) 单击【关于本机】选项,进入展示系统版本的窗口,如图 1-2 所示。

(2) 如果觉得系统版本过低,可以将现有系统升级到最新的 Mac OS X 系统。单击图 1-2 所示的【软件更新】,会看到 App Store 软件的更新界面,如图 1-3 所示。

(3) 单击图 1-3 中的【更新】按钮,显示系统更新的进度。更新完成后,可以通过 Finder 再次查看当前的系统版本。



图 1-1 查看系统版本



图 1-2 系统版本

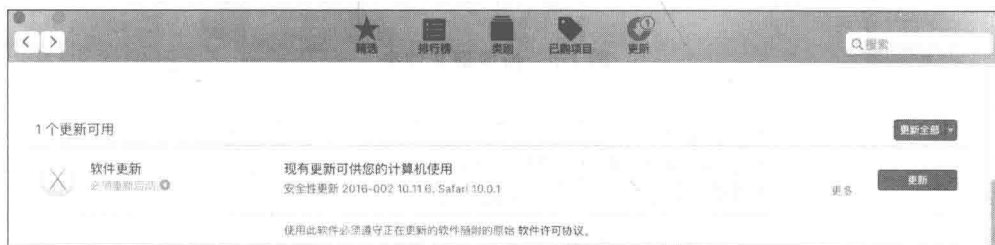


图 1-3 在 App Store 升级 Mac OS X

1.2.3 Xcode 开发工具搭建

Xcode 几乎是与 Mac OS X 同时发行的,不同版本的 Xcode 需要对应不同版本的系统才可以运行,下载时请注意系统环境。以 Xcode 8.1 为例,它需要的 Mac OS X 系统至少应该是 10.11.5 或更高版本,具体信息可以在 App Store 的 Xcode 下载页面右侧信息栏中查看,如图 1-4 所示。

Xcode 工具有多种下载方式,既可以从 App Store 下载,也可以从苹果开发平台 <https://developer.apple.com/download/download>,还可以单独从网上下载。通过 App Store 下载的 Xcode 将直接下载到应用程序的文件夹内,下载成功后,双击运行即可。如果自己从苹果开发者平台或者从网上下载,下载后的文件格式为 xip 或 dmg(如果是 xip 格式,双击解压后为 dmg 格式)。打开 dmg 文件,可以看到“Drag to install Xcode in your Applications folder”,如图 1-5 所示。

按照提示,拖动 Xcode 应用到 Applications 文件夹开始安装 Xcode,接着会自动进行拷贝,如图 1-6 所示。

拷贝完成后,打开应用程序文件夹,就可以看到安装好的 Xcode。双击打开 Xcode,可以看到 Welcome to Xcode 界面,如图 1-7 所示。

至此,iOS 开发环境就搭建好了,就可以开启 Objective-C 的学习之旅。



图 1-4 Xcode 8 对应的系统版本

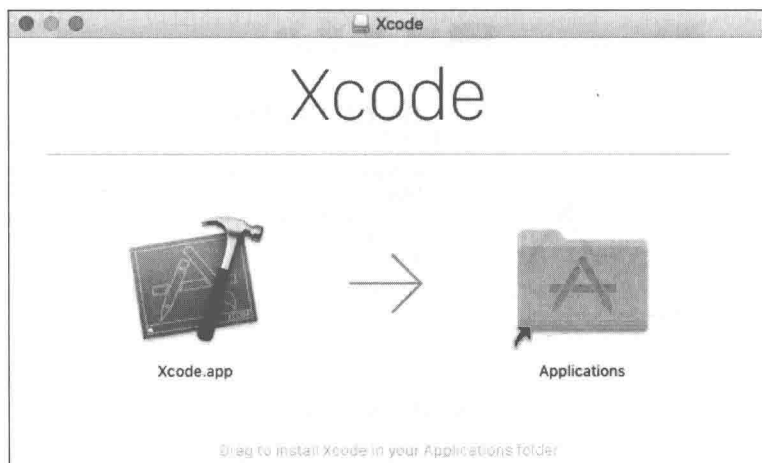


图 1-5 拖曳到文件夹

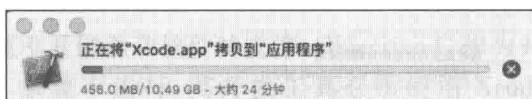


图 1-6 文件拷贝过程



图 1-7 Welcome to Xcode 界面

1.3 开发第一个 Objective-C 程序

1.3.1 使用 Xcode 开发 Objective-C 程序

磨刀不误砍柴工,熟悉了 Objective-C 开发环境后,使用 Xcode 编写第一个 Objective-C 程序,具体步骤如下:

(1) 打开 Xcode,在图 1-7 所示的 Welcome to Xcode 界面选择左侧的 Create a new Xcode project 新建一个工程。新建工程的界面如图 1-8 所示。

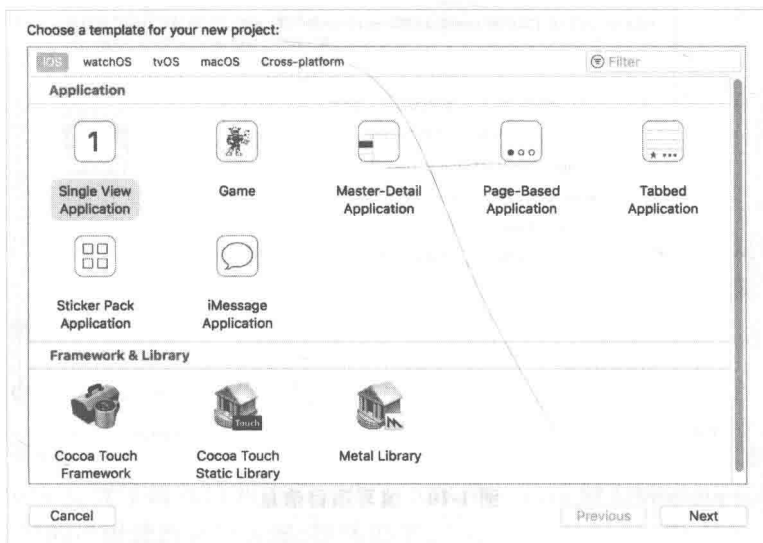


图 1-8 选择 Create a new Xcode project 弹出的窗口

(2) 在图 1-8 所示的窗口中,可以选择项目的类型和模板。由于本书不涉及应用程序界面,这里选择 macOS 下的 Command Line Tool,如图 1-9 所示。

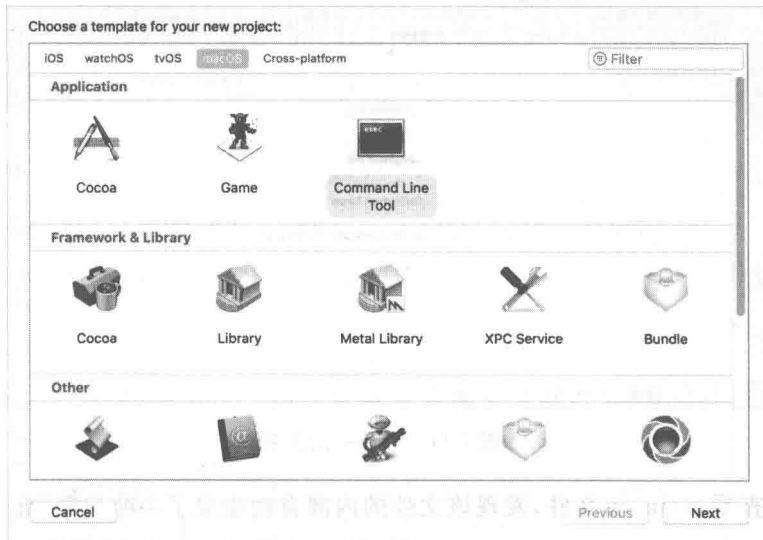


图 1-9 选项 macOS 下的 Command Line Tool

(3) 单击图 1-9 中的 Next 按钮,进入填写项目信息的窗口,填写完毕的界面如图 1-10 所示。

(4) 单击图 1-10 所示的 Next 按钮,进入选择保存项目文件的窗口,设置好项目的保存路径后,单击 Create 按钮完成工程的创建,如图 1-11 所示。