国家信息技术紧缺人才培养工程指定教材

工业和信息化人才培养规划教材

Swift项目开发基础教程

传智播客 编著

本书涵盖了Swift开发的前沿技术，并提供了综合项目——《2048》游戏开发。

提供免费教学资源，包括精美教学PPT、教学视频、教学设计、教学题库等。

添加ＱＱ或微信号208695827，获取教学答案、源码，抢“助学金红包”。

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

工业和信息化人才培养规划教材

Swift项目开发基础教程

传智播客 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

Swift项目开发基础教程 / 传智播客编著. -- 北京 : 人民邮电出版社, 2016.8
工业和信息化人才培养规划教材
ISBN 978-7-115-41960-6

Ⅰ. ①S… Ⅱ. ①传… Ⅲ. ①程序语言－程序设计－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2016)第101125号

内 容 提 要

本书作为一本基于 Swift 3.0 语法的全新教程，系统全面地讲解了使用 Swift 开发项目的知识和技术，可以帮助初学者真正达到从零基础到独立开发项目的技术水平，成为 Swift 开发者。

本书共分为 12 章：第 1～5 章讲解了 Swift 开发的一些基本语法；第 6～7 章讲解了 Swift 面向对象的编程思想；第 8～10 章讲解了 Swift 的开发特性；第 11 章讲解了 Swift 与 Objective-C 项目的相互迁移；第 12 章教大家开发《2048》游戏。本书从始至终保持通俗易懂的描述方式，采用理论与案例相结合的方法帮助初学者更好地理解各个知识点在实际开发中的应用。

本书附有源代码、习题、教学视频等配套资源，而且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。

本书既可作为高等院校本、专科计算机相关专业的程序设计课程教材，也可作为 iOS 开发技术的培训教材。

◆ 编　　著　传智播客
责任编辑　范博涛
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　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：787×1092　1/16
印张：23.5　　2016 年 8 月第 1 版
字数：589 千字　　2016 年 8 月河北第 1 次印刷

定价：49.80 元

读者服务热线：(010)81055256　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广字第 8052 号

序 言

为什么出这本书

现如今，随着互联网时代的发展步伐和技术的更新速度不断加快，企业对互联网人才的要求越来越高。然而，目前高校的 IT 教育教材更新速度相对缓慢，课程相对滞后，导致学生所学不能满足企业应用所需。

如何解决现状

传智播客作为一家专业的 IT 培训机构，一直将“改变中国的 IT 教育”作为自己的事业，并为此拼搏了 10 年。在这 10 年期间，传智播客默默耕耘。10 年的沉淀让传智播客拥有了系统完善的 IT 培训课程体系，毫不夸张地说，传智播客已经为 IT 学子开辟了一条全新的求知之路。

通过对中国 IT 教育和大学生现状的深刻分析，传智播客不断与先进的企业进行对接，对教材及教学方案不断更新，有针对性地出版了计算机书籍 30 多种、教学视频数十套、发表各类技术文章数百篇，直接培养的软件工程师就有 10 万多名，被传智播客公开的免费学习视频（网址 http://yx.boxuegu.com）影响的学生更是多达数百万人。

传智播客投入巨额资金，用于为高校师生提供以下服务。

针对高校教师的服务

（1）为方便老师教学，缩短教师的备课时间，减轻教师的教学压力，有效提高教学质量，传智播客基于 10 年的教育培训经验，精心设计了：

“教案+授课资源+考试系统+题库+教辅案例”IT 系列教学应用课程包

微信扫一扫

申领资源

（2）本书配备由传智播客一线讲师录制的教学视频（网址 http://yx.boxuegu.com），按本书知识结构体系匹配，教师可用于备课参考，也可作为教学资源使用。

针对学习者的服务

本书配套源代码的获取方法：添加播妞 QQ 号 208695827 或微信号 208695827

注意！播妞会随时发放“助学金红包”。

前 言

PREFACE

Swift 是苹果公司于 2014 年 WWDC（苹果开发者大会）发布的最新开发语言，它可与 Objective-C 共同运行在 Mac OS 和 iOS 平台上，用于搭建基于苹果平台的应用程序。之后，苹果公司不断发布其新的版本。从 Swift 1.0 开始，市场上陆陆续续地出版了与之相关的图书，其中多数都是基于 Swift 1.0、Swift 1.2 版本，而这些版本很快就被替代。2015 年 12 月 4 日，苹果公司宣布开放 Swift 编程语言的源代码，并于 2016 年 3 月发布了相对稳定的 Swift 2.2 版本，2016 年 6 月发布了 Swift 3.0。为了帮助更多爱好 Swift 编程的开发人员实际体会到 Swift 简洁的语法和强大的功能，我们在印刷前，再次对本书进行了升级，全面讲解了 Swift 3.0 的新特性。

本书以全新的 OS X 10.11.4 为平台，以 Xcode8 为开发工具，全面介绍了支持 Swift 3.0 的语法，以及使用 Swift 开发 iOS 应用的基本知识。在内容编排上，本书以苹果官方 Swift 开源文档的内容为主线，结合与 C、Objective-C 的对比，采用案例驱动的方式带领读者学习 Swift。本书的所有的语法都提供了大量示例程序，很多地方甚至从正、反两面举例，最后还带领读者开发《2048》游戏。

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、深刻的学会 Swift 开发技能。

自主学习者建议您勤思考、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测试题进行自我检测。

传智播客之所以选择推出这本教材，不仅希望可以填补 Swift 3.0 市场的空白，更是希望广大读者可以从书中有所获益，开发出更多优秀的应用程序。

本教材共分为 12 个章节，具体内容如下。

- 第 1 章主要介绍了 Swift 开发的一些概念知识，包括 Swift 语言的发展及特性，Swift 与 OC 的异同点、Swift 开发环境的配置以及 Xcode 的安装，并在最后带领大家编写了第一个 Swift 程序。通过本章的学习，读者可以对 Swift 开发的背景知识有所了解，并且能够掌握 Swift 项目的结构。

- 第 2 章讲解了 Swift 语言的基本语法，是学习 Swift 语言的基础，包括 Swift 的关键字和标识符、常量和变量的引用、数据类型和运算符等。对本章的内容，读者一定要加强理解，尤其是元祖类型、可选类型和各种运算符。另外，还要勤加练习，熟练使用，为以后的学习奠定好基础。

- 第 3 章主要介绍了控制流语句，包括条件语句和循环语句。本章的内容十分的重要，掌握了本章的内容才能编写更加复杂的程序并且有助于后面章节的学习。

- 第 4 章首先介绍了字符，接着介绍了字符串的初始化和使用，最后介绍了数组、Set 和字典的内容，包括数组、Set 和字典的创建与使用。通过本章的学习，一定要掌握字符串和集合的操作技巧，能够灵活地运用它们。

- 第 5 章首先介绍了与函数相关的内容，包括函数的定义和使用、参数和返回值，接着由嵌套函数引出了闭包，介绍闭包的概念、定义方式，以及尾随闭包的相关内容，最后介绍了枚举的内容，包括枚举的定义和访问、使用 switch 语句匹配枚举值等。通过本章的学习，

读者要掌握函数、闭包、枚举的基本使用，能够灵活地使用这些技术。

- 第 6 章首先介绍了面向对象的基本知识，包括类、结构体、属性、方法、构造函数、析构函数、下标脚本。本章内容是学好面向对象编程的垫脚石，希望大家能够熟练掌握。
- 第 7 章首先介绍了面向对象的三大基本特性，主要是继承的特性，以及在 Swift 语言中如何实现继承和重写，然后介绍了可选链的用法，使用 is 操作符和 as 操作符实现类型检查和类型转换，以及嵌套类型的使用等。本章的大部分内容都是 Swift 特有的语法，希望读者能够认真学习和掌握。
- 第 8 章主要讲解了扩展、协议和代理三个概念。通过本章的学习，大家应该学会使用扩展和协议，并学会使用代理实现协议，真正理解扩展和协议在实际开发中的作用。
- 第 9 章主要介绍了 Swift 中的内存管理机制 ARC（自动引用计数）、类实例的循环强引用形成的原因及解决方法、闭包引起的循环强引用和解决方法等。在实际开发中，需要理解 ARC 的工作机制，并要注意检查代码中是否可能出现循环强引用，如果出现则必须予以解决，防止内存泄露。
- 第 10 章主要讲解了 Swift 语言中的一些高级特性，包括泛型、错误处理机制、访问控制特性、命名空间、高级运算符等。这些高级特性在实际开发中非常重要，希望读者能够多加揣摩练习，熟练掌握。
- 第 11 章主要介绍了 Objective-C 与 Swift 之间的互操作，通过本章的学习，读者要掌握 Objective-C 与 Swift 之间互操作的技巧，以便更好地运用到工作中，提高开发效率。
- 第 12 章用 Swift 开发《2048》游戏，按照项目的实现流程完成开发。希望读者通过本章的学习，能够深入地理解 Swift 的各个知识点，灵活运用到项目中。

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，理清思路，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、刘传梅、王晓娟、李凯、郭敬楠、尹桥印、潘星、齐瑞华等，全体人员在这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们来信来函给予宝贵意见，我们将不胜感激。您在阅读本书时，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。

传智播客

2016-6-29 于北京

目 录

PART 1

第1章 Swift开发入门

学习目标

- 了解什么是 Swift 及其语言特点
- 掌握 Xcode 的安装和基本使用
- 学会编写第一个 Swift 程序

随着 Mac OS X 和 iOS 系统的不断发展，越来越多的移动开发者开始学习 Swift 语言。Swift 语言是由苹果公司于 2014 年推出，用来撰写 Mac OS X、iOS 和 Watch OS 操作系统上的应用程序，如 Mac、iPhone、iPod touch、iPad、Apple Watch 这些设备上的应用都可以用 Swift 语言开发。Swift 的出现旨在替代 Objective-C（简称 OC）语言，因此比 OC 语言更加先进、严谨和易用。一经推出，Swift 已经受到广泛好评。本章将针对 Swift 语言的一些简单知识进行详细讲解，并带领大家开发第一个 Swift 程序。

1.1 Swift 语言概述

1.1.1 什么是 Swift 语言

Swift 是苹果公司推出的一种新的编程语言，用于编写 iOS、Mac OS X 和 Watch OS 应用程序，它结合了 C 语言和 Objective-C 的优点并且不受 C 兼容性的限制，同时支持过程式编程和面向对象的编程，是一种多范式的编程语言。Swift 语言是以迅雷不及掩耳之势出现并发展起来的，为了更好地了解 Swift，下面针对 Swift 的发展历程及变化进行详细讲解。

2010 年 7 月 LLVM 编译器的原作者、苹果开发者工具部门总监克里斯·拉特纳（Chris Lattner）开始着手 Swift 编程语言的工作，用一年的时间完成基本架构，同时参与的还有一个叫做 dogfooding 的团队。截至 2014 年 6 月，Swift 大约历经 4 年的开发期。

2014 年 6 月苹果在发布 Xcode 6.0 的同时发布了 Swift 1.0，从此 Swift 语言走进了程序员的生活。

2015 年 2 月，苹果同时推出 Xcode 6.2 Beta 5 和 6.3 Beta，在完善 Swift 1.1 的同时，推出了 Swift 1.2 测试版。

2015 年 6 月，苹果发布了 Xcode 7.0 和 Swift 2.0 测试版，并且宣称 Swift 会在 2015 年底开源。

2015 年 11 月 9 日，苹果发布了 Xcode7.1.1 和 Swift2.1，Swift 的语法文档也有更新。

2015 年 12 月 4 日，苹果公司宣布其 Swift 编程语言开放源代码。

2016 年 3 月 22 日，苹果公司发布 Swift2.2 版本。

2016 年 6 月 13 日，苹果公司发布了 Swift 3.0 版本。

从发布至今，Swift 一直在以非常惊人的速度不断发展，苹果的每一个举措都彰显其大力推广 Swift 的决心。目前国内很多公司的新项目已经直接采用 Swift 开发，并且公司内部也在做 Swift 的人才储备，可见 Swift 语言取代 Objective-C 势在必行。时势造英雄，掌握 Swift 语言的开发人员将是近年来 IT 职场中受人青睐的稀缺人才。

1.1.2 语言特点

Swift 语言能够如此迅速的发展，离不开它独特的语言特点。需要强调的是，Swift 绝对不是解释性语言，更不是脚本语言，它和 Objective-C、C++一样，编译器最终会把它翻译成 C 语言。与其他语言相比，Swift 语言的特点可以归纳为五点，具体如下。

1. 快速

与其他流行的面向对象语言相比，Swift 主要的优势在于其语法简单，比 Objective-C 的语法还要简单，初学者只需要非常短的时间就可以掌握面向对象编程的核心方法，快速上手。同时，Swift 和 Objective-C 是并存的，如果你是 Objective-C 开发者，完全可以在原来项目的基础上直接用 Swift 进行开发。Swift 的编译器使用高级的代码分析功能来调优代码，让你更专注于开发应用，而不必在性能优化上投入大量的精力。

2. 安全

Swift 是类型安全的，它使用类型推断机制，限制对象指针使用、自动管理内存来使程序更安全，而且，变量总是使用前初始化的。另一个安全功能是默认情况下 Swift 对象永远不会是零。事实上，编译器会阻止在编译时试图出现错误或使用一个空对象，这些功能让开发人员更容易开发出安全稳定的软件。

3. 现代

Swift 具有错误处理、guard 语句和协议扩展等语法新特性，还有 Optional、泛型、元组等现代语言的特性。Swift 有多个返回值闭包统一的函数指针，快速、简洁的迭代或集合比 Objective-C 语言更具灵感，更接近自然语言，使代码可读性更好。

4. 互动

Swift 对于初学者来说也很友好，它是第一个既满足工业标准又像脚本语言一样充满表现力和趣味的编程语言，可以使用 playground 来试验新技术，分析问题，做所见即所得的界面原型。它支持代码预览，这个革命性的特性可以允许程序员在不编译和运行应用程序的前提下运行 Swift 代码并实时查看结果。

5. 开源

苹果公司对 Swift 的编译器、标准库和源码的开源，大大提高了程序员对 Swift 语言的热情。Swift 的开源特性使其更加通用、更加多样化——除了苹果平台的应用，开发者也可以在其他项目中使用这个编程语言。另外，Swift 开源也展示出苹果公司是非常有远见的。费德里希曾经表示："我们认为未来 20 年 Swift 将成为编程的标准语言，我们认为它将成为未来主要的编程语言之一。"

Swift 作为一种新的编程语言，从来没有停止过发展自己的脚步，它的功能一直在不断地进行完善和更新，目的就是使编程更加简单、快速、安全，在学习过程中需要用心体会。

1.1.3 开发框架

在学习 Swift 之前，初学者还需要对开发框架的概念有所了解。框架的功能类似于动态库，即可以在运行时动态的载入应用程序的地址空间，但框架作为一个捆绑（计算机）而非独立文件，其中除了可执行代码外，也包含了资源、库文件、头文件、文档和各种驱动程序等。每种编程语言都有它们自己的框架，而运行在 Mac OS X 平台上的 Cocoa 及运行在 iOS 平台上的 Cocoa Touch，则是苹果公司为 Swift 开发人员提供的一系列强大的开发框架。下面将针对 Cocoa 和 Cocoa Touch 框架进行详细的讲解。

1. Cocoa 框架

Cocoa 是苹果公司为 Mac OS X 所创建的原生面向对象的 API，是 Mac OS X 上五大 API 之一（其他四个是 Carbon、POSIX、X11 和 Java），它包含了 Foundation 和 Application Kit 两大主要框架。其中，Foundation 框架是基于 Core Foundation 的，一般来说和界面无关的类基本都属于 Foundation 框架，如 NSString、NSArray、NSError 和 NSNotification；而 Application Kit 框架则包含了程序与图形用户界面交互所需的代码，它是基于 Foundation 建立的，并且很多代码都使用“NS”前缀，但这些代码只能在 Mac OS X 中使用。

2. Cocoa Touch 框架

Cocoa Touch 是 iOS 的开发框架，它重用了许多 Mac 系统的成熟模式，更加专注于基于触摸的开发接口和性能优化，主要包含 Foundation 和 UIKit 两个框架。UIKit 为开发者提供了在 iOS 上实现图形、事件驱动程序的基本工具，其建立在 Application Kit 框架的基础上，包括文件处理、网络、字符串操作。

此外，Cocoa Touch 还包含了创建世界一流 iOS 应用程序需要的所有框架，如核心动画使用的是 Core Animation 框架，音频音效使用的是 Core Audio 框架，数据存储使用的是 Core Data 框架。

1.1.4 Swift 与 Objective－C 语言对比

在 Swift 之前，所有的 iOS 应用都是使用 Objective-C 语言进行开发的，Swift 的出现必然会引起程序员对这两种语言进行对比，从发布至今，各路大牛早已纷纷对 Swift 进行过各种挖掘了，发现 Objective-C 和 Swift 存在很多相同点和不同点。下面，针对这两种语言的相同点和不同点分别进行介绍。

1. 相同点

Swift 和 Objective-C 是相互兼容的，两者都是基于 Cocoa 和 Cocoa Touch 框架。也就是说，你可以在一个项目中使用两种语言进行开发，两者都可以用来开发 iOS 应用。Swift 跟 OC 共用同一套运行环境。

2. 不同点

关于 Swift 和 Objective-C 的不同点将从文件结构和语法内容两个方面进行讲解。

（1）文件结构

Objective-C 继承了 C++语言的文件格式，把头文件和实现文件分开写，使用.h 作为头文件的后缀，使用.m 文件作为实现文件的后缀，Swift 则使用.swift 作为后缀名，这也体现了 Swift 的简洁性。接下来，通过一张图来对比一下两者的区别，具体如图 1-1 所示。

（2）语法内容

除了文件结构不同，Swift 和 Objective-C 两者在语法上也是有区别的，具体如下。

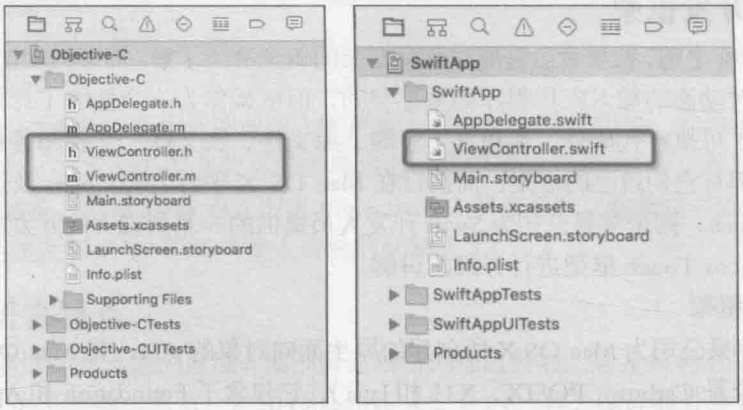

图 1-1　Swift 和 Objective-C 文件结构对比

① Objective-C 中，类的声明和实现分别使用关键字@interface 和@implementation 表示，而在 Swift 中，类是使用关键字 class 表示的。接下来，以两种语言默认创建的 ViewController 类为例来展示两者定义类方式的不同，如图 1-2 所示。

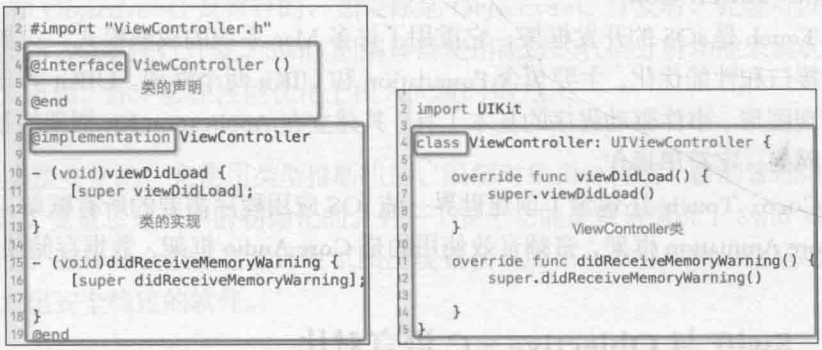

图 1-2　ViewController 类的对比

在图 1-2 中，左边是使用 Objective-C 语言默认定义的 ViewController.m 类，可以看出它使用@interface 声明类，使用@implementation 实现类。右边是 Swift 语言默认定义的 ViewController 类，只需要使用 class 声明。通过对比发现，Swift 定义的类更加简洁。

② 在 Swift 中函数和方法的定义是用 func 声明的，这里摘取图 1-2 中的 viewDidLoad 方法的代码，此方法是没有返回值的，如下所示。

```
override func viewDidLoad() {
    super.viewDidLoad()
}
```

在 Objective-C 中 viewDidLoad 方法的声明格式如下所示：

```
- (void)viewDidLoad {
    [super viewDidLoad];
}
```

③ 在 Swift 中取消了 Objective-C 的指针及其他不安全访问的使用，并且还舍弃了 Objective-C 早期应用 Smalltalk 的语法，全面改为句点表示法。

④ 在 Swift 中使用关键字“let”定义常量，使用关键字“var”定义变量。如下两行代码定义了一个 NSString 类型的变量 name 和一个值为 20 的常量 age。这在 Objective-C 中是没有的。

```
var name:NSString
let age = 20
```

⑤ Swift 提供了类似 Java 的命名空间（namespace）、泛型（generic）、运算对象重载（operator overloading），Swift 被简单地形容为“没有 C 的 Objective-C”。

⑥ 还有一些性质是 Swift 独有的。例如，Swift 独有的范围运算符“a...b”是闭区间包含，如 5...7 就是取值范围 5,6,7；“a..<b”是半开半闭区间包含，如 5..<7 就是取值范围 5,6。

⑦ Swift 独有的元组类型“var point = (x:15,y:20.2)”就表示元组名是 point ，里面有两个元素 x 和 y。

关于 Swift 和 Objective-C 的不同之处肯定不止这些，但是通过本节的学习，相信同学们也可以做到举一反三，在此就不一一列举了。

总的来说，Swift 吸收了很多其他语言的优秀语法，写起来比 Objective-C 简洁得多。而且，Swift 增加了各种功能用以提高其安全性、易用性和表现力。现在苹果的 Cocoa 已经开始使用 Swift 语言进行重写，所以 Swift 语言取代 Objective-C 语言是必然的趋势。

1.2 Swift 开发环境和工具

1.2.1 开发环境

每一种开发语言都有它的开发环境，Swift 也不例外。在正式开发应用程序前，需要搭建 Swift 开发环境，以便更好地使用各种开发工具和语言进行快速应用开发。和 Objective-C 一样，Swift 的开发环境需要在 Mac OS X 系统中运行，因此 Mac OS X 系统的支持是必须的。另外，无论是 Mac OS X 还是 iOS，苹果都建议你使用最新版的 Xcode 进行开发，因为 Swift 对苹果系统的要求是较高的，Swift 3.0 的开发环境要求如下。

（1）必须拥有一台苹果电脑。因为集成开发环境 Xcode 只能运行在 OS X 系统上。

（2）苹果系统 Mac OS X 10.11.4 及以上。

（3）Xcode 开发工具 8 版本及以上。

如果当前系统不是 OS X 10.11.4，就需要对它进行升级。升级的方法比较简单，只需要进入 App Store，在主窗口单击“更新”按钮，会看到一些软件的更新提示，然后单击更新按钮，完成系统更新。App Store 的图标和更新系统的主界面，分别如图 1-3 和图 1-4 所示。

图 1-3 App Store 图标

图 1-4　更新 Mac OS X

多学一招：查看系统版本

单击桌面左上角的苹果按钮，选择关于本机，就会出现当前系统的版本信息，如图 1-5 所示。

图 1-5　本机信息

1.2.2　Xcode 概述

开发 Swift 程序，可以选择用户电脑里应用程序中的终端，通过命令行操作，创建文本文档、编写程序，之后继续通过命令行完成程序编译。但是这样操作非常麻烦，为了方便实际开发，苹果公司向开发人员提供了免费的开发工具——Xcode，它可以用于编辑、编译、运行及调试代码。

俗话说，工欲善其事，必先利其器。要想在 iOS 系统开发应用程序，首先需要在 Mac OS X 计算机上配备一个 Xcode 工具。Xcode 是苹果公司提供的一个集成开发环境，它用于管理工程、编辑代码、构建可执行文件、进行代码调试等。为了更好地认识 Xcode，接下来，从 Xcode 的适用性、辅助设计、开发文档支持三方面进行详细讲解，具体如下。

1. 适用性方面

Xcode 中所包含的编译器除了支持 Swift 以外，还支持 Objective-C、C、C++、Fortran、Objective-C++、Java、AppleScript、Python 及 Ruby 等，同时还提供 Cocoa、Carbon 及 Java 等编程模式。另外，某些第三方厂商也提供了 GNU Pascal、Free Pascal、Ada、C Sharp、Perl、Haskell 和 D 语言等编程语言的支持。

2. 辅助设计方面

使用 Xcode 工具开发应用程序时，只需要选择应用程序对应的类型或者要编写代码的部分，然后 Xcode 工具中的模型和设计系统会自动创建分类图表，帮助开发人员轻松定位并访问相应的代码片段。另外，Xcode 工具还可以为开发人员的应用程序自动创建数据结构，开发人员无需编写任何代码，就可以自动撤消、保存应用程序。

3. 开发文档支持方面

Xcode 提供了高级文档阅读工具，它用于阅读、搜索文档，这些文档可以是来自苹果公司网站的在线文件，也可以是存放在开发人员电脑上的文件。

1.2.3 安装 Xcode 8 开发工具

配置好 Swift 开发环境后，若想进行 Swift 开发，还需要在苹果电脑上安装开发工具。开发 iOS 程序使用的开发工具都是 Xcode，默认情况下，Mac OS X 系统没有安装 Xcode 软件，可以从网上下载 dmg 安装包进行安装，也可以从 App Store 上直接下载。这里，以安装 Xcode 8 为例，针对这两种安装方式进行讲解，具体如下。

1. 使用 dmg 安装包

在 Mac 上安装软件很简单的，双击 dmg 文件可以看到“Drag to install Xcode in your Applications folder”，这时，可以直接拖动 Xcode 到右边应用程序文件夹里，实现 Xcode 安装和自动拷贝，具体如图 1-6 和图 1-7 所示。

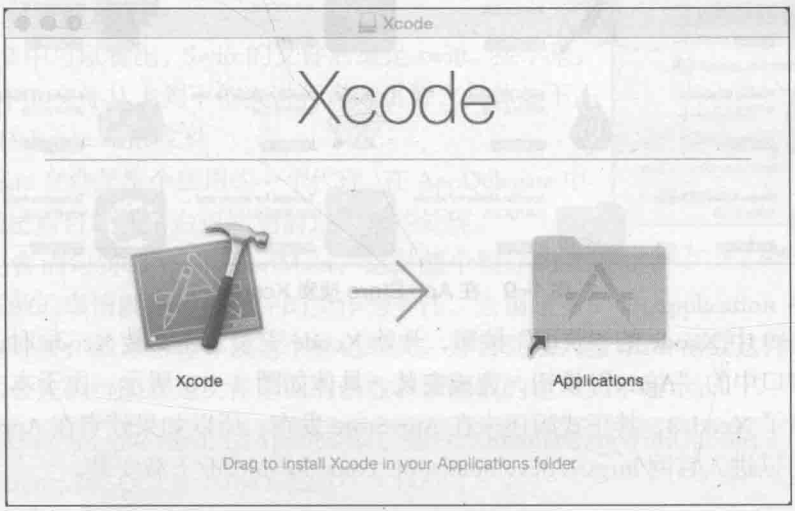

图 1-6 安装包安装 Xcode

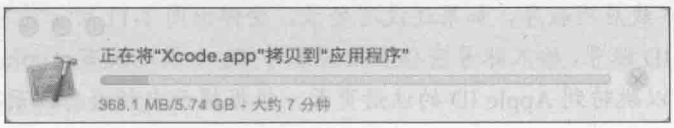

图 1-7 拷贝文件