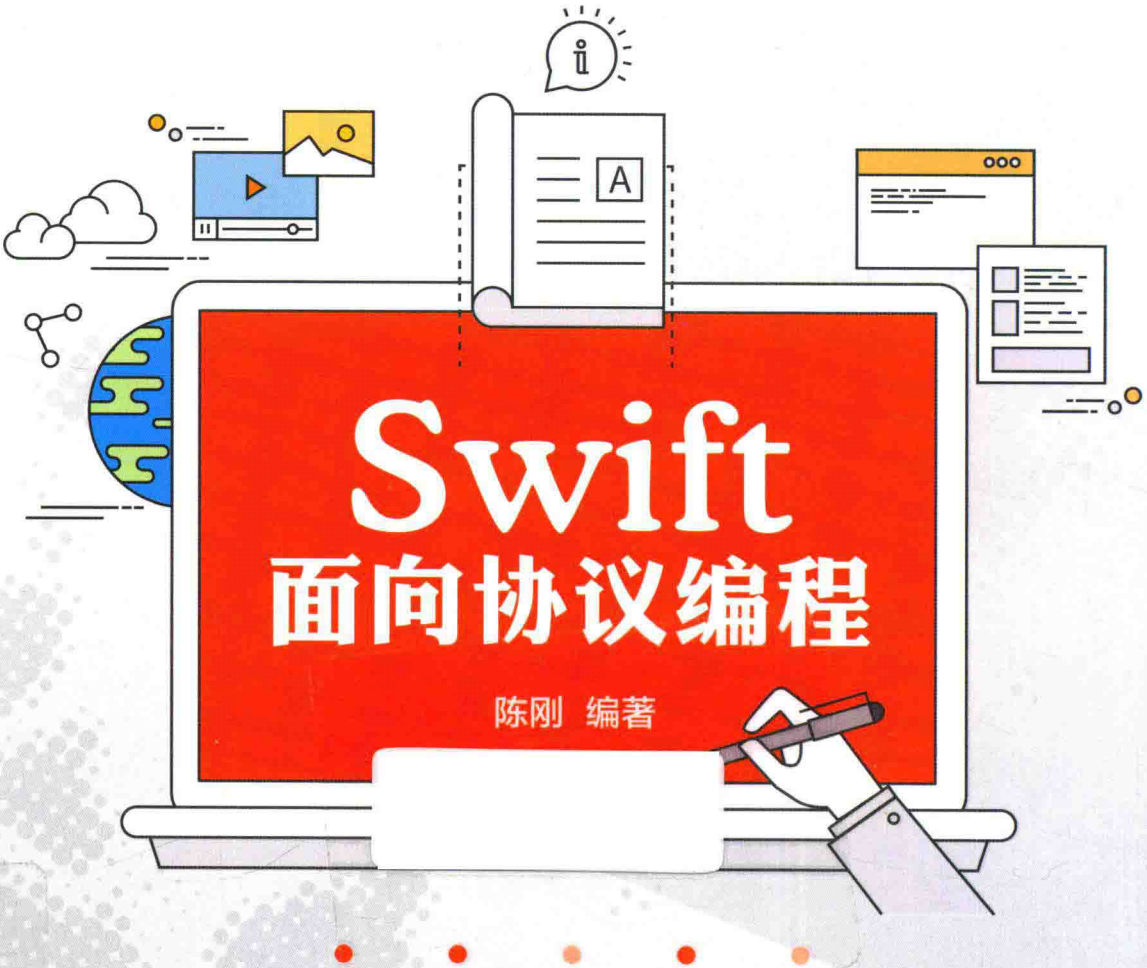
Broadview
www.broadview.com.cn



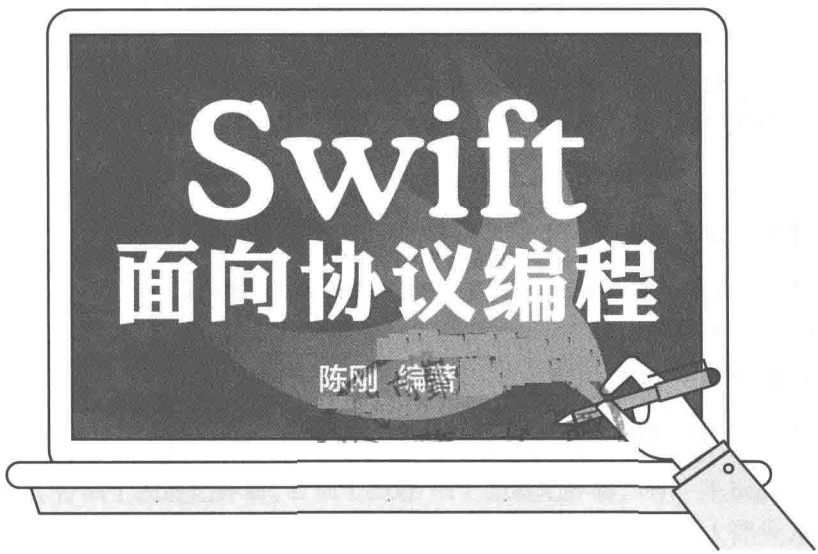
Swift 面向协议编程

陈刚 编著

Swift是一门面向协议编程的语言

2.2与3.0双版本教学，深度解读Swift进化的妙处
语法与实战齐备，梯度化知识点编排引爆你的思维

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书第 1 章简要介绍了 Swift 的发展历史以及 Swift 开发工具的获取途径。第 2 章介绍了 Swift 的基础语法，细致讲解了面向协议编程中至关重要的协议扩展用法与泛型优化。第 3 章讲解了 Swift 的进阶语法，从语言层面深入到内存层面，对 Swift 的特性进行了剖析，帮助读者写出更加 Swift 化的代码。第 4 章通过一个完整的计算器 Demo 指导读者快速完成一个 iOS 应用的开发，熟悉 Xcode 的使用，理解经典的 MVC 编程模式。第 5 章通过一个备忘录 Demo，将传统的 MVC、 MVVM 思想构建的程序与基于协议构建的程序进行了多个维度的对比，揭开了面向协议编程思想的神秘面纱，教会读者如何在实际工作中应用面向协议编程思想。

图书在版编目（CIP）数据

Swift：面向协议编程 / 陈刚编著. —北京：电子工业出版社，2017.1
ISBN 978-7-121-30195-7

Ⅰ. ①S… Ⅱ. ①陈… Ⅲ. ①程序语言－程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 258867 号

责任编辑：安　娜
印　　刷：北京中新伟业印刷有限公司
装　　订：北京中新伟业印刷有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980　1/16　　印张：20.25　　字数：385 千字
版　　次：2017 年 1 月第 1 版
印　　次：2017 年 1 月第 1 次印刷
定　　价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819　faq@phei.com.cn。

推荐序

很高兴因为认识陈刚这位优秀的朋友，有幸让自己的文字能够呈现在读者面前，每一个字都显得很珍贵。

这是一本介绍 Swift 语言的书。说起编程语言，这些年也广泛地涉猎过诸多语言。从初中时开始接触 VB，写一些小程序，然后发送给朋友们。那时的 VB 看起来就像数学公式一样。现在回头一看，可视化的界面编辑也正像 Xcode 的 Interface Builder 一样。到后来接触了算法竞赛，开始写一些 Pascal，告别了图形化界面，在 Turbo Pascal 的蓝屏背景下写一些 begin、end、readln、writeln。后来去外面参加竞赛，考卷有三种语言选择：Pascal、C、C++。当大部分人用 Pascal 的时候，有少数几个人用 C、C++考试，而监考老师发卷到他们那里的时候也会说好厉害。于是潜意识中，觉得 C、C++ 是比 Pascal 更加厉害的语言。于是学了一年 Pascal 的我再度归零，拿起了谭浩强老师编写的《C 程序设计》。C 语言让我感觉很简洁，同时接触了指针、内存管理、字符串库。比起 Pascal，C 代码让我觉得是那样简洁、优雅。每一个字符都有它的力量，无法去掉、无法再简洁了。用它来写深度搜索、图论、二叉树算法，然后去在线评测系统提交代码，再优化优化，接着代码效率能上排名榜的第一页，是一个 15 岁少年所能期待的最美好的事情。

接着读起了刘汝佳前辈的《算法竞赛入门经典》一书，感慨语言之上，还能实现那么多算法，解决一个又一个的问题。就像这本书一样，前面的语言章节仅是铺垫，后面的 iOS 开发章节才是高潮。精通一门语言，然后用它去开发各种灿烂的 iOS 应用，才是高潮。

算法陪我度过了高中。高考之后，我读了《黑客与画家》一书，对 Lisp 语言崇拜不已，于是就学了 Lisp。在 2013 年到 2015 年，又陆续学了 Java、Clojure、JavaScript、Objective-C 和 Swift。

后来创业做 Reviewcode.cn，又学了 PHP、 CSS 和 HTML。开始更多地知道后端系统里，语言起着怎样的作用。后来到了新公司，担任 Go 后端工程师。又在一个星期内学了 Go，并用它做了抢/发红包、充值的 API 接口，协调移动端同时上线了。

当用过很多语言后，你会发现很容易上手一门新语言，并快速用来干活。最近我也重新起航创业了，创立了趣直播——一个知识直播平台，孙源 Sunny、iOS 程序猿都来直播过，可到我们的公众号“平方根平台”来看看，也可加我的微信“lzwjava”来交流产品或技术。

初识 Swift 的原因是公司准备把 Objective-C 示例应用用 Swift 重写。重写是件挺无聊的事情，于是只重写了觉得有趣的部分，其他代码都用 GitHub 上的 Objective-C 转 Swift 的程序先转一遍，然后再去修复。将 Objective-C 自动转成 Swift，在 Swift 1.0 版本时还很好用，不用修复。可惜我当时用的时候，已经是 Swift 1.2 版本了。算是给本书提供了一个 Swift 语法多变的真实例子吧。

后来用 Swift 做了微图项目，旨在模仿国外的 imgur，展示互联网上最热门的图，接触到了 extension、protocal、enum。再后来尝试用 Swift 写了一个 HTML Parser，比如从一大堆 HTML 中解析 class="btn-blue" 的 div 出来， 从而对 Swift 字符串处理、错误处理、AnyObject 有了更深入的了解。

Swift 中的 map、flatMap、filter，这些是 Lisp 系语言函数的基础，通常用它们来构造更高阶的函数。比如下面的 clojure 代码片段，从 map 中获取 values 对应的 keys：

```
(let [m {:x 1 :y 2 :z 3}
      vset #{2 3}]
  (map first (filter (comp vset last) m)))
;;=&amp;gt; (:y :z)
```

很高兴 Swift 有这么多函数式的语法支持，函数式语言很擅长处理数据转换，很优雅也很强大，想想上面的几行代码用其他语言实现需要多写多少代码吧？

我是早前于 SwiftGG 的一次聚餐中认识作者陈刚兄的，后续时有交流。虽然不能经常见面，但互相欣赏、惺惺相惜。希望未来有机会和陈刚兄一起共事。

本书通读下来，不禁对陈刚兄做事的认真感到佩服。相信他能带领大家进入 Swift 的世界，掌控这门语言，写出各种灿烂的应用！

——趣直播创始人 李智维

前言

众所周知，在2015年的 WWDC 上，苹果发布了 Swift 语言的 2.0 版本，并且正式宣布 Swift 成为一门面向协议编程的语言。作为一个从 2014 年就开始接触 Swift 的程序员，我在当时已经具备了一定的 Swift 面向对象编程经验，这些经验帮助我顺利地完成了工程的更新；同样，这些经验也让我变得迟钝。在 2.0 版本发布之初，面向协议编程这个名词就好像一滴水滴入了大海中，从我的关注中溜走了。好在我是个对知识充满敬意的人，此后的数月中，我都在重点研究 WWDC 2015 的视频资料，此时我才惊奇地发现，最初的那滴水变成了我心头的惊涛骇浪，我开始不断地思考与探索面向协议编程的含义。

那么面向协议编程到底是什么呢？一句话就能概括：用协议扩展的方式代替继承，实现代码复用。这个“代替”的影响是深远的，协议扩展使得值类型在保持原有特性的同时，也能和引用类型一样实现代码的复用。定长的值类型保存在栈内存上，值类型没有引用计数、不会被“共享”、只需用常数时间就能完成一次“复制”，等等，这些特性使得值类型相比引用类型拥有更快的读写速度。同时，值类型的方法可以在编译期进行“内联”优化，更多的值类型意味着更大的优化空间。Swift 不但通过协议扩展的方式填补了传统面向对象编程语言中值类型代码复用的鸿沟，同时应用 Copy-on-Write 技术还可免于陷入“绝对复制”带来的性能问题。在 Swift 的代码世界中，值类型得到了空前的解放。如果你正在使用 Swift 3.0 版本，那么你应该已经发现了，那些摘掉了“NS”前缀的 Swift 原生对象，大部分都变成了值类型，毫无疑问，这种演变在后续的版本中还会继续进行下去。所以从内存的角度来看，面向协议编程正朝着面向栈内存编程的方向前进。

这本书诞生于我在重新研究 WWDC 2015 视频的阶段，我所编著的《Swift 开发手册：技巧与实战》首印即将售罄，该书的责编安娜（也是本书的责编）与我商讨重印的细节。受写作时 Swift

版本的影响，《Swift 开发手册：技巧与实战》是一本面向对象编程的教材；所以在深思熟虑之后，我决定放弃该书的重印，重新创作一本面向协议编程的教材，也就是你现在看到的《Swift：面向协议编程》。在本书的创作过程中，我保留了旧作中的部分目录和示例，加入了我对 WWDC 2015 以及 WWDC 2016 重新研究后的心得体会、个人在工作中的面向协议编程实战经验以及对编程模式的思考。Swift 开源之后，除了可以在 iOS 平台应用外，在其他平台也有强劲表现，所以本书的内容编排侧重于语言本身及编程方式的思考，弱化了 iOS 开发的教学。由于本书的创作时间比较久，所以主体内容是基于 Swift 2.2 版本创作的。在 Swift 3.0 版本发布之后，我第一时间投身到新版本的学习中，在本书的第二次排版时加入了 Swift 3.0 的改动，力争做到全面和准确。可以预见，3.0 版本不会是 Swift 的最后一个版本，所以请读者在学习时以当前的语言版本做参考。

最后感谢电子工业出版社的编辑安娜为本书所做的付出，多次合作我们已经建立了深厚友谊与足够的默契。感谢公司对我的信任，让我如愿以偿地成为了一名职业 iOS 开发者，获得了很多应用 Swift 的机会。感谢智维，创业辛苦仍不失对技术的热情，在百忙中抽出时间帮我写序，祝趣直播越做越好。感谢帮我写书评的王巍大神、InfoQ 主编徐川老师、还有圈内好友 RxSwift 大神小青和 SwiftGG 翻译组的前辈羊叔，你们的支持是我前进的动力，同时也感谢唐巧、付若愚和李明杰三位老师的关注。感谢师父时永昌对我的关怀和指导，感谢和我合作开发第一个 Swift 项目的同事杨帅以及同组的所有同事，感谢帮我纠错的同事蒙朦，感谢共同学习进步的其他 TStar 们（白建国、薛刚、施洁、邓飞）。

为了更好地解答读者疑问，第一时间分享 Swift 心得，我开通了微信订阅号：SwiftTime 雨燕时光，欢迎读者订阅交流。

走心的 Swift 翻译组地址：http://swift.gg。

CSDN 个人博客地址：http://blog.csdn.net/cg1991130。

本书第 4 章和第 5 章的 Demo 地址：https://pan.baidu.com/s/1gfMC7RX。

目录

第 1 章

搭建 Swift 开发环境

正所谓“工欲善其事，必先利其器”，在开始学习 Swift 之前，我们需要做好相关的准备。由于 Swift 是苹果公司的“亲儿子”，所以你可以在 Mac 上快速开启你的 Swift 之旅，本书使用的开发工具是苹果的 Xcode。

1.1 Swift 介绍

1.1.1 Swift 的前世今生

在阅读本书之前，可能你已经从其他渠道了解了一些 Swift 的相关信息，下面简单介绍一下 Swift 语言的前世今生。作为编程语言界的“小鲜肉”，Swift 是苹果公司在 2014 年 WWDC（苹果全球开发者大会）上发布的全新的编程语言。Swift 是供 iOS 和 OS X 应用编程的新编程语言，基于 C 和 Objective-C，没有 C 的一些兼容约束。Swift 采用了安全的编程模式，并添加新的功能，以使得编程更加简单、灵活、有趣。界面则基于深受广大码民喜爱的 Cocoa 和 Cocoa Touch 框架，展示了软件开发的新方向。

有人笑言 Swift 语言是语言进化链顶端的语言，因为它融合了很多现代编程语言的优点，加入了诸如闭包这样的高级语言特性。在语法结构上，Swift 有点类似于 JavaScript 这样的脚本语言，但更加简洁优雅。

2010 年 7 月，LLVM 编译器的原作者，暨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，还有一个 Dogfooding 团队大力参与其中。至 2014 年 6 月发表时，Swift 历经了大约 4 年的开发期。克里斯在开发 Swift 之前的一项伟大成就是为苹果公司开发了 LLVM 编译框架。由于他在编译框架方面的丰富经验，使得 Swift 不但语法简洁，而且对编译期的速度也有所优化，读者在使用 Swift 进行开发时一定深有感触。

开源之后，Swift 得到了更多开发者的青睐，在最近几个月的 Tiobe 世界编程语言排行榜上，Swift 都反超了 Objective-C，排名稳中有升，2016 年 8 月的 Tiobe 世界编程语言排行榜如图 1.1 所示。

Aug 2016	Aug 2015	Change	Programming Language	Ratings	Change
1	1		Java	19.010%	-0.26%
2	2		C	11.303%	-3.43%
3	3		C++	5.800%	-1.94%
4	4		C#	4.907%	+0.07%
5	5		Python	4.404%	+0.34%
6	7	^	PHP	3.173%	+0.44%
7	9	^	JavaScript	2.705%	+0.54%
8	8		Visual Basic .NET	2.518%	-0.19%
9	10	^	Perl	2.511%	+0.39%
10	12	^	Assembly language	2.364%	+0.60%
11	14	^	Delphi/Object Pascal	2.278%	+0.87%
12	13	^	Ruby	2.278%	+0.86%
13	11	v	Visual Basic	2.046%	+0.26%
14	17	^	Swift	1.983%	+0.80%
15	6	vv	Objective-C	1.884%	-1.31%
16	37	^^	Groovy	1.637%	+1.27%
17	20	^	R	1.605%	+0.60%
18	15	v	MATLAB	1.538%	+0.31%

图 1.1　2016 年 8 月的 Tiobe 世界编程语言排行榜

1.1.2　Swift 与 Objective-C

我们知道，以前进行 iOS 开发时使用的是 Objective-C（以下简称 OC）这门古老的语言，Swift 语言的发布对于熟悉 OC 开发的程序员来说是一件令人兴奋的事情。因为 OC 是 C 语言的超集，比较像 C++这样的传统面向对象语言，而没有很多现代高级语言的特性。并且由于 OC 是一门消息传递语言，而不是传统的函数调用语言，因此 OC 里面中的括号语法使得很多从其他编程语言转投 OC 开发的程序员很不适应。

但是这并不代表 OC 不好。任何一门新兴的语言都在不断变化之中，由于 Swift 的发布时间较短，而笔者学习 Swift 语言的时候还处在 Swift 语言的元年，笔者的学习过程跨越了 Swift 1.1 版本到 Swift 3 版本，语法有很多变动。如果你正在维护一个大型项目，那么语法的变动肯定是你不想看到的。所幸苹果承诺 3.0 版本将是 Swift 最后一个“破坏性”升级的版本。对于有志于从事 iOS 开发工作或者是服务端开发的读者，使用 Swift 可能是一件“当下痛苦，未来愉悦”的事情。本书在介绍 Swift 语法时也会穿插 OC 中的一些代码做对比，让读者更加了解两种语言的差异。

最后，笔者建议在学习 Swift 语言的同时，应熟悉 OC 语言，两件兵器同时在手，笑傲职场，岂不更加游刃有余！

1.2　Mac OS X 操作系统

OS X 是苹果公司为 Mac 系列产品开发的专属操作系统（新版本已经正式更名为 Mac OS 了），基于 UNIX 系统，有一定 Linux 基础的人使用 OS X 时会更加得心应手。OS X 操作系统的获取方式有两种，最简单的一种是买一台苹果电脑，无论是哪个系列，这些电脑都预装了 OS X 操作系统。

如果你不打算购买苹果电脑，另外一种方式是在 Windows 环境下用虚拟机安装 OS X 操作系统，也就是我们通常所说的“黑苹果”，网上有详细的教程，笔者在此不做过多的介绍。但需要强调的是，“黑苹果”下的开发有着各种各样的问题，会拖累我们的学习进度，如果你只是想尝尝鲜，了解一下 Swift 语言，那么你大可不必购买一台真正的苹果电脑。但是对于一个有志于在现在或者将来进行 iOS 系统开发的读者来说，一台真正的苹果电脑是必不可少的；因为不仅企业中的 iOS 开发全部是在苹果电脑上进行的，甚至一些其他领域的开发也会选择苹果电脑，而苹果电脑的易用性和良好的操控性可以大大提升你的编程舒适度。