

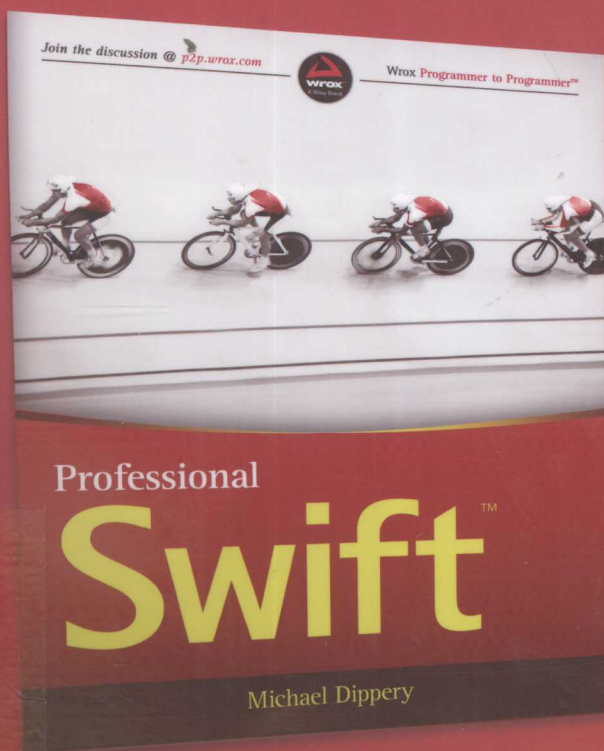
Join the discussion @ p2p.wrox.com



Wrox Programmer to Programmer™

移动开发经典丛书

Swift 2.2高级编程



[美] Michael Dippery
冯宝隆

著
译



清华大学出版社

移动开发经典丛书

Swift 2.2 高级编程

[美] Michael Dippery 著

冯宝隆 译

清华大学出版社

北 京

Michael Dippery
Professional Swift

EISBN: 978-1-119-01677-9

Copyright © 2015 by John Wiley & Sons, Inc., Indianapolis, Indiana

All Rights Reserved. This translation published under License.

Trademarks: Wiley, Wrox, the Wrox logo, Programmer to Programmer, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates, in the United States and other countries, and may not be used without written permission. Swift is a trademark of Apple, Inc. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。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。

北京市版权局著作权合同登记号 图字: 01-2016-1648

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

本书封面贴有 Wiley 公司防伪标签, 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

Swift 2.2 高级编程 / (美) 迈克尔·迪普瑞(Michael Dippery) 著; 冯宝隆 译. —北京: 清华大学出版社, 2016
(移动开发经典丛书)

书名原文: Professional Swift

ISBN 978-7-302-44539-5

I. ①S... II. ①迈... ②冯... 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 174426 号

责任编辑: 王 军 韩宏志

装帧设计: 牛静敏

责任校对: 成凤进

责任印制: 宋 林

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

印 装 者: 清华大学印刷厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 17.25 字 数: 431 千字
版 次: 2016 年 9 月第 1 版 印 次: 2016 年 9 月第 1 次印刷

印 数: 1~3500

定 价: 49.80 元

产品编号: 064485-01

译者序

作为 Apple 公司推出的全新语言，Swift 吸收了 C、C++、Objective-C、Javascript、Python、C#等诸多语言的优秀特性，并提供了极强的安全性。目前该语言仍然在蓬勃发展，本书的英文版使用的是 Swift 1.0 的语法，而在翻译本书时 Swift 已经进化到 2.2 版，其语法变化较大。为了让读者能够学习和领略最新的 Swift 知识，本书的中文版根据 Swift 2.2 的语法对代码进行了更新，这样读者就可以直接使用最新版本的 Xcode 执行本书代码而不会出错。

目前 Swift 语言在 TIOBE 中的排名仍在快速攀升，已跻身前 20 名，已经非常接近 Objective-C 的排名。Apple 公司已经宣布将 Swift 开源，Swift 除了支持 Apple 公司自身的 iOS 和 OS X 系统外，还支持 Linux 操作系统。这样开发人员不仅可使用 Swift 语言开发前端应用，还可以开发后端应用。还有一些组织和公司正将 Swift 移植到 Android 和 Windows，这样还可以使用 Swift 开发跨平台应用。总之，学习 Swift 能获得丰厚回报，可使你蜕变为一名全栈工程师。

本书面向具有一定经验的 Objective-C 开发者，通过阅读本书可快速掌握 Swift 的基础和高级知识，从而快速使用 Swift 开发自己的应用程序。

这里要感谢清华大学出版社的编辑们，他们在本书的编辑和出版过程中倾注了极大心血，正是由于编辑们的辛勤劳动，才使得本书的译本在最短时间内与广大读者见面。

在本书翻译过程中，译者本着严谨的翻译态度，字斟句酌，将大量心血和汗水投入本书，力求为读者献上一本经典译作。本人虽然有十几年的使用多种语言进行开发的经验，但由于 Swift 是一门新语言，在翻译过程中难免会有一些错误和问题，还希望各位 Swift 高手批评指正。我们的联系邮箱为 wkservice@vip.163.com 或 fbswiftpro@sina.com。

冯宝隆

作者简介

Michael Dippery 是一位供职于旧金山工业光魔公司的软件开发者，拥有布尔耐克大学计算机科学专业的硕士学位。之前他还在旧金山的 SocialCode 公司和纽约的 The New York Review of Books 公司工作过。尽管他大部分时间都在编写 Python 代码，但他一直很喜欢 Objective-C。他为 OS X 中的 Adium 和 Colloquy 项目贡献代码，并编写和发布了许多 Objective-C 开源库。在他不使用 Objective-C 时，他热衷于使用 Haskell 和 Erlang 编写代码。他期待 Swift 的潜力能够改善 iOS 和 OS X 软件开发。在业余时间，Michael 喜欢阅读、写小说以及摄影。

致 谢

感谢我的父母 Steven 和 Kim，他们坚定不移地支持我的教育和事业，尽管他们从来没有完全理解我做的是什么工作。

感谢 Liza Veloz，她给了我成为技术作家和程序员的第一个机会。

感谢 Luiz Felipe Perrone 教授，在过去十年里对我关怀备至，精心指导我，与我结下深厚友谊。

感谢 Mary James，是她邀请我编写本书。

前言

iOS 的流行将许多程序员吸引到 Apple 公司的平台。很长时间以来，OS X 和 iOS 上的软件都是用 Objective-C 编写的，这门语言是在 20 世纪 80 年代开发的，它非常适合被描述为 C 和 Smalltalk 的混合体。Objective-C 缺乏现代程序员期盼的许多编程语言特性。近十年来，Apple 一直在调整和改进 Objective-C，但许多程序员仍然觉得它是一门外星语言。

Swift 是一门现代编程语言，在建立该语言时使用了近 30 年来在编程语言研究领域获得的成果。开发该语言的研究者还开发了 clang，它是 Apple 用来构建 OS X 和 iOS 软件的现代编译器工具链。Swift 包含了许多现代编程语言应该具有的特性。它支持一个强类型系统来避免在 C 和 Objective-C 编程中普遍存在的许多错误。它还提供一个精炼的支持静态和可计算属性的类系统。它支持闭包，并将函数视为一级类对象，从而可将它们保存到变量或作为参数传递给函数和方法。在阅读本书时，你还会继续发现 Swift 提供的许多高级特性。

Swift 经过了重新设计，从而成为一种适合在 Apple 平台上进行系统开发和应用程序开发的现代编程语言。Apple 已经为 Swift 提供了大量支持，且该语言仍在不断改进。Swift 代表了 iOS 和 OS X 软件开发的未来，在这些平台上进行开发的程序员也期望 Swift 在这些系统中日益占据主导地位。

本书读者对象

本书面向具有在 OS X 和 iOS 上开发 Objective-C 应用程序经验的软件开发者。Swift 的知识和经验虽然有用，但对于理解本书的内容来说并不是必需的。本书引入的 Swift 关键概念甚至对于高级 iOS 和 OS X 程序员来说也是陌生的。本书使用了几个小项目来说明其引入的概念。本书还大量使用了 playground，Xcode 的这个新特性允许不建立整个 Xcode 项目就能编写 Swift 代码并与 Swift 代码交互。

本书第 1 章主要介绍 Swift 的语法和语义，但没有介绍该语言的任何高级特性。第 2 章介绍 Xcode 的使用，其中包括 playground。如果已经对 Swift 和 Xcode 比较熟悉，可以跳过第 1 章和第 2 章而直接从第 3 章开始。

本书主要内容

本书首先对 Apple 于 2014 年 6 月发布的 Swift 编程语言做了初步介绍，然后快速移

到更高级的 Swift 编程主题，包括 Xcode 中新的 playground 特性；类、结构和枚举；并发编程以及 Swift 类型系统的高级特性。本书还介绍 Swift 程序员需要经常处理的实际问题，包括使用 JSON 与 Web 服务交互、Core Data 以及在 Swift 项目中使用 C 和 Objective-C 代码。本书还演示 Xcode 的一些高级特性，包括如何使用 Xcode 内建的调试器，并讨论 Swift 和 Objective-C 运行时的底层细节。因为在 OS X 和 iOS 上编写软件时无法避免使用 C 语言，所以附录中提供了 C 语言的简要介绍。

本书结构

本书共分两部分。第 I 部分介绍了 Swift 编程语言和 Swift 程序员可能遇到的实际问题。第 II 部分深入讨论 Swift 错综复杂的细节。

- 第 1 章：回顾 Swift 编程语言的语法和语义。这对于那些对该语言陌生或想要快速了解该语言的程序员来说非常有用。
- 第 2 章：介绍 Xcode 中与 Swift 有关的变化。该章介绍 Xcode 的新特性 playground，它提供了一种以交互方式使用 Swift 代码的方法。
- 第 3 章：以比第 1 章更详细的方式介绍 Swift 的类、结构和枚举，并为 Swift 中的面向对象编程打下坚实基础。
- 第 4 章：讨论 Swift 上下文环境中的并发编程以及 iOS 和 OS X 中的 Foundation 框架。
- 第 5 章：演示如何使用 JSON 与远程 Web 服务通信。
- 第 6 章：讨论如何使用 Core Data 存储和搜索程序使用的信息。
- 第 7 章：演示如何使用协议和类扩展来扩展类。
- 第 8 章：详细介绍 Swift 的类型系统，并演示如何使用 Swift 的强类型系统来编写错误更少的代码。
- 第 9 章：演示如何在 Swift 程序中混合 C 和 Objective-C 代码。
- 第 10 章：介绍调试器并演示如何跟踪和修复代码中的问题。
- 第 11 章：讨论 Swift 和 Objective-C 的运行时结构，并演示程序是如何在 iOS 和 OSX 中被加载和执行的。
- 附录 A：介绍 C 语言引入的特性和概念。

阅读本书的条件

若要运行本书中的例子，需要具备下列条件：

- 运行 OS X 10.9 或更新版本的 Mac 电脑。
- Xcode 6。
- 如果想要在真实设备而不是 Xcode 的 iOS 模拟器中运行本书的例子，则还需要 iPhone、iPod Touch 或 iPad。

源代码

可以在 www.wrox.com 中通过英文 ISBN(本书的英文 ISBN 为 978-1-119-01677-9)搜索本书来找到它的源代码。Wrox 当前出版的所有书籍的源代码都可从 www.wrox.com/dynamic/books/download.aspx 下载。

也可登录 www.tupwk.com.cn/downpage, 输入本书中文书名或中文 ISBN, 下载源代码。注意, 本书译者针对最新版本对源代码做了更新, 更新后的代码可从该网站下载。

www.wrox.com 上的大多数源代码都被压缩为了 .zip、.rar 或类似的适合特定平台的文档格式。在下载源代码后, 需要使用对应的压缩工具来解压缩。



注意: 因为许多书籍都有类似的标题, 所以可通过搜索 ISBN 来更容易地查找需要的书籍; 本书的英文 ISBN 为 978-1-119-01677-9。

在下载源代码后, 可使用自己喜欢的压缩工具来解压缩。另外还可以进入位于 www.wrox.com/dynamic/books/download.aspx 处的 Wrox 代码下载主页中来查看本书和所有其他 Wrox 书籍的源代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误, 但是错误总是难免的, 如果你在本书中找到了错误, 例如拼写错误或代码错误, 请告诉我们, 我们将非常感激。通过勘误表, 可以让其他读者避免受挫, 当然, 这还有助于提供更高质量的信息。

要在网站上找到本书的勘误表, 可以登录 <http://www.wrox.com>, 通过 Search 工具或书名列表查找本书, 然后在本书的细目页面上, 单击 Book Errata 链接。在这个页面上可以查看 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表, 网址是 www.wrox.com/misc-pages/booklist.shtml。

如果在 Book Errata 页面上没有看到你找出的错误, 请进入 www.wrox.com/contact/techsupport.shtml, 填写表单, 发电子邮件, 我们就会检查你的信息, 如果是正确的, 就在本书的勘误表中粘贴一个消息, 我们将在本书的后续版本中采用。

p2p.wrox.com

P2P 邮件列表是为作者和读者之间的讨论而建立的。读者可以在 p2p.wrox.com 上加入 P2P 论坛。该论坛是一个基于 Web 的系统, 用于传送与 Wrox 图书相关的信息和相关技术, 与其他读者和技术用户交流。该论坛提供了订阅功能, 当论坛上有新帖子时, 会

给你发送你选择的主题。Wrox 作者、编辑和其他业界专家和读者都会在这个论坛上进行讨论。

在 <http://p2p.wrox.com> 上有许多不同的论坛，帮助读者阅读本书，在读者开发自己的应用程序时，也可以从这个论坛中获益。要加入这个论坛，必须执行下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读其内容，单击 Agree 按钮。
- (3) 提供加入论坛所需的信息及愿意提供的可选信息，单击 Submit 按钮。
- (4) 然后就可以收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。



提示：不加入 P2P 也可以阅读论坛上的信息，但只有加入论坛后，才能发送自己的信息。

加入论坛后，就可以发送新信息，回应其他用户的帖子。可以随时在 Web 上阅读信息。如果希望某个论坛给自己发送新信息，可以在论坛列表中单击该论坛对应的 **Subscribe to this Forum** 图标。

对于如何使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作原理，以及许多针对 P2P 和 Wrox 图书的常见问题解答。要阅读 FAQ，可以单击任意 P2P 页面上的 FAQ 链接。

目 录

第 I 部分 构建 Swift 应用

第 1 章	Swift 入门	3
1.1	什么是 Swift	3
1.2	为什么要学习 Swift	4
1.3	使用常量和变量	5
1.3.1	理解 Swift 数据类型	6
1.3.2	使用类型注解	12
1.3.3	使用类型推导简化类型注解	13
1.3.4	使用类型别名简化代码	14
1.4	使用运算符	15
1.4.1	使用基本运算符	15
1.4.2	使用复合赋值运算符	16
1.4.3	使用溢出运算符	16
1.4.4	使用范围运算符	17
1.4.5	使用逻辑运算符	17
1.4.6	使用比较运算符	18
1.4.7	使用自定义运算符	19
1.5	使用控制流进行判断	19
1.5.1	使用条件语句	19
1.5.2	使用循环	22
1.5.3	控制转移语句	24
1.6	使用枚举组织类型	25
1.7	使用函数	26
1.7.1	声明函数	27
1.7.2	指定参数名称	27
1.7.3	定义默认参数	28
1.7.4	指定可变参数	28
1.7.5	指定常量、变量和输入-输出参数	29
1.7.6	函数类型	30

1.7.7	使用闭包	31
1.8	本章小结	33
第 2 章	编写 Swift 程序	35
2.1	设置 Xcode	35
2.1.1	从 Apple 开发者门户网站下载	36
2.1.2	下载 Xcode	36
2.2	试验 Playground	37
2.2.1	在 Xcode 中使用 Playground	38
2.2.2	使用 Playground 的简单示例	38
2.2.3	一个更令人兴奋的 Playground 示例	40
2.3	在 Xcode 中编写 Swift 程序	41
2.3.1	建立一个新的 Swift 项目	42
2.3.2	建立 Swiftisms 应用	45
2.4	调试 Swift 应用	51
2.4.1	理解断点类型	52
2.4.2	设置断点	52
2.4.3	使用调试器	53
2.5	本章小结	57
第 3 章	类、结构和枚举	59
3.1	使用类和结构	59
3.1.1	定义类和结构	60
3.1.2	初始化类和结构	61
3.1.3	析构类	67
3.2	使用属性	67
3.2.1	使用简单属性	67
3.2.2	使用延迟属性	69
3.2.3	使用计算属性	69

3.2.4	观察属性的变化	70
3.3	理解方法	71
3.4	理解类和结构的不同之处	73
3.5	使用枚举	75
3.6	本章小结	76
第 4 章	Swift 中的并发	77
4.1	什么是并发	77
4.1.1	理解进程	78
4.1.2	理解线程	79
4.1.3	为什么使用并发	80
4.1.4	设计并行程序	80
4.2	iOS 和 OS X 中的并发	81
4.2.1	理解事件循环	81
4.2.2	理解运行循环	81
4.2.3	理解异步方法	82
4.2.4	理解 NSTimer	83
4.2.5	理解 NSThread	85
4.2.6	理解 Grand Central Dispatch	86
4.3	本章小结	88
第 5 章	与 Web 服务交互	89
5.1	理解 Web 服务	89
5.2	实现 Web 服务客户端	91
5.2.1	注册 Firebase 账号	91
5.2.2	布局用户界面	91
5.2.3	挂接 Add Album View Controller	96
5.2.4	建立专辑模型	98
5.2.5	与 Firebase 通信	99
5.2.6	连接视图和数据模型	106
5.2.7	连接过渡	110
5.2.8	运行应用	111
5.3	本章小结	112
第 6 章	使用 CoreData 存储数据	113
6.1	什么是 Core Data	113
6.2	Core Data 栈	114
6.3	在 Swift 中使用 Core Data	115

6.3.1	设置 Birthdays 应用	116
6.3.2	获取用户输入	122
6.3.3	显示数据	133
6.4	本章小结	142

第 II 部分 高级 Swift 概念

第 7 章	扩展类	145
7.1	使用类扩展	145
7.1.1	向类中添加方法	146
7.1.2	向类中添加属性	148
7.1.3	在扩展中修改类	148
7.2	使用协议指定行为	149
7.2.1	采用协议	149
7.2.2	声明属性和方法	150
7.2.3	使用可选方法和属性	152
7.2.4	协议是类型	153
7.2.5	继承其他协议	156
7.2.6	在类扩展中采用协议	157
7.3	使用泛型	158
7.3.1	泛型类	159
7.3.2	使用类型约束	160
7.4	本章小结	162
第 8 章	高级数据类型	163
8.1	使用枚举和代数数据类型	163
8.1.1	匹配模式	164
8.1.2	通过 JSON 串联这些知识	166
8.2	使用可选类型	168
8.2.1	使用 null 值	168
8.2.2	拒绝 null 值	168
8.2.3	理解 Swift 中的 null	169
8.2.4	可选值链	170
8.3	理解类型转换	171
8.4	使用元组组织值	173
8.5	自定义运算符	174
8.5.1	自定义运算符	179
8.5.2	定义优先级和结合性	179
8.5.3	和运算符有关的最后 一些话	182

8.6 使用函数和闭包.....	183	10.2.2 添加滑动条.....	206
8.7 本章小结	184	10.2.3 完成项目.....	207
第 9 章 桥接 Swift 和 Objective-C.....	185	10.3 输出值.....	209
9.1 Objective-C 的继承者	185	10.4 使用调试器.....	211
9.2 引入命名空间和模块.....	186	10.5 使用 lldb 检查错误	211
9.2.1 使用命名空间组织代码	187	10.5.1 基本断点.....	212
9.2.2 使用模块分发代码.....	188	10.5.2 使用 lldb 检查程序	213
9.3 Swift 和 Objective-C 如何 交互	190	10.5.3 移除或禁用断点.....	217
9.3.1 在 Objective-C 中使用 Swift 类.....	190	10.5.4 设置条件断点	217
9.3.2 在 Swift 中使用 Objective-C	194	10.5.5 设置符号断点	218
9.4 在 Swift 中使用 C 和 C++代码	197	10.5.6 探索调试器的使用.....	219
9.4.1 使用 C 标量类型	198	10.6 本章小结.....	223
9.4.2 使用 C 指针访问内存.....	198	第 11 章 Swift 运行时	225
9.4.3 在 Swift 中使用全局常量和 定义	199	11.1 运行时的含义	225
9.4.4 在 Swift 中调用 C 代码	200	11.2 理解 Objective-C 运行时	226
9.4.5 在 Swift 中使用 C++代码.....	201	11.2.1 在 Objective-C 中动态 委派方法.....	227
9.5 本章小结	202	11.2.2 消息传递.....	228
第 10 章 调试 Swift 应用	203	11.2.3 解析方法.....	229
10.1 调试的艺术.....	203	11.2.4 消息和方法.....	233
10.2 创建 CircleView	204	11.3 研究 Swift 运行时	234
10.2.1 绘制圆形.....	204	11.3.1 理解虚拟方法表	234
		11.3.2 加载 Swift 程序.....	237
		11.4 本章小结.....	241
		附录 A C 语言概述	243

第 I 部分

构建Swift应用

- 第 1 章：Swift 入门
- 第 2 章：编写 Swift 程序
- 第 3 章：类、结构和枚举
- 第 4 章：Swift 中的并发
- 第 5 章：与 Web 服务交互
- 第 6 章：使用 Core Data 存储数据



第 1 章

Swift 入门

本章内容

- 了解 Swift
- 常量和变量的声明以及 Swift 数据类型的使用
- 使用运算符转换值
- 使用条件语句和循环控制代码的执行
- 定义和使用 Swift 的枚举数据类型
- 理解、声明和使用函数、匿名函数和闭包

本章介绍 Swift 编程语言的关键概念以及该语言的新语法和数据类型。本章并不是 Swift 的入门指南，它的目的是让已用过 Swift 的程序员复习一下该语言的知识。之前在 iOS 和 OS X 平台上使用 C 和 Objective-C 编程的知识对于理解本章仍然有用，而且本章中的知识对于没有 C 和 Objective-C 经验的人也有意义。

如果读者已经熟悉 Swift 编程语言的基础知识，可直接跳转到本书的第 2 章。第 2 章还介绍了如何使用 Xcode 的 playground 新特性，使用该特性可在编写 Swift 代码时直接看到其效果。本章的例子可直接输入到 playground，从而可立即看到执行结果，而不用创建 Xcode 项目和编译代码。

1.1 什么是 Swift

Swift 是一门全新的语言，由 Apple 公司开发，用来为开发 iOS 和 OS X 的 Objective-C 提供一种备选方案。尽管该语言被设计为可和 Objective-C 以及 C 和 C++无缝交互，但它并不是 Objective-C 的进化，而是一门和其血统大不相同的全新语言。它抛弃了很多经典的

Objective-C 语言特性，同时引入了大量使 iOS 和 OS X 程序开发更快、更安全的新特性，使得开发过程变得更便捷。

Swift 是一个开发了几年的产品，在其研发过程中参考了很多现有语言。Swift 远远不是 Objective-C 的简单改进，它的语言特性源自诸多编程语言，如 Haskell、C#、Ruby、Python 和 Rust 等。Swift 还吸收了 Cocoa 和 Cocoa Touch 框架的很多特性，如键-值观察。Swift 编译器还利用了很多在创建 LLVM 和 Objective-C 的 clang 编译器时获得的研究成果和经验。

与 Objective-C 不同，Swift 不是动态类型的语言。相反，它使用静态类型来帮助确保程序的完整性和安全性。Swift 还避免了许多 C 语言(包括 Objective-C)的固有问题，尤其是在确保内存完整性方面。尽管有经验的 Objective-C 开发者可能会认为这会丧失程序设计和构造的灵活性，但 Swift 的新特性会让程序的编写更简单和更容易，而且仍为程序员保留了很大的自由度。Apple 想让 Swift 既适合应用开发又适合系统编程，该语言的设计也反映了这一点。

幸运的是，Apple 已经非常注意将 Swift 整合到现有的生态系统。Swift 可以无缝地操作现有的 Objective-C 代码、库和框架，对于用 C 和 C++编写的代码也是如此。通过一些额外的设置甚至可以在程序中混合使用这些语言。Xcode 完全支持 Swift 代码，从而开发者可继续使用其在使用 Objective-C 进行开发时已经熟悉的开发工具，如编译器和调试器等。

实际上，Swift 已经为 Xcode 生态系统引入了一个新工具：playground。playground 是一个交互式开发环境，它会实时反馈代码块的执行结果。Swift 还支持“读取-求值-输出”循环(read-eval-print loop, REPL)，该过程可用于在控制台中测试 Swift 代码片段是否正确。有关 playground 的更多细节可参见本书的第 2 章。

Swift 是 iOS 和 OS X 开发的巨大飞跃。尽管 Objective-C 会在相当长的时间内被继续支持，但 Swift 是 iOS 和 OS X 开发的未来发展方向，掌握 Swift 引入的新技术是非常重要的。所幸开发者目前已掌握的 iOS 和 OS X 开发知识仍然十分重要，即使是完全使用 Swift 编写程序也是如此。最重要的是，Swift 的新特性使得 iOS 和 OS X 应用的编写变得比以前更刺激有趣。

在深入到 Swift 开发的细节前，首先应该熟悉一下 Swift 的基本概念。如果读者已经牢牢掌握了这些概念，可以跳转到本书的第 2 章。

1.2 为什么要学习 Swift

这里有一个更大的问题，那就是为什么不使用 Objective-C 编写 iOS 和 OS X 应用而是去学习 Swift？事实是 Objective-C 不会在短时间内消失。iOS 和 OS X 中的大多数框架都是用 Objective-C 编写的，另外还有大量可供选择的第三方库和框架，以及大多数 iOS 和 OS X 开发教程也都是用 Objective-C 编写的。透彻掌握 Objective-C 的知识对于 iOS 和 OS X 这