

TURING

图灵原创

- 畅销书《iOS 开发指南》作者最新力作
- 基于 Swift 2.x, 教你如何从零开始开发并上架应用
- 涵盖 Cocoa Touch 框架、混合编程 (Objective-C、C/C++ 与 Swift)、并发编程、SpriteKit 等

关东升 著

从零开始学 *Swift*



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创

从零开始学 Swift



关东升 著

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

从零开始学Swift / 关东升著. -- 北京 : 人民邮电出版社, 2016.3
(图灵原创)
ISBN 978-7-115-41589-9

I. ①从… II. ①关… 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018385号

内 容 提 要

本书基于 Swift 2.x, 通过大量案例全面介绍苹果平台的应用开发。全书共分 5 部分, 包括 Swift 语法篇、Cocoa Touch 框架编程篇、混合编程篇、进阶篇、项目实战篇, 囊括了开发工具介绍、语言基础知识、编写 HelloWorld 程序, 直至应用在 App Store 上线的所有内容。此外, 本书前 20 章专门配有同步练习, 以帮助读者掌握所学知识。

本书适合 iOS 开发者、其他移动平台开发者及计算机专业大学生参考阅读, 也非常适合作为培训教材使用。

-
- ◆ 著 关东升
责任编辑 毛倩倩
责任印制 杨林杰
 - ◆ 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: 880×1230 1/16
印张: 38.75
字数: 1250千字 2016年3月第1版
印数: 1-3 500册 2016年3月河北第1次印刷
-

定价: 99.00元

读者服务热线: (010)51095186转600 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

前 言

Swift语言推出已经一年多了，历经多个版本，现在已经升级到Swift 2.x。Swift 2.x较1.x有很多变化。更可喜的是，2015年12月3日苹果将Swift开源，这是苹果为开源社区做出的贡献，Swift程序可以在Linux下编译和运行了。这一年多的变化中，Swift已经迎来了一个新的时代。

智捷课堂与图灵教育在Swift推出之际就合作出版了国内第一本Swift图书——《Swift开发指南》，当时是基于Swift 1.x编写的。随着Swift版本的升级，广大读者也亟需了解2.x的新特性，并学习更加深入的Swift内容。基于上述原因，我们又编写了这本《从零开始学Swift》。

内容和组织结构

本书是我们团队编写的iOS系列图书之一，旨在使从事iOS开发的广大读者掌握苹果Swift语言，并使原来有Objective-C开发经验的人能够快速转型到Swift iOS应用开发上来。全书共分5部分。

第一部分为Swift语法篇，共21章，介绍了Swift的一些基础知识。

第1章介绍了Swift的开发背景以及本书约定。

第2章介绍了运行Swift程序的交互式方式和编译为可执行文件方式，介绍了Swift的程序结构。希望大家能够熟悉Xcode工具的使用，以及如何在Linux下搭建Swift开发环境。

第3章介绍了Swift的基本语法，其中包括标识符、关键字、常量、变量、表达式和注释等内容。

第4章介绍了Swift的基本运算符，包括算术运算符、关系运算符、逻辑运算符、位运算符等。

第5章介绍了Swift的数据类型，例如UInt8、Int8和Double等，此外还有元组（tuple）等类型。

第6章介绍了Swift中的字符和字符串，以及字符串可变性和字符的比较等内容。

第7章介绍了Swift语言的控制语句，包括分支语句（if和switch）、循环语句（while、repeat-while、for和for-in）和跳转语句（break、continue、fallthrough和return）等。

第8章介绍了Swift提供的几种数据结构的实现：数组、字典和Set集合。

第9章介绍了Swift中的函数：Swift中的函数可以独立存在，即全局函数；也可以在别的函数中存在，即函数嵌套；还可以在类、结构体和枚举中存在，即方法。

第10章介绍了Swift语言中的闭包，包括闭包的概念、闭包表达式、尾随闭包和捕获值等内容。

第11章首先介绍了现代计算机语言中面向对象的基本特性，然后介绍了Swift语言中面向对象的基本特性，主要包括枚举、结构体和类等基本概念及其定义。最后，该章还介绍了Swift面向对象类型嵌套、可选类型和可选链等基本概念。

第12章介绍了Swift中属性和下标的基本概念及其使用规律，主要包括存储属性、计算属性、静态属性和属性观察者等重要的属性概念。此外，该章还介绍了下标的概念及使用。

第13章介绍了Swift语言中方法的概念、定义以及调用等内容，并讲述了实例方法和静态方法的声明和调用。

第14章介绍了Swift语言对象类型的构造过程和析构过程，以及构造函数和析构函数的使用方法。

第15章讨论了Swift语言的继承性，介绍了Swift中继承只能发生在类类型上，而枚举和结构体不能发生继承。此外，这一章还介绍了Swift中子类继承父类的方法、属性、下标等特征的过程，以及如何重写父类的方法、属性、下标等特征。

第16章介绍了Swift中扩展和协议的基本概念及重要性。该章具体讲述了如何扩展属性、方法、构造函数和下标。

第17章介绍了协议的概念、方法和属性，阐述了如何把协议当作一种类型使用，以及协议的继承和合成机制。另外，该章还说明了“面向协议编程”的重要意义。

第18章介绍了Swift中泛型的重要性，内容涵盖泛型概念、泛型函数和泛型类型，最后还介绍了泛型扩展。

第19章介绍了Swift中的内存管理机制，讲述了ARC内存管理的原理，以及如何解决对象间的强引用循环问题和闭包与引用对象之间的强引用循环问题。

第20章介绍了Swift 2之后的错误处理模式，带大家熟悉Swift 2.x错误处理，包括捕获错误、错误类型、声明抛出错误，以及函数或方法中抛出错误等内容。

第21章介绍了Swift编码规范，包括命名规范、注释规范、声明规范和代码排版。

第二部分为Cocoa Touch框架编程篇，共5章，介绍了基于Swift语言的Cocoa Touch框架。

第22章介绍了Foundation框架，以及通过Swift语言使用Foundation框架的方式方法。我们将带大家了解Foundation框架中的常用类：数字、字符串、数组、字典和NSSet等。此外，该章还阐述了文件管理、字节缓存、日期与时间、谓词NSPredicate和正则表达式。

第23章介绍了iOS开发基础知识，包括开发环境Xcode、iOS SDK和iOS API等内容。大家还将在此通过一个基于iPhone的HelloWorld示例项目了解iOS应用的基本运行原理。

第24章介绍了Core Foundation框架，以及通过Swift语言使用Core Foundation框架的方式方法。我们将带大家了解Swift原生数据类型、Foundation框架和Core Foundation框架等数据类型之间的对应关系，并介绍Core Foundation框架的内存管理知识。

第25章介绍了Cocoa Touch框架中的设计模式：单例模式、目标与动作、输出口、选择器、委托与数据源、通知机制、MVC模式和响应者链等。

第26章首先介绍了视图控制器和视图，以及应用界面构建层次，然后介绍界面构建（包括使用故事板、XIB文件和代码实现），最后介绍了开关、滑块、分段、工具栏、导航栏和表视图等控件。

第三部分为混合编程篇，共两章，主要介绍了Swift与Objective-C的混合编程，以及Swift与C/C++的混合编程等相关问题。

第27章介绍了Swift与Objective-C的混合编程，其中包括：同一应用目标中的混合编程和同一框架目标中的混合编程。

第28章介绍了Swift与C/C++的混合编程，其中包括：应用目标中的混合编程和框架目标中的混合编程。

第四部分为进阶篇，共3章，介绍了Swift进阶开发的相关知识点。

第29章首先介绍了Xcode调试工具，然后讲解设置和参考断点，最后介绍了iOS真机调试。

第30章介绍了并发编程，其中包括并发的相关概念。其中，GCD技术是学习的重点。

第31章介绍了苹果公司的2D游戏引擎SpriteKit，内容涵盖SpriteKit中的节点、精灵、场景切换、动作、粒子系统、游戏的音乐与音效、物理引擎等内容。

第五部分为项目实战篇，共3章，详细介绍了两个iOS App的开发过程，并最后介绍了App Store上架过程。

第32章完整地介绍了iPhone计算器应用的分析与设计、编程过程，并通过多种方式实现了iPhone计算器应用。此外，我们介绍了iOS的分层架构设计。

第33章完整地介绍了游戏项目的分析与设计、编程过程，使广大读者能够了解采用SpriteKit引擎开发手机游戏的过程。通过对本章的学习，读者能够将前面学到的知识串联起来。

第34章介绍了如何在App Store上发布应用，以及发布之前需要处理哪些问题。

本书配套网站

为了更好地为广大读者服务，我们专门为本书建立了一个服务平台（<http://51work6.com/swiftbook1.php>），大家可

以在此查看相关出版进度，并对书中内容发表评论，提出宝贵意见。

源代码

本书包括200多个完整的示例项目源代码，大家可以到本书配套网站<http://51work6.com/swiftbook1.php>下载，或者免费注册图灵社区（iTuring.cn）至本书主页下载。

同步练习

为了帮助读者消化吸收本书内容，我们在前20章后面都安排了数量不等的同步练习题。为了能够让广大读者主动思考，同步练习题的参考答案并没有放在书中，而是放在了本书网站上，我们还为此专门设立了一个讨论频道；大家也可以到图灵社区本书主页下载和参考。

勘误与支持

我们在本书网站上建立了一个勘误专区，以便及时地把书中的问题、失误和纠正信息反馈给广大读者。如果你发现了任何问题，均可以在网上留言或发送电子邮件到eorient@sina.com，我们会在第一时间回复你。此外，你也可以通过新浪微博（@tony_关东升）与我联系。

致谢

在此感谢图灵编辑王军花给我们提供的宝贵意见，感谢智捷课堂团队的赵志荣参与内容的讨论和审核，感谢赵大羽老师手绘了书中全部草图，并从专业的角度修改书中图片，力求更加真实完美地奉献给广大读者。此外，还要感谢我的家人容忍我的忙碌，以及他们对我的关心和照顾，使我能抽出这么多时间，投入全部精力专心编写此书。

由于时间仓促，书中难免存在不妥之处，请读者原谅。

关东升
2015年11月于北京

目 录

第一部分 Swift 语法篇

第 1 章 准备起航.....2

- 1.1 本书约定.....2
 - 1.1.1 示例代码约定.....2
 - 1.1.2 图示约定.....3
 - 1.1.3 函数和方法签名约定.....4
 - 1.1.4 承接上一行代码约定.....5
 - 1.1.5 代码行号约定.....5
- 1.2 Swift 开发工具.....6
 - 1.2.1 Xcode 开发工具.....6
 - 1.2.2 AppCode 开发工具.....12
- 1.3 本章小结.....13
- 1.4 同步练习.....13

第 2 章 第一个 Swift 程序.....14

- 2.1 使用 REPL.....14
 - 2.1.1 启动 Swift REPL.....14
 - 2.1.2 使用 Swift REPL.....15
- 2.2 使用 Playground.....16
 - 2.2.1 编程利器 Playground.....16
 - 2.2.2 编写 HelloWorld 程序.....18
- 2.3 使用 Web 网站.....21
- 2.4 通过 Xcode 创建 OS X 工程.....22
 - 2.4.1 创建 OS X 工程.....22
 - 2.4.2 编译和运行.....23
- 2.5 通过 AppCode 创建 OS X 工程.....24
 - 2.5.1 创建 OS X 工程.....24
 - 2.5.2 编译和运行.....25
- 2.6 使用 swiftc 命令.....27
 - 2.6.1 编译.....27
 - 2.6.2 运行.....28
- 2.7 代码解释.....28
- 2.8 在 Linux 下编译和运行 Swift 程序.....29
- 2.9 本章小结.....32
- 2.10 同步练习.....32

第 3 章 Swift 语法基础.....33

- 3.1 标识符和关键字.....33
 - 3.1.1 标识符.....33
 - 3.1.2 关键字.....34
- 3.2 常量和变量.....35
 - 3.2.1 常量.....35
 - 3.2.2 变量.....36
 - 3.2.3 使用 var 还是 let.....36
- 3.3 注释.....37
- 3.4 表达式.....38
- 3.5 本章小结.....39
- 3.6 同步练习.....39

第 4 章 运算符.....41

- 4.1 算术运算符.....41
 - 4.1.1 一元运算符.....41
 - 4.1.2 二元运算符.....41
 - 4.1.3 算术赋值运算符.....43
- 4.2 关系运算符.....43
- 4.3 逻辑运算符.....45
- 4.4 位运算符.....46
- 4.5 其他运算符.....47
- 4.6 本章小结.....48
- 4.7 同步练习.....48

第 5 章 Swift 原生数据类型.....50

- 5.1 Swift 数据类型.....50
- 5.2 整型.....50
- 5.3 浮点型.....51
- 5.4 数字表示方式.....52
 - 5.4.1 进制数字表示.....52
 - 5.4.2 指数表示.....52
 - 5.4.3 其他表示.....53
- 5.5 数字类型之间的转换.....53
 - 5.5.1 整型之间的转换.....53
 - 5.5.2 整型与浮点型之间的转换.....54
- 5.6 布尔型.....54

5.7 元组类型	54	7.6.3 guard 中使用 where 语句	86
5.8 可选类型	56	7.7 本章小结	87
5.8.1 可选类型概念	56	7.8 同步练习	87
5.8.2 可选类型值拆包	56		
5.8.3 可选绑定	57	第 8 章 Swift 原生集合类型	91
5.9 本章小结	57	8.1 Swift 中的数组集合	91
5.10 同步练习	57	8.1.1 数组声明和初始化	92
第 6 章 Swift 原生字符串	59	8.1.2 可变数组	93
6.1 字符	59	8.1.3 数组遍历	94
6.1.1 Unicode 编码	59	8.2 Swift 中的字典集合	94
6.1.2 转义符	60	8.2.1 字典声明与初始化	95
6.2 创建字符串	61	8.2.2 可变字典	96
6.3 可变字符串	61	8.2.3 字典遍历	97
6.3.1 字符串拼接	61	8.3 Swift 中的 Set 集合	98
6.3.2 字符串插入、删除和替换	62	8.3.1 Set 声明和初始化	98
6.4 字符串比较	63	8.3.2 可变 Set 集合	99
6.4.1 大小和相等比较	63	8.3.3 Set 集合遍历	100
6.4.2 前缀和后缀比较	64	8.3.4 Set 集合运算	101
6.5 本章小结	65	8.4 本章小结	102
6.6 同步练习	65	8.5 同步练习	102
第 7 章 控制语句	66	第 9 章 函数	104
7.1 分支语句	66	9.1 使用函数	104
7.1.1 if 语句	66	9.2 传递参数	105
7.1.2 switch 语句	68	9.2.1 使用外部参数名	105
7.1.3 guard 语句	69	9.2.2 省略外部参数名	105
7.2 循环语句	71	9.2.3 参数默认值	106
7.2.1 while 语句	71	9.2.4 可变参数	106
7.2.2 repeat-while 语句	73	9.2.5 参数的传递引用	107
7.2.3 for 语句	74	9.3 函数返回值	107
7.2.4 for-in 语句	75	9.3.1 无返回值函数	107
7.3 跳转语句	76	9.3.2 多返回值函数	108
7.3.1 break 语句	76	9.4 函数类型	108
7.3.2 continue 语句	78	9.4.1 作为函数返回类型使用	109
7.3.3 fallthrough 语句	79	9.4.2 作为参数类型使用	110
7.4 范围与区间运算符	81	9.5 嵌套函数	111
7.4.1 switch 中使用区间运算符	81	9.6 本章小结	112
7.4.2 for-in 中使用区间运算符	82	9.7 同步练习	112
7.5 值绑定	83	第 10 章 闭包	115
7.5.1 if 中的值绑定	83	10.1 回顾嵌套函数	115
7.5.2 guard 中的值绑定	84	10.2 闭包的概念	115
7.5.3 switch 中的值绑定	85	10.3 使用闭包表达式	116
7.6 where 语句	85	10.3.1 类型推断简化	117
7.6.1 switch 中使用 where 语句	86	10.3.2 隐藏 return 关键字	117
7.6.2 for-in 中使用 where 语句	86	10.3.3 省略参数名	118

10.3.4 使用闭包返回值	118	12.5.1 下标概念	157
10.4 使用尾随闭包	119	12.5.2 示例：二维数组	157
10.5 捕获上下文中的变量和常量	120	12.6 本章小结	159
10.6 本章小结	121	12.7 同步练习	159
10.7 同步练习	121	第 13 章 方法	160
第 11 章 Swift 语言中的面向对象特性	123	13.1 实例方法	160
11.1 面向对象概念和基本特征	123	13.1.1 使用规范的命名	160
11.2 Swift 中的面向对象类型	123	13.1.2 结构体和枚举方法变异	163
11.3 枚举	124	13.2 静态方法	165
11.3.1 成员值	124	13.2.1 结构体静态方法	165
11.3.2 原始值	126	13.2.2 枚举静态方法	166
11.3.3 相关值	127	13.2.3 类静态方法	166
11.4 结构体与类	128	13.3 本章小结	167
11.4.1 类和结构体定义	128	13.4 同步练习	167
11.4.2 再谈值类型和引用类型	129	第 14 章 构造与析构	168
11.4.3 引用类型的比较	131	14.1 构造函数	168
11.4.4 运算符重载	132	14.1.1 默认构造函数	168
11.5 类型嵌套	133	14.1.2 构造函数与存储属性初始化	169
11.6 可选链	134	14.1.3 使用外部参数名	170
11.6.1 可选链的概念	134	14.2 构造函数重载	172
11.6.2 使用问号 (?) 和感叹号 (!)	136	14.2.1 构造函数重载概念	172
11.7 访问限定	136	14.2.2 构造函数代理	172
11.7.1 访问范围	137	14.2.3 类构造函数横向代理	173
11.7.2 访问级别	138	14.3 析构函数	174
11.7.3 使用访问级别最佳实践	138	14.4 本章小结	176
11.8 选择类还是结构体最佳实践	140	14.5 同步练习	176
11.8.1 类和结构体的异同	141	第 15 章 类继承	178
11.8.2 选择的原则	141	15.1 从一个示例开始	178
11.9 本章小结	142	15.2 构造函数继承	179
11.10 同步练习	142	15.2.1 构造函数调用规则	179
第 12 章 属性与下标	146	15.2.2 构造过程安全检查	180
12.1 存储属性	146	15.2.3 构造函数继承	184
12.1.1 存储属性概念	146	15.3 重写	186
12.1.2 延迟存储属性	147	15.3.1 重写实例属性	186
12.2 计算属性	148	15.3.2 重写静态属性	188
12.2.1 计算属性的概念	148	15.3.3 重写实例方法	189
12.2.2 只读计算属性	150	15.3.4 重写静态方法	190
12.2.3 结构体和枚举中的计算属性	150	15.3.5 下标重写	190
12.3 属性观察者	151	15.3.6 使用 final 关键字	191
12.4 静态属性	153	15.4 类型检查与转换	193
12.4.1 结构体静态属性	154	15.4.1 使用 is 进行类型检查	194
12.4.2 枚举静态属性	155	15.4.2 使用 as、as! 和 as? 进行类型转换	195
12.4.3 类静态属性	156	15.4.3 使用 Any 和 AnyObject 类型	197
12.5 使用下标	156		

15.5 本章小结	198	19.1.1 引用计数	227
15.6 同步练习	198	19.1.2 示例: Swift 自动引用计数	228
第 16 章 扩展	200	19.2 强引用循环	229
16.1 “轻量级”继承机制	200	19.3 打破强引用循环	233
16.2 声明扩展	200	19.3.1 弱引用	233
16.3 扩展计算属性	201	19.3.2 无主引用	236
16.4 扩展方法	202	19.4 闭包中的强引用循环	238
16.5 扩展构造函数	203	19.4.1 一个闭包中的强引用循环示例	238
16.5.1 值类型扩展构造函数	203	19.4.2 解决闭包强引用循环	239
16.5.2 引用类型扩展构造函数	204	19.5 本章小结	241
16.6 扩展下标	204	19.6 同步练习	241
16.7 本章小结	205	第 20 章 错误处理	243
16.8 同步练习	205	20.1 Cocoa 错误处理模式	243
第 17 章 协议	206	20.2 do-try-catch 错误处理模式	244
17.1 协议概念	206	20.2.1 捕获错误	244
17.2 协议	206	20.2.2 错误类型	244
17.3 协议方法	207	20.2.3 声明抛出错误	245
17.3.1 协议实例方法	207	20.2.4 在函数或方法中抛出错误	245
17.3.2 协议静态方法	208	20.2.5 try? 和 try! 的使用区别	246
17.3.3 协议变异方法	208	20.3 示例: MyNotes 应用数据持久层实现	247
17.4 协议属性	210	20.3.1 MyNotes 应用介绍	247
17.4.1 协议实例属性	210	20.3.2 MyNotes 应用数据持久层设计	248
17.4.2 协议静态属性	211	20.3.3 实现 Note 实体类	248
17.5 面向协议编程	211	20.3.4 NoteDAO 代码实现	249
17.5.1 协议类型	212	20.3.5 使用 defer 语句释放资源	250
17.5.2 协议的继承	214	20.3.6 测试示例	251
17.5.3 协议扩展	215	20.4 本章小结	252
17.5.4 协议的合成	216	20.5 同步练习	252
17.5.5 扩展中遵从协议	217	第 21 章 Swift 编码规范	253
17.5.6 面向协议编程示例: 表视图中使用 扩展协议	217	21.1 命名规范	253
17.6 本章小结	220	21.2 注释规范	255
17.7 同步练习	220	21.2.1 文件注释	255
第 18 章 泛型	222	21.2.2 文档注释	255
18.1 一个问题的思考	222	21.2.3 代码注释	256
18.2 泛型函数	222	21.2.4 使用地标注释	257
18.2.1 使用泛型函数	223	21.3 声明	259
18.2.2 多类型参数	223	21.3.1 变量或常量声明	259
18.3 泛型类型	223	21.3.2 属性声明	260
18.4 泛型扩展	225	21.4 代码排版	260
18.5 本章小结	226	21.4.1 空行	260
18.6 同步练习	226	21.4.2 空格	261
第 19 章 Swift 内存管理	227	21.4.3 断行	262
19.1 Swift 内存管理概述	227	21.4.4 缩进	263
		21.5 本章小结	265

第二部分 Cocoa Touch 框架编程篇

第 22 章 Foundation 框架 268

22.1	数字类 NSNumber 268
22.1.1	获得 NSNumber 对象 268
22.1.2	比较 NSNumber 对象 270
22.1.3	数字格式化 271
22.1.4	NSNumber 与 Swift 原生数字类型之间的 桥梁 272
22.2	字符串类 273
22.2.1	NSString 类 273
22.2.2	NSMutableString 类 275
22.2.3	NSString 与 String 之间的桥梁 277
22.3	数组类 278
22.3.1	NSArray 类 278
22.3.2	NSMutableArray 类 279
22.3.3	NSArray 与 Swift 原生数组之间的 桥梁 280
22.4	字典类 281
22.4.1	NSDictionary 类 281
22.4.2	NSMutableDictionary 类 282
22.4.3	NSDictionary 与 Swift 原生字典之间 的桥梁 283
22.5	NSSet 集合类 283
22.5.1	NSSet 类 284
22.5.2	NSMutableSet 类 284
22.5.3	NSSet 与 Swift 原生 Set 之间的 桥梁 285
22.6	文件管理 286
22.6.1	访问目录 286
22.6.2	目录操作 287
22.6.3	文件操作 288
22.7	字节缓存 289
22.7.1	NSData 和 NSMutableData 类 289
22.7.2	示例: Base64 解码与编码 291
22.8	日期与时间 292
22.8.1	使用 NSDate 类 292
22.8.2	日期时间格式化 294
22.8.3	使用 NSCalendar 和 NSDateComponents 类 295
22.8.4	示例: 时区转换 296
22.9	使用谓词 NSPredicate 过滤数据 297
22.9.1	一个过滤员工花名册的示例 298
22.9.2	使用谓词 NSPredicate 300
22.9.3	NSPrdicate 与集合 300

22.9.4	格式说明符 301
22.9.5	运算符 302
22.10	使用正则表达式 304
22.10.1	在 NSPredicate 中使用正则 表达式 305
22.10.2	使用 NSRegularExpression 305
22.10.3	示例: 日期格式转换 307
22.11	本章小结 309

第 23 章 iOS 开发基础 310

23.1	iOS 介绍 310
23.2	第一个 iOS 应用: HelloWorld 310
23.2.1	创建工程 311
23.2.2	Xcode 中的 iOS 工程模板 316
23.2.3	程序剖析 317
23.3	iOS API 简介 319
23.4	本章小结 319

第 24 章 Core Foundation 框架 320

24.1	数据类型映射 320
24.2	内存管理 321
24.2.1	Objective-C 的 MRC 内存管理 321
24.2.2	对象所有权 322
24.2.3	内存托管对象 323
24.2.4	内存非托管对象 324
24.3	本章小结 324

第 25 章 Cocoa Touch 设计模式及应用 325

25.1	单例模式 325
25.1.1	问题提出 325
25.1.2	解决方案 325
25.2	目标与动作 326
25.2.1	问题提出 326
25.2.2	解决方案 327
25.3	输出口 331
25.3.1	问题提出 331
25.3.2	解决方案 331
25.4	选择器 334
25.4.1	问题提出 334
25.4.2	解决方案 334
25.4.3	示例: 使用定时器变换视图背景 335
25.5	委托与数据源 337
25.5.1	问题提出 337
25.5.2	委托解决方案 339
25.5.3	数据源解决方案 343
25.6	通知机制 346

25.6.1 问题提出	346	27.4.3 Objective-C 调用 Swift	399
25.6.2 解决方案	346	27.5 同一框架目标中的混合编程	402
25.6.3 示例: 关闭和打开键盘	347	27.5.1 链接库和框架	402
25.7 MVC 模式	348	27.5.2 Swift 调用 Objective-C	404
25.7.1 MVC 模式概述	348	27.5.3 测试框架目标	407
25.7.2 Cocoa Touch 中的 MVC 模式	349	27.5.4 Objective-C 调用 Swift	410
25.8 响应者链与触摸事件	351	27.6 本章小结	413
25.8.1 响应者对象与响应链	351	第 28 章 Swift 与 C/C++ 混合编程	414
25.8.2 触摸事件	352	28.1 数据类型映射	414
25.9 本章小结	357	28.1.1 C 语言基本数据类型	414
第 26 章 UIKit 框架	358	28.1.2 C 语言指针类型	415
26.1 视图控制器	358	28.2 应用目标中的混合编程	419
26.1.1 与导航相关的视图控制器	358	28.2.1 Swift 调用 C API	419
26.1.2 视图控制器的生命周期	358	28.2.2 Swift 调用 C++ API	421
26.2 视图与 UIView	360	28.3 框架目标中的混合编程	425
26.2.1 UIView 继承层次结构	360	28.3.1 同一框架目标中 Swift 调用 C 或 C++ API	425
26.2.2 视图分类	362	28.3.2 Swift 调用第三方库中的 C 或 C++ API	428
26.2.3 应用界面的构建层次	362	28.4 示例: 用 SQLite 嵌入式数据库实现 MyNotes 数据持久层	430
26.3 界面构建技术	363	28.4.1 Note 实体类代码	430
26.3.1 使用故事板	363	28.4.2 创建表	431
26.3.2 使用 XIB 文件	365	28.4.3 插入数据	432
26.3.3 使用代码	368	28.4.4 查询数据	434
26.4 开关控件	370	28.4.5 应用沙箱目录	435
26.5 滑块控件	371	28.4.6 表示层开发	436
26.6 分段控件	373	28.5 本章小结	440
26.7 工具栏	374	第四部分 进阶篇	
26.8 导航栏	376	第 29 章 程序调试	442
26.9 表视图	379	29.1 Xcode 调试工具	442
26.9.1 表视图组成	379	29.1.1 调试功能	442
26.9.2 单元格组成和样式	380	29.1.2 输出窗口	443
26.9.3 示例: 纯代码创建简单表视图	381	29.1.3 变量查看窗口	444
26.10 本章小结	386	29.2 设置和查看断点	445
第三部分 混合编程篇		29.2.1 断点设置	445
第 27 章 Swift 与 Objective-C 混合编程	388	29.2.2 管理断点	446
27.1 选择语言	388	29.3 使用 Xcode 中的 Instruments 工具	447
27.2 文件扩展名	388	29.3.1 跟踪内存使用	447
27.3 Swift 与 Objective-C API 映射	388	29.3.2 内存泄漏查找	449
27.3.1 构造函数映射	389	29.4 iOS 真机调试	453
27.3.2 方法名映射	391	29.4.1 Xcode 设置	453
27.4 同一应用目标中的混合编程	393		
27.4.1 什么是目标	394		
27.4.2 Swift 调用 Objective-C	396		

29.4.2 设备设置	455
29.5 本章小结	456
第 30 章 并发编程	457
30.1 并发相关概念	457
30.1.1 CPU 与并发执行	457
30.1.2 进程与线程	457
30.2 GCD 技术	457
30.2.1 GCD 概念	458
30.2.2 调度队列	458
30.3 GCD 示例	458
30.3.1 主线程阻塞问题	458
30.3.2 使用 GCD 解决主线程阻塞	460
30.4 GCD 与单例模式	460
30.5 本章小结	461
第 31 章 SpriteKit 游戏引擎	462
31.1 移动平台游戏引擎介绍	462
31.2 第一个 SpriteKit 游戏	462
31.2.1 创建工程	462
31.2.2 工程剖析	465
31.3 一切都是节点	468
31.3.1 节点“家族”	469
31.3.2 节点树	469
31.3.3 节点中重要的方法	470
31.3.4 节点中重要的属性	470
31.4 精灵	471
31.4.1 精灵类 SKSpriteNode	471
31.4.2 示例：沙漠英雄场景	473
31.4.3 使用纹理图集性能优化	478
31.5 场景切换	481
31.5.1 场景切换方法	481
31.5.2 场景过渡动画	481
31.5.3 示例：沙漠英雄场景切换	481
31.6 动作	484
31.6.1 常用动作	484
31.6.2 组合动作	487
31.6.3 示例：帧动画实现	491
31.7 粒子系统	493
31.7.1 粒子系统属性	494
31.7.2 内置粒子系统模板	495
31.8 游戏音乐与音效	499
31.8.1 音频文件介绍	499
31.8.2 iOS 和 OS X 平台音频优化	499
31.8.3 背景音乐	500
31.8.4 3D 音效	501

31.9 物理引擎	502
31.9.1 物理引擎核心概念	502
31.9.2 物理引擎中的物体	503
31.9.3 接触与碰撞	504
31.10 本章小结	507

第五部分 项目实战篇

第 32 章 项目实战：基于分层架构的多版本 iPhone 计算器应用	510
32.1 应用分析与设计	510
32.1.1 应用概述	510
32.1.2 需求分析	510
32.1.3 原型设计	511
32.2 分层架构设计	511
32.2.1 低耦合应用架构	511
32.2.2 详细设计	512
32.2.3 分层架构设计有多种模式	514
32.2.4 选择语言	514
32.3 基于同一工程同一目标纯 Swift 的实现	515
32.3.1 创建工程	515
32.3.2 业务逻辑层开发	516
32.3.3 表示层开发	520
32.4 基于同一工程同一目标 Swift 调用 Objective-C 的实现	531
32.4.1 在 Swift 工程中添加 Objective-C 类	531
32.4.2 Objective-C 实现业务逻辑层	532
32.5 基于同一工程不同目标 Swift 调用 Objective-C 的实现	534
32.5.1 添加业务逻辑层目标	535
32.5.2 暴露的头文件 CalcLogic.h	535
32.5.3 修改表示层	536
32.6 基于同一工作空间不同工程下 Swift 调用 Objective-C 的实现	536
32.6.1 创建工作空间	537
32.6.2 暴露头文件 CalcLogic.h	537
32.6.3 修改表示层	537
32.7 本章小结	537
第 33 章 游戏 App 实战：《迷失航线》	538
33.1 《迷失航线》游戏分析与设计	538
33.1.1 《迷失航线》故事背景	538
33.1.2 需求分析	538
33.1.3 原型设计	539
33.1.4 游戏脚本	540

33.2	任务 1: 游戏工程的创建与初始化	540	33.8	任务 7: 游戏结束场景	573
33.2.1	迭代 1.1: 创建工程	540	33.8.1	迭代 7.1: 设计场景	573
33.2.2	迭代 1.2: 自定义类型维护	540	33.8.2	迭代 7.2: 实现代码	574
33.2.3	迭代 1.3: 添加资源文件	543	33.9	本章小结	575
33.3	任务 2: 创建 Loading 场景	543	第 34 章 《迷失航线》游戏上架苹果 App Store		576
33.3.1	迭代 2.1: 设计场景	544	34.1	苹果 App Store	576
33.3.2	迭代 2.2: Loading 动画	545	34.2	还有“最后一公里”	577
33.3.3	迭代 2.3: 预处理加载纹理	545	34.2.1	添加图标	577
33.4	任务 3: 创建 Home 场景	546	34.2.2	添加启动界面	579
33.4.1	迭代 3.1: 设计场景	546	34.2.3	修改发布产品属性	581
33.4.2	迭代 3.2: 实现代码	547	34.3	为发布进行编译	582
33.5	任务 4: 创建设置场景	549	34.3.1	创建开发者证书	583
33.5.1	迭代 4.1: 设计场景	550	34.3.2	创建 App ID	587
33.5.2	迭代 4.2: 实现代码	550	34.3.3	创建描述文件	589
33.6	任务 5: 创建帮助场景	552	34.3.4	发布编译	593
33.6.1	迭代 5.1: 设计场景	553	34.4	发布上架	594
33.6.2	迭代 5.2: 实现代码	553	34.4.1	创建应用	595
33.7	任务 6: 实现游戏场景	554	34.4.2	应用定价	597
33.7.1	迭代 6.1: 设计场景	555	34.4.3	基本信息输入	597
33.7.2	迭代 6.2: 创建敌人精灵	556	34.4.4	上传应用	601
33.7.3	迭代 6.3: 创建玩家飞机精灵	560	34.4.5	提交审核	603
33.7.4	迭代 6.4: 创建子弹精灵	560	34.5	审核不通过的常见原因	604
33.7.5	迭代 6.5: 初始化游戏场景	561	34.5.1	功能问题	604
33.7.6	迭代 6.6: 玩家移动飞机	565	34.5.2	用户界面问题	604
33.7.7	迭代 6.7: 游戏循环与任务调度	566	34.5.3	商业问题	605
33.7.8	迭代 6.8: 游戏场景菜单实现	566	34.5.4	不当内容	605
33.7.9	迭代 6.9: 玩家飞机发射子弹	568	34.5.5	其他问题	605
33.7.10	迭代 6.10: 子弹与敌人的碰撞检测	569	34.6	本章小结	605
33.7.11	迭代 6.11: 玩家飞机与敌人的碰撞检测	571			

Part 1

第一部分

Swift 语法篇

本 部 分 内 容

- 第 1 章 准备起航
- 第 2 章 第一个 Swift 程序
- 第 3 章 Swift 语法基础
- 第 4 章 运算符
- 第 5 章 Swift 原生数据类型
- 第 6 章 Swift 原生字符串
- 第 7 章 控制语句
- 第 8 章 Swift 原生集合类型
- 第 9 章 函数
- 第 10 章 闭包
- 第 11 章 Swift 语言中的面向对象特性
- 第 12 章 属性与下标
- 第 13 章 方法
- 第 14 章 构造与析构
- 第 15 章 类继承
- 第 16 章 扩展
- 第 17 章 协议
- 第 18 章 泛型
- 第 19 章 Swift 内存管理
- 第 20 章 错误处理
- 第 21 章 Swift 编码规范

当你拿到本书的时候，我相信你已经下定决心学习Swift语言了。那么应该怎么开始呢？这一章我们不讨论技术，而是告诉大家本书中的一些约定，并介绍开发工具等内容。

1.1 本书约定

为了方便大家阅读，本节会介绍一下书中示例代码和图示的相关约定。

1.1.1 示例代码约定

本书包含大量示例代码，我们可以从图灵网站iTuring.cn本书主页免费注册下载，或者从智捷教育提供的本书服务网站（www.51work6.com）下载，解压后可看到如图1-1所示的目录结构。

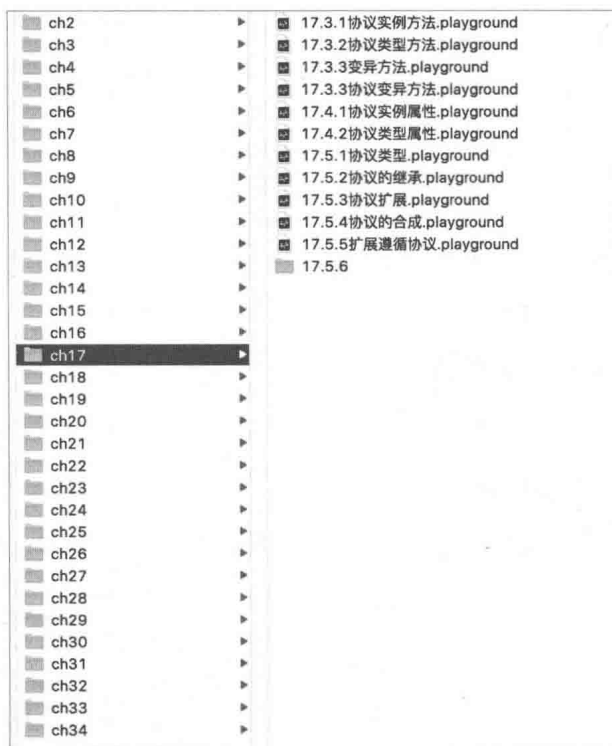


图1-1 源代码文件目录

ch2~ch34代表第2章到第34章的示例代码或一些资源文件，其中Playground文件、工程或工作空间的命名有如下几种形式。

- 二级目录标号，如“13.1实例方法”说明是13.1节中使用的“实例方法” Playground文件或工程或工作空间示例。图标为的是Playground文件，图标为的是Xcode工程，图标为的是Xcode工作空间。
- 三级目录标号，如“13.1.2结构体和枚举方法变异”说明是13.1.2节中使用的“结构体和枚举方法变异”。带有“~”的情况，如“29.1~29.2 SQLiteDBApp”说明是29.1节到29.2节共同使用的SQLiteDBApp的 Playground文件、工程或工作空间示例。
- ch33和ch34目录下只有一个LostRoutes目录，没有标号，说明该章中都在使用LostRoutes这个示例。
- 资源目录，是保存了本章中用到的一些资源文件，如：图片、声音和配置文件等。

1.1.2 图示约定

为了更形象有效地说明知识点或描述操作，本书添加了很多图示，下面简要说明图示中一些符号的含义。

- 图中的圆框。有时读者会看到图1-2中所示的圆框，其中是选中的或重点说明的内容。

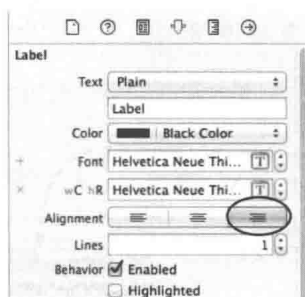


图1-2 图中的圆框

- 图中的箭头。如图1-3所示，实线箭头用于说明用户的动作，一般箭尾是动作开始的地方，箭头指向动作结束的地方。图1-4所示的虚线箭头在书中用得比较多，常用来描述设置控件的属性等操作。

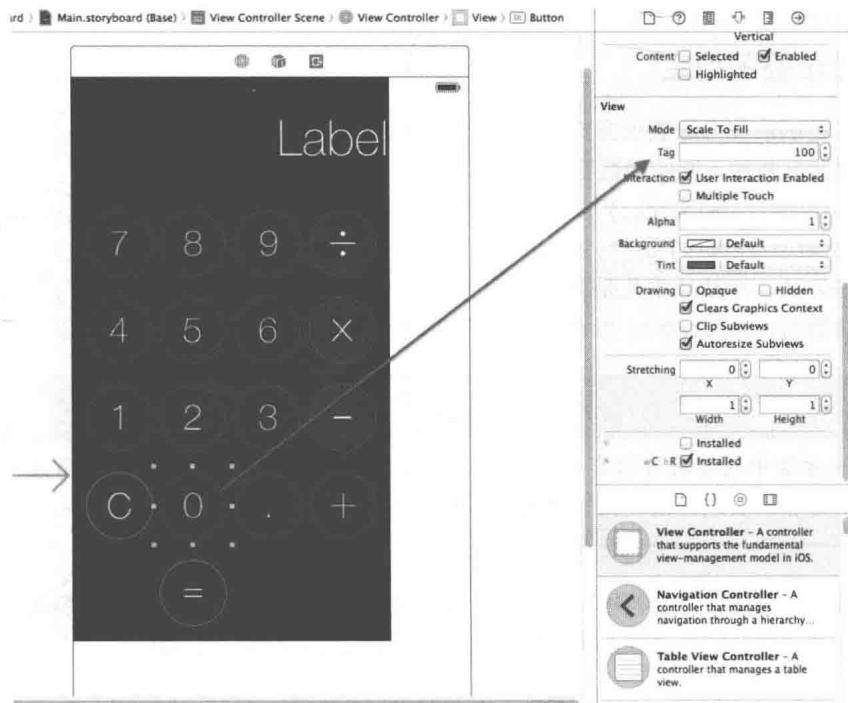


图1-3 图中实线箭头