

老码说编程

玩转



基于XCode 6.1

详解用Swift开发App全过程

老码用故事教你学iOS开发

全新技术，全新讲法

Swift

江湖

老码团队 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

老码说编程②

玩转



Swift江湖

老码团队 王芳杰 安秋雨 秦平雷 李捷 冉静 著

电子工业出版社

Publishing House of Electronics Industry

内 容 简 介

本书由老码团队的 5 位资深 IT 工程师钻研 4 个月学习整理和提炼而成, 全书分为 10 个章节: 第 1~2 章讲述如何搭建开发环境, 讲解 XCode 6.1 的用法; 第 3~6 章重点讲述 Swift 的语法知识; 第 7 章讲解 Swift 和 Objective-C 之间的相互调用; 最后 3 章通过一个完整的案例“魔法盒子”来讲解如何使用 Swift 语言从零开始开发一款商用的 App。本书在知识布局上不仅有对 Swift 基础语法的讲解, 同时也对 iOS 开发过程中的众多关键技术节点进行阐述, 如库管理工具 Cocoapods、原型设计工具 Axure Pro、云端测试工具、Cordova/Phonegap HTML5 HyberApp 的入门知识、真机调试、打包上传等。我们旨在通过本书给读者朋友们绘制一条我们走过且读者朋友未来要走的路。本书被 Swift 社区的读者们誉为“Swift 世界的地图”。

为了保证 Swift 语言的稳定和优良, 苹果公司先后为 XCode 6 推出了 7 个 Beta 版本, 一个 GM (Golden Master) 版本, 最后稳定于正式版 XCode 6.1, 本书是基于当前最新正式版编写的。老码团队在编写本书的同时, 也在持续维护翻译《The Swift Programming Language》中文版。

本书普遍采用诙谐、幽默、轻松的语言, 力图为读者们创造一个轻松的学习氛围, 书中的案例紧贴社会热点, 比如老码去蓝翔、铡美案、韩剧《继承者们》、江湖武侠、杨过、小龙女、甚至网络红人凤姐、芙蓉等都会相继出场, 使大家在嬉笑怒骂间领悟 Swift 语言精髓和用法。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

老码说编程之玩转 Swift 江湖 / 老码团队著. —北京: 电子工业出版社, 2015.1
ISBN 978-7-121-24741-5



I. ①老… II. ①老… III. ①程序语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2014) 第 262311 号

责任编辑: 董 英

印 刷: 北京京科印刷有限公司

装 订: 三河市皇庄路通装订厂

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: 31

字数: 714 千字

彩插: 1

版 次: 2015 年 1 月第 1 版

印 次: 2015 年 1 月第 1 次印刷

印 数: 3000 册 定价: 79.00 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。

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。

服务热线: (010) 88258888。

前言

Hi, 亲爱的小伙伴们:

你们好!

我是老码团队带头大哥 Tyrion, 江湖人称“小恶魔”, 不管您已是我们的读者, 还是即将成为我们的读者, 又或者是路过的驻留, Tyrion 都代表老码团队全体老码农们对大家表示深深的感谢, 正因为你们的关注才使我们有了前进的动力。

本书结构背后的故事

当老码团队把本书结构图发布出去倾听各方意见的时候, 当我们做第二轮开放式 Review 的时候, 我们收到了各种各样的建议, 总结一下分为以下三类。

- 第一类: 掐头去尾, 保留 Swift 的章节即可。
- 第二类: 删除书里 Cocoapods、Axure 及真机调试部分。
- 第三类: 全部保留, 越多越好。

前两类的呼声均来自有数年开发经验的朋友们, 而第三类则来自大多数刚入门或者寻求入门的小伙伴们那里。在万难的境地, 老码团队的小伙伴们多次讨论, 最终投票决定留下, 为什么留下? 因为我们都沉迷于一个属于老码的梦。

老码的梦里, 我们这群老码农不再是工作 10 年左右的 C++ 或 Java 开发者, 我们不再是什么外企的技术经理, 我们是一群站在 iOS 移动开发大门口, 睁着迷茫的双眼, 寻找入门契机的小白, 或者说是学生, 又或者是其他行业打算投入 iOS 开发领域的小朋友。最终我们开始了学习, 首先迷茫的是, 我们想知道怎么开始, 该用 Windows 还是 Mac, 买不起 Mac 怎么办? 据说虚拟机可以。有了环境, 我们终于可以在 XCode 的世界里写几行 Swift 版的 Helloworld 了, 于是我们苦苦地看语法, 发现 Swift 的语法远没有外界传说得那么简单, 所以我们需要以幽

默、风趣、直接的语气来学习。最后，辛苦地学完 Swift 语法，我们终于可以跃跃欲试地开发一款 iPhone 上的 App 了，感觉还不错，但是 Cocoa 的库都用 Objective-C，于是我们想知道 Swift 如何与 Objective-C 互通，最后终于明白了，原来 Swift 可以调用任意的 Objective-C 函数。此时我们感觉自己的 App 还不错，于是我们想放到真机上调试，怎么放？如何申请开发者账号、如何申请证书、如何申请标识文件等问题统统蹦了出来，我们需要这些！当一切搞好，我们还想知道上传 AppStore 的过程。当自学的小应用变成一个商用的 App 时，一切变得复杂了，我们需要知道规划创意，创意有了，原型是必需的，据说有 Axure，我们想知道！当然一款好的 App 避免不了数据更新，所以我们让软件的拓扑从单机变成了服务器/客户端模式，那么我们的 App 如何和服务端通信？因此作为小白的我们应该知道什么是 HTML+JSON 通信。当然这一切都不需要我们开发，已经有一群大牛帮我们写好了工具，怎么用？Cocoapods，我们要知道啊，当 Swift 技术学到一定程度，我们自然需要想着怎么可以在 iOS 开发的行业有所作为，于是我们想知道如何赚钱，有哪些商业模式。其实走 iOS 的路不仅仅是 Swift+Objective-C+Cocoa Touch，我们还可以选择 Web App 的模式，Phonegap&Cordova，作为小白的我们应该知道。

正因为有这个梦存在，老码团队的小伙伴们坚持留下被人诟病的部分，其实 10 年前的老码也曾站在 C++ 的大门口，也曾站在 Java 的大门口，茫然不知所措，正如亲爱的小伙伴们站在 iOS 开发的大门口一样。所以老码想用这本书给小伙伴们展现一幅步入 iOS 移动开发殿堂的地图。这张地图里，你会看到有哪些路要走，路上有哪些路标，至于每个路标，每个节点，肯定有独特的风景，小伙伴们则需要感兴趣的点驻留研究，览尽无限风情。正如原型设计是 iOS 个体开发到商业化开发过程中一个必需的路标，老码有责任让读者知道这个路标的存在，但是到了这个路标，小伙伴们打算驻留还是匆匆离开则随自便。

所以老码再次强调：本书的目标读者是学生、Swift 入门者、移动开发小白和跨行业 0 基础者，如果小伙伴们有一定的 iOS 开发经验，这本书可能不适合你。本书是老码团队成员学习 Swift 的笔记整理，无法覆盖 Swift 语言的每个细节，所以朋友们可以参考阅读老码团队负责翻译维护的《The Swift Programming Language 中文版》在线文档¹。朋友们相互对照参考学习，将会收获颇丰。

此致

敬礼

王芳杰 (Tyron)

¹ 该文档由老码团队翻译组成员 Bran（北航大学生梁杰）发起，曾因 9 天翻译完成而闻名业内，该文档是目前市面上所有 Swift 中文教程的母本，地址请咨询@未来眼之老码团队官方微博。

关于作者

老码团队是 30 岁以上老码农的聚集地，我们的团队有 5 位固定成员，也是本书的 5 位作者，因为我们都是美剧《冰与火之歌》的粉丝，所以我们的网名都来自剧情里面，以下是几位作者的感言。

“@Tyrion 让我用 200 字来对买这本书的朋友们说说关于我自己的事情，200 真的挺少的，我是个话唠来着，200 是绝对不够用的啊，而且我现在还浪费掉了这么多，不能再浪费下去了，我要进入正题了。正题就是我是个非常厉害的幸运儿，因为在我目前为止经历的生活里，我拥有着亲情、爱情和友情。论亲情，亲爱的爸妈，你们对我绝对是真爱，因为你们居然打算买我的这本书，是要学 iOS 开发吗？论爱情，亲爱的，每次我以写书为借口逃避带孩子实际躲在书房看片的时候，你总能及时地发现，你是爱我的啊！论友情，‘老码团队’、‘贵妇团’，当我要求你们捧场买我这本书，你们告诉我‘滚，家里没地方放’的时候，我明白你们是不可多得的朋友啊！最后，我美丽的女儿，由由，妈妈想把这本书送给你当玩具！”

——冉静（网名 Snow）

“从开始写书的那一刻起，几乎所有的业余时间都投入进来，查阅资料，动手实验，总结知识点，精炼语句……一切都在紧张有序地展开，时间的概念越来越模糊。转眼间 iOS8 正式发布，我们的 Review 工作也夜以继日进行着。昨天突然接到 Tyrion 的消息，说要为书写一段感言，这才意识到原来我们的工作已经到了收尾阶段，我们的书很快会和小伙伴们见面了。同时，我们家的小宝宝也在他妈妈的肚子里顽皮地蹦跶了 28 周，他也很快会和我见面了。无论是书还是我们家的小宝宝，这段孕育的过程都是我这一生珍贵的回忆。感谢 Tyrion、Snow、Baelish、Oberyn 的陪伴，是你们的专业态度和为梦想奋斗的精神让我们能一路走来披荆斩棘。感谢我的老婆 Annie，还有什么能比一个坚强独立的准妈妈更值得尊敬的呢？最后要感谢给予老码掌声和批评的各位前辈和小伙伴们，你们是老码一直以来的前进动力！老码的故事由你们来撰写，在这里我要说：‘未完，待续！’”

——安秋雨（网名 Robb）

“加入老码团队时，已经年过 30，是一位名副其实的老码农。有七八年的 C 和 C++ 开发经验，对 C 和 C++ 非常熟悉，套用一句主持人常说的话说，就是知道 C 和 C++ 中的每一个“梗”在哪里。但由于年纪的原因，记忆力衰退，有时具体的细节还是会记不住，需要借助网络的力量来查询。所以比新人唯一的优势就是经验吧，笑！之所以会写这本书，有两个原因：一是觉得既然在学习 Swift 过程中写了很多笔记，那为什么不能把它整理成册；二是想把学习 Swift 过程中所遇到问题都记录下来跟天下码农一起分享。当然最终这本书能顺利完成并出版，也得感谢整个老码团队所有组员之间的相互理解、支持和付出，以及家人的体谅，没有大家的共同努力也就没有这本书的最终发售，在这里想再次对大家说声由衷的谢谢！”

——秦平雷（网名 Baelish）

“在那个毫无生气的初夏，我离开了项目，辞掉了多年的工作。待业，寻途中，反观这几年来的通信行业和 C/C++ 编程，我更确定它们早已不是时代的宠儿，虽然还有自己一亩七分的自留田，但却在互联网、移动互联网的滚滚浪潮面前黯淡无光，跌入尘埃。各种新兴的拉勾们，两眼烁烁紧盯互联网，哪怕传统的 51 上也见不到许多 C++ 工程师的岗位。我寻求变更，学习 Java，期望能在半身已掩埋在棺材底下的时候，能够奋力爬出跟上时代的步伐。但直面的残酷是非应届毕业生，无相关项目经验根本不进互联网巨头的法眼，哪怕崛起在新世界中的各种创业公司，也看不起我们这些来自大航海世界前半程的老海盗们。就这般苟延残喘，继续在温水煮青蛙的环境中腾挪，续命？在那段日子里，未来是越来越黑暗的。

然而，希望的火苗，总是孕育在黑暗之中，看似不可能的地方才会有奇迹的诞生。

那个下午，我在 Snow 的微博里，看到了 Tyrion 那篇充满激情，极其有趣，又孕育希望的‘战斗檄文’，一群三十多岁的老屌丝，老码农，不为金钱，只为做些有趣的事，投身开源；无所畏惧，即便在而立之年，毅然创业。在这些看似‘荒诞’的事情里面，希望在慢慢萌芽。而恰逢 Swift 的推出，它的推陈出新，它的简单易懂，它的便捷快速，它的安全健壮，把编程变成了一种乐趣，我隐约看到了一个新的时代即将到来，决然地，我拨起了 Tyrion 的电话，加入了老码团队。

这段日子，是莫可名状的。从一开始的新鲜，激情，再到慢慢消化新东西带来的巨大信息量和完成逐渐庞大的任务分配表所带来的疲惫。我身边的这群老码们是专业的，值得尊敬的，大家都还在岗，完成了一天的工作，下了班回家已一身疲惫。同时，随着 Robb 的妻子腹中怀子，我们已成了一个名副其实的爸爸妈妈创业团，每个人回家还得承担家务，照顾宝宝。但就是这样的情况下，各位还是毫无怨言地继续奉献开源，为完成这本书而奋斗。我想，能让我们坚持下来的，如若不是对未来的美好希冀，还能是什么呢？

在这里，必然的，我得感谢我的家人，是他们的理解和支持，让我得以花费这么多时间在这项事业上。同时，也得感谢 Tyrion，是他饱满的激情，高效的执行力，远瞩的目光，让这本书的出版得以促成。这里，是本章的终结，但却也是新篇章的开始，在这大航海时代的下半段，我们在新世界里扬帆起航。”

——李捷（网名 Oberyin）

“在逝去的 4 个月内，我们追逐着每一次 XCode 6 的更新，我们因每一次 Swift 语言的变化而增删章节，我们曾经迷茫，疲惫，但是我们坚持了下来。感谢老码团队的兄弟们，感谢老码团队翻译组美女赵珏映（小 K）、梁任冬、梁杰，也由衷地感谢参与我们的开放式 Review 的几位 iOS 资深工程师：付宇轩 (@Dev Talking)、吴建军、何林、吴跃刚。感谢老码团队优秀的 UI 设计专家李敏，同时也感谢《Swift 兴趣交流二群》的 980 位朋友们，特别是“小波说雨燕”的夜会美、Solomon（马宏亮）、Geeks（袁超）、Chenhshin_Wang（王宸旻）、蜗牛君（李艺），还有 Vampire（郑蔚）、卑鄙三先生（唐三龙）。因为有你们的建议和支持，当我们疲倦打算敷衍收笔时心里便会有莫名的羞愧，于是会再 Review 一遍，坚持直至今日。最后请允许 Tyrion 代表所有作者对陪伴我们走过整个出版过程的美女编辑董英还有专业的美工设计师李玲以及其他各位为本书上市做出不懈努力的电子工业出版社的朋友们，你们的专业、热情、耐心是我们巨大的动力，感谢你们！”

作为带头大哥的我，要感谢陪我走过每一个写书时光的 Robb、Baelish、Snow 和 Oberyin，也要深深地感谢我的老大叠拓 NGN 中国区总经理许可先生，以及那帮可爱的同事们，感谢你们的信任和支持。必须感谢我那美丽的女儿，每次写至深夜，疲惫时，看到你笨笨地喊爸爸的样子，我会想象如果有一天你会指着这本书告诉你的小伙伴们说这是我爸爸写的那骄傲表情。最后感谢我的妻子，感谢你无怨无悔的付出，虽然每天 12 个小时工作，我经常忽略你的存在，但是桌上那杯永远的热茶给了我无限的动力，当收笔，摘下眼镜，看着窗外的成都，我坚决地认为这一切都是值得的。

虽然近 30 万字，我们依然无法做到详细、细致地写完读者应该知道的知识点，因此老码团队未来会对重点章节和后面的案例部分制作视频，免费给各位小伙伴们以帮助理解，也算是对读者未来产生的不确定失望一点点的补偿。技术是很严肃的东西，如果小伙伴们发现本书有任何的不解和错误，可以在新浪微博中@未来眼之老码团队找老码寻求帮助。我们每天都有很多工作要做，但是你们的问题我们都会记录，然后找时间学习，研究，并一一作答！”

——王芳杰（Tyrion）

目 录

第 1 章 从零搭建 Swift 开发环境..... 1

- 1.1 虚拟机安装 Mac OS 1
 - 1.1.1 准备工作.....1
 - 1.1.2 动手吧.....2
- 1.2 X86 电脑安装 Mac OS..... 33
 - 1.2.1 安装方式简介.....33
 - 1.2.2 规划分区.....34
 - 1.2.3 使用硬盘助手 Leopard.....37
 - 1.2.4 使用变色龙 Chameleon.....38
 - 1.2.5 使用 HFS+.....39
 - 1.2.6 正式安装.....41
- 1.3 结语..... 42

第 2 章 玩转 XCode..... 43

- 2.1 下载与安装 XCode..... 43
 - 2.1.1 下载 XCode43
 - 2.1.2 安装 XCode43
- 2.2 XCode 之初体验..... 44
 - 2.2.1 欢迎界面.....44
 - 2.2.2 第一个 Swift 程序46
 - 2.2.3 初识主界面.....50
 - 2.2.4 工具栏.....50
 - 2.2.5 导航区域.....51
 - 2.2.6 编辑区域.....57
 - 2.2.7 调试区域.....58

- 2.2.8 组件区域.....60

2.3 XCode 的调试大法 60

- 2.3.1 简单粗暴的 NSLog60
- 2.3.2 断点调试之 LLDB.....64
- 2.4 老码的锦囊——帮助文档 69
- 2.5 结语..... 71

第 3 章 Swift 语法那些事儿..... 72

- 3.1 Swift 的碑文 72
- 3.2 常量和变量..... 73
 - 3.2.1 常量和变量的命名.....74
 - 3.2.2 打印常量和变量.....75
- 3.3 类型标注 76
- 3.4 注释..... 77
 - 3.4.1 单行注释.....77
 - 3.4.2 多行注释.....77
- 3.5 分号的落寞..... 77
- 3.6 语法届的直性子——整型 78
- 3.7 语法届学术范——浮点型..... 79
- 3.8 类型安全和类型推导..... 79
- 3.9 数字字面量..... 81
- 3.10 数字类型的转换..... 83
 - 3.10.1 整型间的转换.....83
 - 3.10.2 整型和浮点型间的转换84
- 3.11 类型别名 85
- 3.12 布尔型..... 85

3.13 字符串和字符.....	86
3.13.1 字符串字面量.....	87
3.13.2 字符串的创建和初始化.....	88
3.13.3 字符串和字符的连接.....	89
3.13.4 字符计数.....	90
3.13.5 字符串内插.....	91
3.13.6 比较字符串.....	92
3.14 元组.....	94
3.15 可选类型.....	96
3.15.1 If 语句和强制解封可选 类型.....	97
3.15.2 可选绑定.....	97
3.15.3 隐式解封可选类型.....	98
3.16 断言.....	99
3.17 集合.....	100
3.17.1 集合的可变性.....	100
3.17.2 数组的类型.....	100
3.17.3 数组字面量.....	101
3.17.4 访问和修改数组.....	103
3.17.5 数组遍历.....	105
3.17.6 数组创建和初始化.....	106
3.18 字典.....	107
3.18.1 字典字面量.....	108
3.18.2 访问和修改字典.....	109
3.18.3 字典遍历.....	111
3.18.4 创建空字典.....	112
3.19 结构体类型.....	112
3.19.1 基础数据类型都是结构体.....	112
3.19.2 结构体的定义.....	113
3.20 枚举类型.....	114
3.20.1 枚举的定义.....	115
3.20.2 通过 Switch 语句匹配枚 举的值.....	116
3.20.3 关联值.....	117
3.20.4 原始值.....	118
3.21 结语.....	119

第 4 章 离不开的运算符，丢不下的 流程控.....	120
4.1 运算符的家谱.....	120
4.2 赋值运算符.....	122
4.3 算术运算符.....	124
4.3.1 加减乘除.....	124
4.3.2 求余运算符.....	128
4.3.3 自变运算符.....	129
4.3.4 阴阳运算符.....	131
4.4 区间运算符.....	132
4.5 逻辑运算符.....	133
4.5.1 9 种逻辑运算符综合演练.....	134
4.5.2 三目运算符(?:).....	135
4.5.3 空合运算符(??).....	136
4.5.4 特征等于运算符(==)和 特征不等运算符(!=).....	137
4.6 位运算符.....	137
4.6.1 位求非(~).....	137
4.6.2 位求与(&).....	139
4.6.3 位求或().....	140
4.6.4 位求异或(^).....	141
4.6.5 移位操作.....	142
4.7 运算符专题.....	144
4.7.1 运算符里的江湖等级.....	144
4.7.2 圆括号——打破江湖规矩 的利器.....	147
4.7.3 定制自己的运算符套餐.....	147
4.8 语法界的建筑师.....	153
4.8.1 for 语句.....	153
4.8.2 while 循环.....	156
4.8.3 if 判断.....	158
4.8.4 switch 分拣.....	158
4.8.5 switch 中的 case 专题.....	159
4.9 打破控制流程的壁垒.....	164
4.9.1 continue 的用法.....	164
4.9.2 break 的用法.....	165

4.9.3 return 的用法	166	第 6 章 Swift 之面向对象	196
4.9.4 fallthrough 的用法	166	6.1 面向对象的身世之谜	196
4.10 结语	167	6.2 Swift 中的 OOP	199
第 5 章 语法届的高富帅之函数和闭包	168	6.2.1 何为类	199
5.1 写个函数来瞧瞧	168	6.2.2 何为实例	200
5.2 函数怎么用	170	6.2.3 何为属性	201
5.3 参数列表的奥妙	170	6.2.4 何为方法	212
5.3.1 多参数的参数列表	170	6.2.5 继承者们	218
5.3.2 没有参数怎么办	171	6.2.6 让我们构造一个类的实 例吧	223
5.3.3 谁说参数不能有名字	171	6.2.7 析构——回光返照时的 精彩	236
5.3.4 酷炫的可变参数	173	6.3 一根藤上三朵花：类、结构体和 枚举	238
5.3.5 默认参数的魅力	174	6.4 Swift 内存总管——自动引用计数 (ARC)	243
5.3.6 当默认参数遇上可变参数 ...	176	6.4.1 强引用——霸道的关系	243
5.3.7 参数性别是常量还是变量 ...	177	6.4.2 闭环与解环	246
5.3.8 让我变，不管函数内外	178	6.4.3 解环妙法之弱引用	249
5.4 仔细分析一下返回值的各种情况 ...	179	6.4.4 解环妙法之无主引用	250
5.4.1 没返回值也行	179	6.4.5 解环妙法之捕获列表	252
5.4.2 一个返回值没问题	180	6.5 玩转 Swift 的各种类型	257
5.4.3 多返回值也可以	180	6.5.1 可选链——用前就知道 是不是 nil 哦	257
5.5 挖挖函数的老底	181	6.5.2 类型检查和转换	266
5.6 函数的高级玩法	182	6.5.3 俄罗斯娃娃——类型里面 还有类型	275
5.6.1 函数类型的变量与常量	182	6.5.4 我们都是 Swift 的主人—— 扩展	278
5.6.2 函数类型作为参数	183	6.5.5 用需求规划蓝图——协议 ...	287
5.6.3 函数类型作为返回值	184	6.6 通用是一种美德——泛型	307
5.7 嵌套函数	185	6.6.1 泛型函数	309
5.8 柯里化函数	185	6.6.2 通用类型	310
5.9 闭包——牛人的玩物	189	6.6.3 类型约束	312
5.10 闭包表达式很好用	189	6.6.4 关联类型	315
5.11 简化闭包表达式	191	6.7 结语	318
5.12 尾闭包	192		
5.13 捕获值	193		
5.14 挖挖闭包的老底	195		
5.15 结语	195		

第 7 章 Swift 与 Cocoa 框架	319	8.4.3 沙盒的操作	405
7.1 Swift 和 Objective-C 的互用	319	8.5 结语	408
7.1.1 将 Objective-C 导入 Swift	319	第 9 章 Swift 商用版魔法盒	409
7.1.2 将 Swift 导入 Objective-C	324	9.1 确定商用版魔法盒架构	409
7.1.3 互用性浅析	326	9.1.1 功能化确认	409
7.2 Cocoa 框架	328	9.1.2 客户端 UI 级架构	410
7.2.1 Cocoa 的前世今生	329	9.1.3 系统部署架构	410
7.2.2 CocoaTouch 的世界	330	9.2 认识 UIKit	411
7.2.3 基础类库之 Foundation 框架	331	9.3 编码之主框架搭建	413
7.2.4 用户界面之 UIKit/AppKit	340	9.3.1 部署文件目录	413
7.2.5 Core Data	340	9.3.2 刷新项目目录	415
7.2.6 使用 Cocoa 设计模式	350	9.3.3 添加 Cocoapods 库	415
7.3 Cocoapods——开源界的利器	359	9.3.4 桥文件中增加库引用	417
7.3.1 为什么需要 Cocoapods	359	9.3.5 实现主视图和主视图控 制器	418
7.3.2 什么是 Cocoapods	359	9.3.6 增加子页面视图 (View)	420
7.3.3 安装 Cocoapods	359	9.3.7 增加主页面控制器 (Controller)	423
7.3.4 Cocoapods 环境设置	361	9.3.8 建立控制器和视图间的 连接	425
7.3.5 构建 Cocoapods 项目	362	9.3.9 主视图控制器把子页面组 织起来	426
7.4 结语	371	9.4 编码之实现 BoxPlaza 页面	432
第 8 章 踏上 Swift 开发 iOS 之旅	372	9.4.1 准备供测试的图片	432
8.1 无规划, 不产品	372	9.4.2 准备配置文件	433
8.2 伟大的建筑从草图开始	373	9.4.3 实现配置文件解析	435
8.2.1 原型工具万花筒	373	9.4.4 为 Box App 添加 Model	438
8.2.2 打造“魔法盒子”的原型	374	9.4.5 实现 Collection View	440
8.3 动工吧	379	9.4.6 Autolayout 的使用	442
8.3.1 认识我们的 Magic Box 项目	380	9.4.7 设计 Collection View Cell	446
8.3.2 五个文件——梦想起飞 的地方	381	9.4.8 添加代码	447
8.3.3 Baby 版魔法盒诞生	390	9.5 编码之实现 MyBox 页面	449
8.4 神马是沙盒	401	9.5.1 设计 MyBox UI	449
8.4.1 沙盒原理	401	9.5.2 添加代码	449
8.4.2 沙盒在哪里	402	9.6 编码之实现 MyInfo 页面	451

9.6.1 准备一个 HTML 5 小应用	452
9.6.2 添加代码.....	454
9.7 结语.....	455
第 10 章 Swift 致富版魔法盒.....	456
10.1 移动 App 赢利模式探讨	456
10.2 广告平台的选择	457
10.3 多盟广告平台 SDK 的使用	458
10.3.1 注册平台应用.....	458
10.3.2 设置广告类型.....	459
10.3.3 下载广告发布标识 和 SDK.....	460
10.3.4 添加代码.....	460
10.4 真机调试	462
10.4.1 基础知识.....	462
10.4.2 创建应用标识.....	463
10.4.3 创建开发证书 (Development)	466
10.4.4 创建标识文件.....	469
10.4.5 添加设备	473
10.4.6 启用真机调试	474
10.5 打包 IPA 测试.....	475
10.6 云测试平台 FIR.im 使用.....	479
10.7 上传 AppStore.....	482
10.8 结语	484

第 1 章 从零搭建 Swift 开发环境

小伙伴们，感谢你看完了絮絮叨叨的前言部分，你应该对本书整体的目录和结构有了全面的了解。接下来欢迎步入第 1 章。小伙伴们是不是和老码一样迫不及待地想在 XCode 中写几行代码，看看那漂亮的 HelloWorld 呢？好吧，我们开始吧。

目前开发苹果应用程序的设备有苹果电脑和供学习用的安装移植版本的普通 X86 电脑。前者所谓的苹果电脑分别有便携式机箱设备（Mac）、笔记本电脑（MacBook）和一体式电脑（iMac），具体的区别小伙伴们可以直接去网上搜。这里我们要关注的是后者，作为仅供学习用的 X86 电脑，我们称之为苹果学习机。学习机又分为实体学习机和虚拟学习机。

- **实体学习机**：即在 X86 结构的普通 PC 上直接安装 Mac 操作系统，所安装的 Mac 操作系统需要经过移植，以及开源社区对主流显卡、网卡及附属设备的驱动所做的适配。简单地说，就是在我们平时用的普通 PC 直接安装苹果操作系统。
- **虚拟学习机**：即在我们普通 PC 上的 Windows 操作系统中，利用 VMware 或者 Vbox 虚拟安装苹果操作系统。相对而言，这种方式安装简单，可以在 Windows 与苹果操作系统间自由切换，但是对硬件的要求较高，特别是内存。

这里老码需要从版权保护的角度对这两种安装方式强调一下，学习机只是供学生群体及想学习苹果系统的小伙伴们学习之用，且不可以用于任何商业活动。如果小伙伴们想做 iOS 开发，老码强力推荐大家购买一台苹果电脑。相信老码，使用苹果电脑是一种跨越艺术和科技的美感体验。那么接下来我们就先从虚拟学习机开始吧！这也是老码强力推荐的作为学习用途的最富性价比的方式。

1.1 虚拟机安装 Mac OS

1.1.1 准备工作

准备工作如表 1.1 所示。

表 1.1

类型	配置	备注
硬件	• 内存大于或者等于 4GB • CPU 无限制，目前主流的 CPU 都可以 • 硬盘大于 120GB，最好选择 SSD（固态硬盘），如果选择机械硬盘，则内存需要大于 6GB	如果内存过小或者采用机械硬盘，则会出现令人窒息的卡顿。 本案例的电脑配置如下。 • 处理器：Intel(R) Core(TM) i5-4310U CPU@2.00GHZ 2.6GHZ • 内存：8.00GB • 操作系统：Windows 7 Enterprise • 硬盘：SSD 256GB
软件	VMware® Workstation 9.0 以上	当下 Windows 上最流行虚拟机软件，当然你也可以选择 Virtual PC、Vbox 等
	Maverick Install_10.9.3.cdr	Mac OS 10.9.3 懒人版安装镜像，为什么叫懒人版：我们从 AppStore 下载的 Mac OS 安装镜像以.app 结尾，开源社区基于苹果的使用协议，制作了供学习所用的安装镜像，这个制作过程比较复杂，但开源社区帮我们做了这些背后的工作，最后发布出一个供我们方便安装的版本，意思是供“懒人”使用，因此这个开源社区帮我们做的版本就叫作懒人版安装镜像
	VMware 工具集(unlock-all-v120)	默认安装的 VMware 没有安装 Mac OS 的选项。此工具集可以打开 VMware Mac OS 安装选项
	VMSVGA2 v1.2.5 OS 10.9.zip	该工具集可以增强所安装系统的 VGA 显卡显示效果，画面更加清晰，切换更加流畅
	VMSVGA Guested Patches2.zip	当安装上面的显卡增强工具集，则可能无法自动调节虚拟机分辨率，使用该工具集可以解决这个问题
	VMTool 6.0.2.iso	VMware 辅助工具，提供客户机和主机之间的适配，增强功能，安装之后，可以使虚拟机中的 Mac 系统全屏，或者直接在 Windows 系统与 Mac 系统之间直接复制数据、拖曳文件等
	Tinker Tool 5.2.app	该工具可以安装在 Mac 系统内，做一些通用配置，比如禁用特效、增强系统流畅性

1.1.2 动手吧

小伙伴们，一定要明白学编程是不能偷懒的，不管多么复杂的技术，动手跟着做一遍即可。正如我们的虚拟机安装，请按照下面的图片和老码的解释一步步来操作。当然在开始之前，老码已经认为你顺利安装了：VMware® Workstation 9.0/10.0 的软件，老码使用的是

VMware® Workstation 10.0 软件，后续章节，我们简称 VMware 10.0。

1.1.2.1 安装虚拟机

这个软件的功能已经在上一节给小伙伴们做了一个大致的描述，即便你不是首次安装虚拟机镜像而是复制别人的镜像直接使用，仍然必须确保安装这个工具集（图 1.1），否则小伙伴们的虚拟机将进入黑屏状态，原因是缺少该工具集提供 Darwin 的引导程序。并且务必记得需要右键单击 `install.cmd` 文件（图 1.2），然后选择 `Run As Administrator`（以管理员方式执行）命令才可以，否则即使安装也没有什么效果。

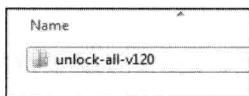


图 1.1

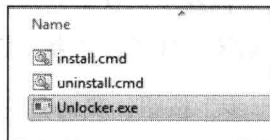


图 1.2

安装好了补丁，我们打开 VMware 10.0 虚拟机软件，如图 1.3 所示。

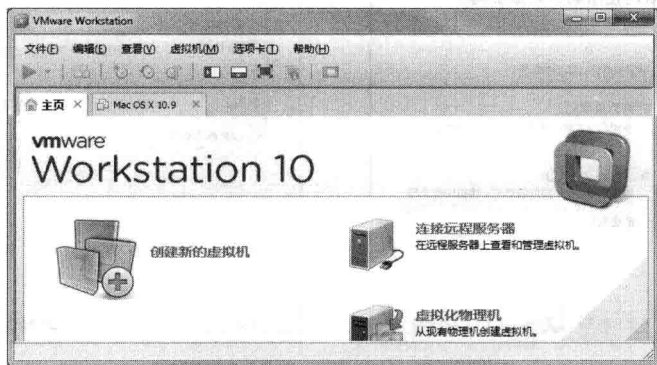


图 1.3

如果小伙伴们是第一次安装虚拟机，打开该软件时会首先到如图 1.3 所示“主页”界面，当然如果你已经有了虚拟机，则会直接进入虚拟机的选择页面，这里老码按照首次安装的方式介绍。在如图 1.3 所示的页面中有三个选项。

- **创建新的虚拟机**：该功能会引导小伙伴们从头创建虚拟机镜像，这也是我们最常用的功能。
- **连接远程服务器**：这里指连接远程的虚拟机服务器，该功能一般用在公司化的环境里，有些公司会有统一的虚拟机服务器，用户通过分配的账号密码即可连接服务器。
- **虚拟化物理机**：这属于高级功能，如果小伙伴们的硬件资源足够，可以试试这个功能。

1.1.2.2 配置新的虚拟机

我们接着单击“创建新的虚拟机”按钮，进入虚拟机安装界面，如图 1.4 所示。小伙伴们要留意这里的两个选项。

- **典型（推荐）**：这里是 VMware 为了简化用户操作提供一些更加便捷的方式创建虚拟机，很多软件都有类似这样的功能。如果想安装一个 Windows 8、Linux 等普通的虚拟机选这个选项即可，但是安装 MacOS 这样的虚拟机就不能选用这个。
- **自定义（高级）**：这里允许小伙伴们做一些额外的配置功能，比如硬盘类型、内存大小等，后续章节老码会给大家介绍这里的奥秘。

单击“下一步”按钮，我们会进入如图 1.5 所示的界面，这里我们需要选择硬件兼容类型，如图 1.6 所示。

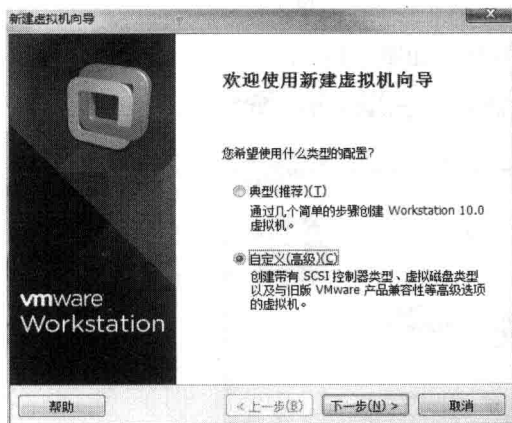


图 1.4

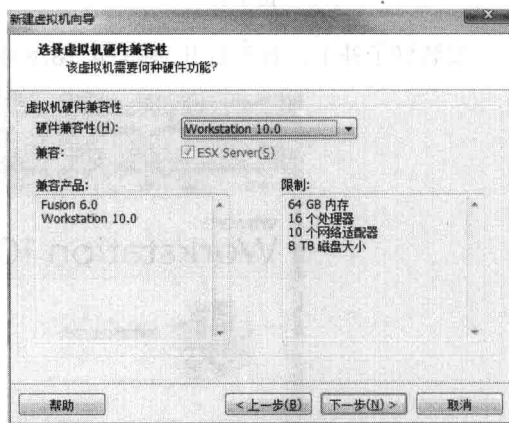


图 1.5

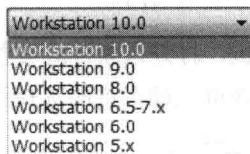


图 1.6

老码在这里选择的是 Workstation 10.0，它可以支持目前主流的大部分硬件类型。选好后我们进入如图 1.7 所示的界面。

在如图 1.7 所示的界面，小伙伴们注意一定得选“稍后安装操作系统”选项，因为在安装系统前我们还要做一些配置。单击“下一步”按钮，进入如图 1.8 所示的界面，这里有很多客