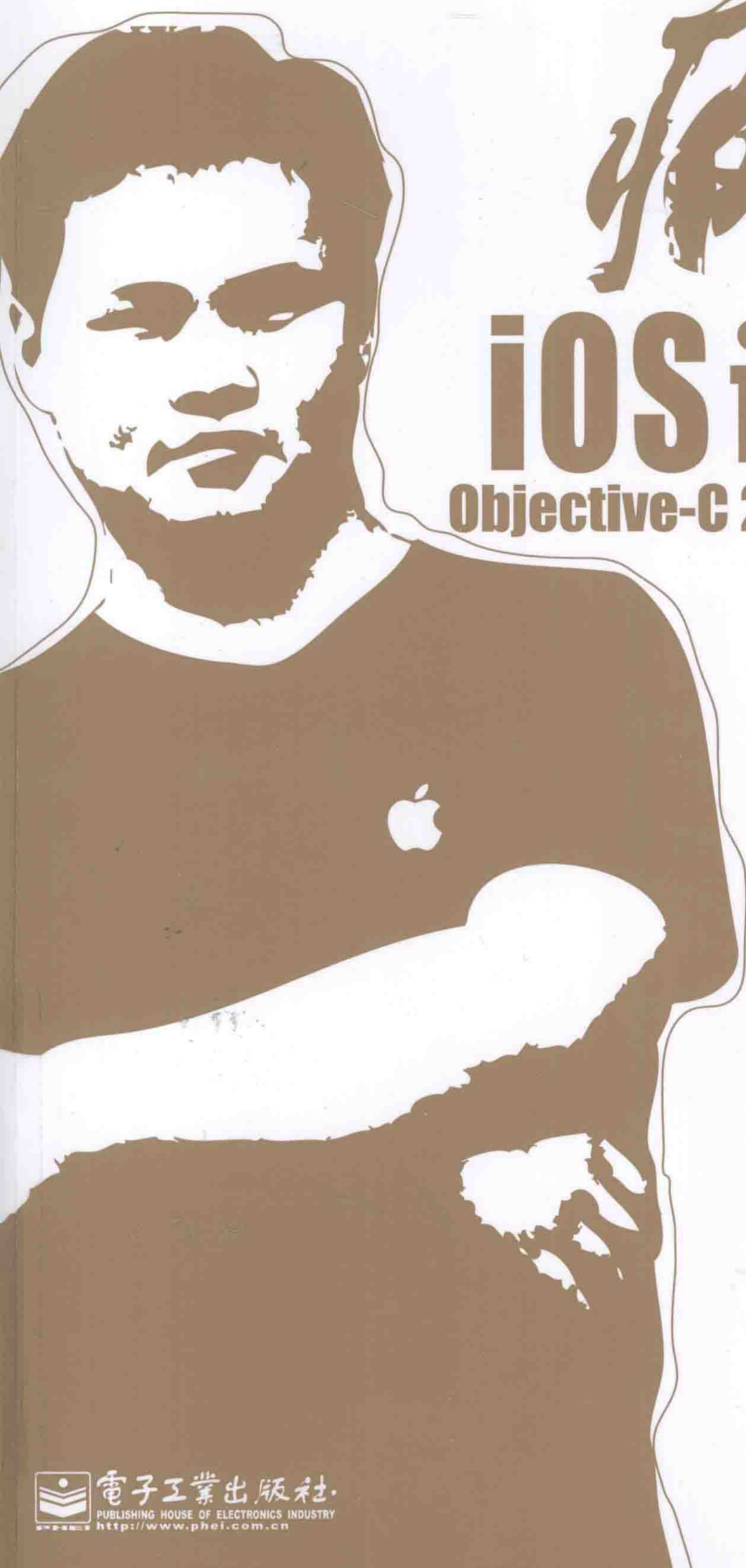




疯狂

iOS讲义(上)

Objective-C 2.0与iPhone/iPad
应用开发基础

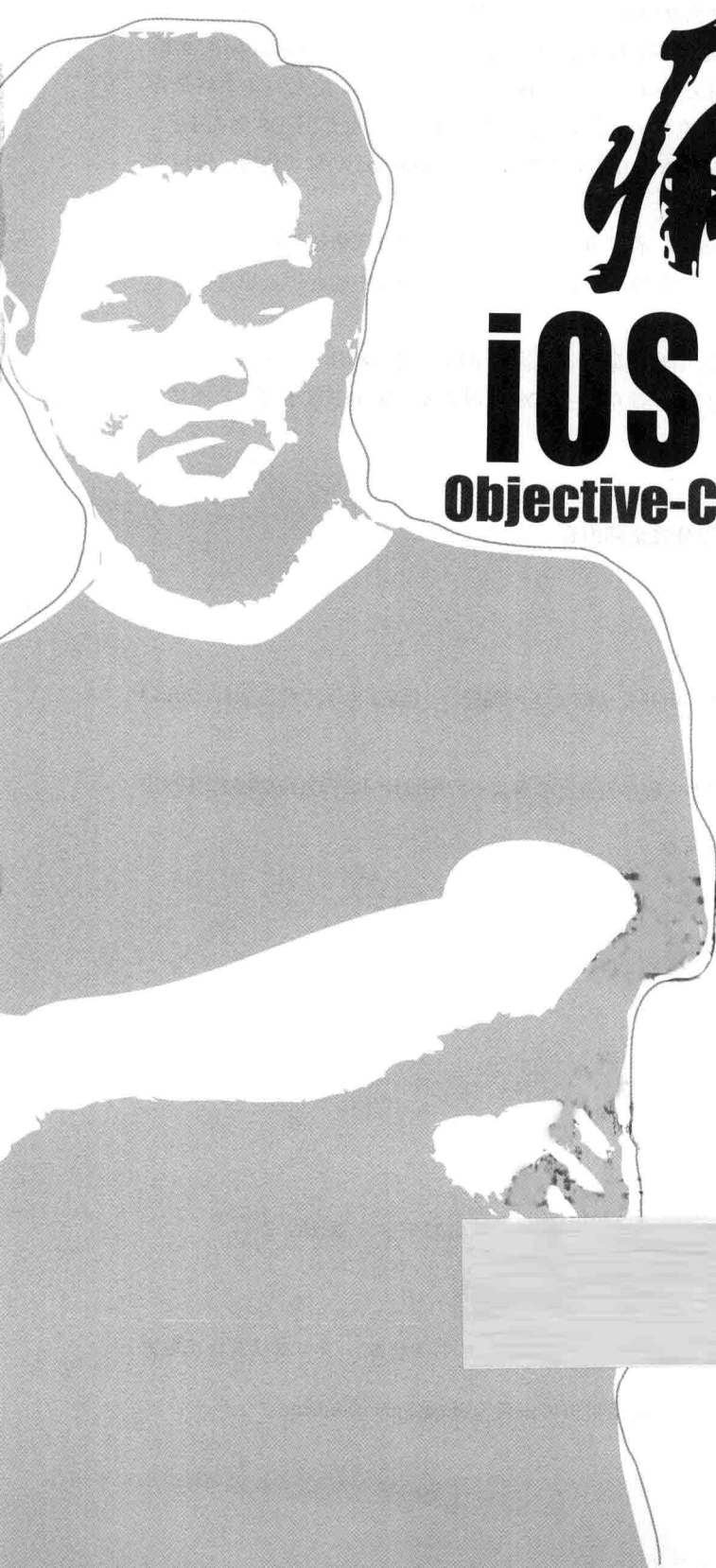
李刚 编著

疯狂源自梦想

技术成就辉煌

疯狂源自梦想
技术成就辉煌





疯狂

iOS讲义(上)

Objective-C 2.0与iPhone/iPad
应用开发基础

李刚 编著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书以最新的 iOS 7.0、Xcode 5 为平台, 全面介绍 iOS 应用开发的基础知识。全书从 Objective-C 基本语法开始, 详细介绍了 Objective-C 的基本语法结构、Objective-C 的面向对象特征、Foundation 框架的核心类库用法、文件 I/O 体系等知识, 掌握 Objective-C 语言是 iOS 引用开发的基础。之后, 本书将带领读者掌握 iOS 应用开发的基本理论以及 iOS 应用的 MVC 设计和事件处理编程, 重点介绍 iOS 界面编程、iOS 应用的生命周期、Quartz 2D 绘图、坐标变换、Core Image 滤镜、Core Animation 动画等内容。最后介绍了一个《疯狂连连看》游戏实例。

本书并不局限于介绍 iOS 编程的各种理论知识, 而是从“项目驱动”的角度来讲授理论, 全书一共包括近百个实例, 这些示范性的实例既可帮读者更好地理解各知识点在实际开发中的应用, 也可供读者在实际开发时作为参考, 拿来就用。

本书为所有打算深入掌握 iOS 编程的读者而编写, 适合各层次的 iOS 学习者和开发者阅读, 也适合作为大学教育、培训机构的 iOS 教材。但如果只是想简单涉猎 iOS 应用开发, 则本书过于庞大, 不适合阅读。

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

疯狂 iOS 讲义. 上, Objective-C 2.0 与 iPhone/iPad 应用开发基础 / 李刚编著. —北京: 电子工业出版社, 2014.1
ISBN 978-7-121-19107-7

I. ①疯… II. ①李… III. ①移动终端—应用程序—程序设计②C 语言—程序设计 IV. ①TN929.53②TP312

中国版本图书馆 CIP 数据核字 (2013) 第 251375 号

策划编辑: 张月萍

责任编辑: 李利健

印 刷: 北京中新伟业印刷有限公司

装 订: 河北省三河市路通装订厂

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: 787×1092 1/16

印张: 42.75

字数: 1122千字 彩插: 2

印 次: 2014年1月第1次印刷

印 数: 4000册

定价: 99.00元 (含光盘1张)

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。

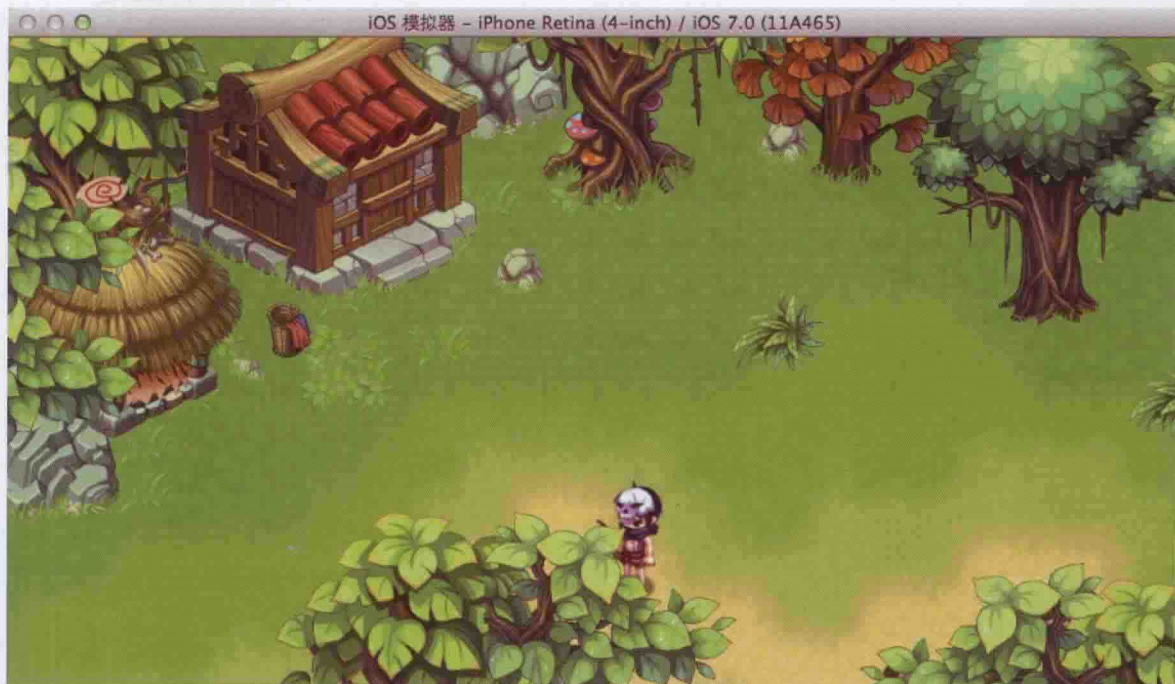
质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

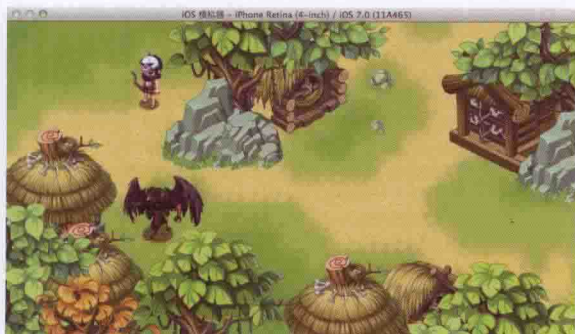
精彩手游，尽在掌控

疯狂 iOS 讲义
(上)(下)册部分

苹果 精彩案例



苹果 萌仙：玩家



苹果 萌仙：玩家遇见怪物



苹果 萌仙：玩家射箭



苹果 萌仙：怪物血花四溅



苹果 萌仙：怪物发射魔法

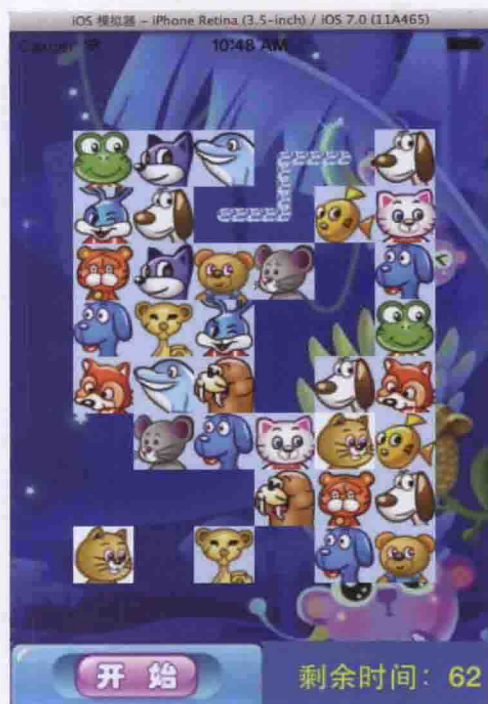


苹果 萌仙：玩家被怪物击中

精彩手游，尽在掌控

疯狂 iOS 讲义
(上)(下)册部分

精彩案例



疯狂连连看



疯狂连连看



打飞机：打中敌机



打飞机：打大Boss

精彩手游，尽在掌控

疯狂 iOS i#X
(上)(下)册部分

精彩案例



怪物足球



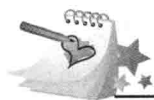
贪食蛇



天气预测首页



天气预测详情



前言

移动互联网热潮在全世界引起了巨大反响，移动互联网正在改变着传统互联网的格局，全世界的 IT 公司争相将业务重心向移动互联网转型，移动互联网业务也成为业内最大的利润增长点。

Apple 公司的 iOS 系统作为与 Android 并驾齐驱的两大操作系统，具有非常完善的生态圈，而且用户具有良好的付费习惯，吸引了大量的专业软件和小团队，甚至包括个人用户。

Apple 公司提供的 App Store 作为应用程序市场，它赋予了缺乏市场运作能力的第三方开发商和草根开发者新的商业机会：开发商和草根开发者只要专注于应用的开发，而市场和销售则交给 Apple 公司管理。在这种模式下，即使小型团队（甚至个人开发者）也可以创造财富神话。

Apple 公司在全世界 IT 行业是一个极富传奇色彩的公司，它一度成为全球市值最大的 IT 公司，Apple 公司的产品对于年轻人始终具有神奇的魔力，尤其是 iPhone 和 iPad 两款广为人知的产品，更是让无数追求时尚的年轻人为之痴迷。目前 Apple 公司也在逐步推出 iTV（基于 iOS 系统的智能电视）、iWatch（智能手表）等各种个人电子消费品，这些产品也吊足了时尚一族的胃口。Apple 公司独特的设计、市场运作方式、商业模式和公司经营管理理念，也是众多 IT 公司竞相研究、模仿的对象。

本书特点



本书是一本介绍 iOS 应用开发的实用图书，全面介绍了 iOS 7.0 平台上应用开发各方面的知识。与市面上介绍 iOS 编程的一些图书不同，本书并没有花太多篇幅介绍 Apple 公司、iOS 的发展历史（因为这些内容到处都是），完全没有介绍 App Store（因为它只是一个交易网站，与 iOS 开发无关，但有些图书甚至用一章内容介绍它）。

本书只用了一章来介绍如何搭建 iOS 开发环境和 Xcode 工具的使用法。全书内容大致可分为两部分。

第一部分详细介绍 Objective-C 这门看似冷门的编程语言——实际上，Objective-C 现在非常热门，Objective-C 语言目前雄踞编程语言排行第 3 的位置（前两名是 Java、C）。本书在介绍 Objective-C 基本语法、面向对象语法时没有过多地使用 IDE 工具，而是带领读者使用最基本的文本编辑器学习编程，使读者真正掌握 Objective-C 语法。

第二部分介绍 iOS 开发的基础，详细介绍 iOS 项目的结构、各文件的作用及其调用关系，并归纳了 iOS 应用的 MVC 设计、在代码中获取 UI 控件的 3 种方式（必须先获取 UI 控件，然后才可修改它来更新用户界面），以及 iOS 应用事件处理的 3 种方式，通过这种方式可以让读者快速掌握 iOS 编程的思路。第二部分的重点是 iOS 应用的 UI 界面开发和图形、图像处理——全书全面、系统地介绍了 iOS 的各种 UI 控件的功能、用法及其定制方法，并通过相应的实例程序给出了示范。

本书是《疯狂 iOS 讲义》（上卷），因此关于手势、数据库操作、多线程编程、网络通信、cocos2d、Box2D、SpriteKit 等高级内容将会在下卷介绍。

需要说明的是，本书只是一本介绍 iOS 应用开发的图书，不是一本关于所谓“思想”的书，不要指望学习本书能提高你所谓的“iOS 思想”，所以那些希望提高编程思想的读者请不要阅读

本书。

本书更不是一本看完之后可以“吹嘘、炫耀”的书——因为本书并没有堆砌一堆“深奥”的新名词和“高深”的思想，本书保持了“疯狂体系”的一贯风格：操作步骤详细、编程思路清晰，语言平实。只要读者想学习 iOS 开发，无须担心没有 Apple 电脑（网络上使用普通电脑装黑苹果的文章铺天盖地），也无须担心没有 iPhone（Xcode 集成的模拟器在大部分场景下完全可以胜任开发）——即使读者以前没有编程基础，只要有决心和毅力，阅读本书不会有任何问题（本书从最基本的 Objective-C 语法开始讲起）。

认真看完本书，并把书中所有的示例都练习一遍，本书带给你的只是 9 个字：“看得懂、学得会、做得出”。本书最后综合运用前面介绍的 UI 编程、绘图等知识编写了一个《疯狂连连看》小游戏，因此认真阅读本书，掌握书中知识后，即可开发各种棋类、牌类、益智类的小游戏。

不管怎样，只要读者在阅读本书时遇到了技术问题，都可以登录“疯狂 Java 联盟”（<http://www.crazyit.org>）与广大学习者交流，笔者也会通过该平台与大家一起交流、学习。

本书还具有如下几个特点。

1. 知识全面，覆盖面广

本书深入阐述了 Objective-C 的基本语法，包括类、对象、方法、成员变量、KVC、KVO、继承和多态等面向对象的基础知识，也包括类别（category）、扩展（extension）、协议与委托、内存管理与 ARC 等 Objective-C 的高级知识。本书详细介绍了 iOS 全部图形界面控件的功能和用法，事件处理机制，iOS 图形、图像处理。

2. 内容实际，实用性强

本书并不局限于枯燥的理论介绍，而是采用“项目驱动”的方式来讲授知识点，每个知识点几乎都可找到对应的参考实例。本书最后还提供了《疯狂连连看》游戏应用，具有极高的参考价值。

3. 讲解详细，上手容易

只要读者有学习的决心和毅力，学完本书内容后就打下扎实的 iOS 开发基础，再通过本书下卷来掌握 iOS 开发的高级知识，即可完全胜任企业中 iOS 开发者的实际要求。

本书读者



本书为所有打算深入掌握 iOS 编程的读者而编写，适合各层次的 iOS 学习者和开发者阅读，也适合作为大学教育、培训机构的 iOS 教材。但如果只是想简单涉猎 iOS 应用开发，则本书过于庞大，不适合阅读。

2013 年 9 月



CONTENTS

第 1 章 iOS 应用与开发环境 1

- 1.1 Objective-C 与 iOS 简介 2
 - 1.1.1 Objective-C 简介 2
 - 1.1.2 iOS 简介 3
- 1.2 搭建 iOS 开发环境 3
 - 1.2.1 下载并安装 Xcode 和 SDK 4
 - 1.2.2 安装辅助工具和文档 6
- 1.3 第一个 Objective-C 程序 6
 - 1.3.1 使用文本编辑器开发 Objective-C 程序 7
 - 1.3.2 使用 Xcode 工具开发 Objective-C 程序 9
- 1.4 熟悉 Xcode 10
 - 1.4.1 创建 iOS 项目 11
 - 1.4.2 熟悉导航面板 12
 - 1.4.3 熟悉检查器面板 14
 - 1.4.4 熟悉库面板 15
 - 1.4.5 使用 Xcode 的帮助系统 17
- 1.5 小结 20

第 2 章 数据类型和运算符 21

- 2.1 注释 22
- 2.2 标识符和变量 23
 - 2.2.1 分隔符 23
 - 2.2.2 标识符规则 25
 - 2.2.3 Objective-C 关键字 25
- 2.3 数据类型分类 25
- 2.4 基本数据类型 26
 - 2.4.1 整型 26
 - 2.4.2 NSLog 输出函数及格式字符 27
 - 2.4.3 字符型 29
 - 2.4.4 浮点型 30
 - 2.4.5 枚举型 31
 - 2.4.6 BOOL 类型 32
- 2.5 类型转换 34
 - 2.5.1 自动类型转换 34
 - 2.5.2 强制类型转化 36
 - 2.5.3 表达式类型的自动提升 36
- 2.6 运算符 37

- 2.6.1 算术运算符 37
- 2.6.2 赋值运算符 39
- 2.6.3 位运算符 40
- 2.6.4 扩展后的赋值运算符 42
- 2.6.5 比较运算符 42
- 2.6.6 逻辑运算符 43
- 2.6.7 逗号运算符 43
- 2.6.8 三目运算符 44
- 2.6.9 运算符的结合性和优先级 45
- 2.7 小结 46

第 3 章 流程控制与数组 47

- 3.1 顺序结构 48
- 3.2 分支结构 48
 - 3.2.1 if 条件语句 48
 - 3.2.2 switch 分支语句 52
- 3.3 循环结构 54
 - 3.3.1 while 循环语句 54
 - 3.3.2 do while 循环语句 56
 - 3.3.3 for 循环 56
 - 3.3.4 嵌套循环 59
- 3.4 控制循环结构 60
 - 3.4.1 使用 break 结束循环 61
 - 3.4.2 使用 continue 结束本次循环 61
 - 3.4.3 使用 return 结束函数 62
- 3.5 goto 语句 62
- 3.6 数组 64
 - 3.6.1 定义数组 64
 - 3.6.2 数组的初始化 65
 - 3.6.3 使用数组 66
 - 3.6.4 多维数组 67
 - 3.6.5 多维数组的初始化 68
 - 3.6.6 字符数组、字符串与结束标志 73
 - 3.6.7 输出字符串和字符串函数 74
 - 3.6.8 数组的应用举例 76
- 3.7 小结 78

第 4 章 C 语言特性 79

- 4.1 函数 80
 - 4.1.1 定义函数 80

4.1.2	函数声明.....	83	4.10	块.....	136
4.1.3	函数的参数传递机制.....	84	4.10.1	块的基本语法.....	136
4.1.4	递归函数.....	88	4.10.2	块与局部变量.....	137
4.1.5	数组作为函数参数.....	89	4.10.3	直接使用块作为参数.....	139
4.1.6	内部函数与外部函数.....	93	4.10.4	使用 typedef 定义块变量 类型.....	139
4.2	局部变量与全局变量.....	94	4.11	小结.....	140
4.2.1	局部变量.....	95	第 5 章	面向对象 (上).....	141
4.2.2	全局变量.....		5.1	类和对象.....	142
4.2.3	外部全局变量与内部全局 变量.....	97	5.1.1	定义类.....	142
4.2.4	动态存储与静态存储.....	99	5.1.2	对象的产生和使用.....	146
4.3	预处理.....	101	5.1.3	对象和指针.....	148
4.3.1	使用#define、#undef 执行宏 定义.....	101	5.1.4	self 关键字.....	150
4.3.2	带参数的宏定义.....	103	5.1.5	id 类型.....	153
4.3.3	使用#ifdef、#ifndef、#else、 #endif 执行条件编译.....	104	5.2	方法详解.....	154
4.3.4	使用#if、#elif、#else、 #endif 执行条件编译.....	106	5.2.1	方法的所属性.....	154
4.3.5	#include 与#import.....	107	5.2.2	形参个数可变的方法.....	155
4.4	指针.....	108	5.3	成员变量.....	156
4.4.1	指针的概念.....	108	5.3.1	成员变量及其运行机制.....	156
4.4.2	指针变量及其基本用法.....	109	5.3.2	模拟类变量.....	158
4.4.3	指针变量作为函数参数.....	111	5.3.3	单例 (Singleton) 模式.....	159
4.5	指针与数组.....	113	5.4	隐藏和封装.....	160
4.5.1	指向数组的指针变量.....	113	5.4.1	理解封装.....	161
4.5.2	指针运算.....	114	5.4.2	使用访问控制符.....	161
4.5.3	数组变量作为函数参数.....	116	5.4.3	理解@package 访问控制符.....	165
4.5.4	指向多维数组的指针变量.....	118	5.4.4	合成存取方法.....	166
4.6	字符串与指针.....	121	5.4.5	使用点语法访问属性.....	171
4.6.1	使用字符指针表示字符串.....	121	5.5	键值编码 (KVC) 与键值监听 (KVO)	172
4.6.2	字符指针作为函数参数.....	122	5.5.1	简单的 KVC.....	172
4.7	函数与指针.....	123	5.5.2	处理不存在的 key.....	175
4.7.1	用函数指针变量调用函数.....	123	5.5.3	处理 nil 值.....	176
4.7.2	用函数指针变量作为函数 参数.....	124	5.5.4	Key 路径.....	178
4.7.3	返回指针的函数.....	126	5.5.5	键值监听 (KVO).....	179
4.8	指针数组和指向指针的指针.....	127	5.6	对象初始化.....	182
4.8.1	指针数组与 main() 函数形参.....	127	5.6.1	为对象分配空间.....	182
4.8.2	指向指针变量的指针.....	130	5.6.2	初始化方法与对象初始化.....	183
4.9	结构体.....	130	5.6.3	便利的初始化方法.....	185
4.9.1	定义结构体.....	131	5.7	类的继承.....	187
4.9.2	typedef 语句.....	133	5.7.1	继承的特点.....	187
4.9.3	初始化结构体变量.....	134	5.7.2	重写父类的方法.....	189
4.9.4	结构体数组.....	135	5.7.3	super 关键字.....	190
			5.8	多态.....	193
			5.8.1	多态性.....	193

5.8.2 指针变量的强制类型转换.....	196	6.8.2 @autoreleasepool 块.....	251
5.8.3 判断指针变量的实际类型.....	197	6.9 小结.....	251
5.9 小结.....	198	第7章 Foundation 框架详解.....	252
第6章 面向对象（下）.....	199	7.1 字符串（NSString 与 NSMutableString）.....	253
6.1 Objective-C 的包装类.....	200	7.1.1 创建字符串.....	253
6.1.1 它们不是包装类.....	200	7.1.2 NSString 的常用功能.....	254
6.1.2 NSValue 和 NSNumber.....	201	7.1.3 可变字符串（NSMutableString）.....	255
6.2 处理对象.....	202	7.2 日期与时间.....	256
6.2.1 打印对象和 description 方法.....	202	7.2.1 日期与时间（NSDate）.....	256
6.2.2 == 和 isEqual 方法.....	205	7.2.2 日期格式器（NSDateFormatter）.....	258
6.3 类别与扩展.....	209	7.2.3 日历（NSCalendar）与日期组件（NSDateComponents）.....	260
6.3.1 类别（category）.....	210	7.2.4 定时器（NSTimer）.....	261
6.3.2 利用类别对类进行模块化设计.....	212	7.3 对象复制.....	263
6.3.3 使用类别来调用私有方法.....	212	7.3.1 copy 与 mutableCopy 方法.....	263
6.3.4 扩展（extension）.....	214	7.3.2 NSCopying 与 NSMutableCopy 协议.....	264
6.4 协议（protocol）与委托.....	216	7.3.3 浅复制与深复制.....	266
6.4.1 规范、协议与接口.....	216	7.3.4 setter 方法的复制选项.....	268
6.4.2 使用类别实现非正式协议.....	217	7.4 Objective-C 集合概述.....	269
6.4.3 正式协议的定义.....	218	7.5 数组（NSArray 与 NSMutableArray）.....	270
6.4.4 遵守（实现）协议.....	219	7.5.1 NSArray 的功能与用法.....	270
6.4.5 协议与委托（delegate）.....	222	7.5.2 对集合元素整体调用方法.....	275
6.5 使用@try 处理异常.....	225	7.5.3 对 NSArray 进行排序.....	276
6.5.1 使用@try...@catch...@finally 捕捉异常.....	226	7.5.4 使用枚举器遍历 NSArray 集合元素.....	278
6.5.2 访问异常信息.....	228	7.5.5 快速枚举（for...in）.....	279
6.5.3 使用@finally 回收资源.....	229	7.5.6 可变数组（NSMutableArray）.....	279
6.5.4 抛出异常与自定义异常类.....	230	7.5.7 NSArray 的 KVC 与 KVO.....	281
6.6 Objective-C 反射机制.....	232	7.6 集合（NSSet 与 NSMutableSet）.....	282
6.6.1 获得 Class.....	232	7.6.1 NSSet 的功能与用法.....	282
6.6.2 检查继承关系.....	233	7.6.2 NSSet 判断集合元素重复的标准.....	285
6.6.3 动态调用方法.....	235	7.6.3 NSMutableSet 的功能与用法.....	288
6.7 手动内存管理.....	237	7.6.4 NSCountedSet 的功能与用法.....	289
6.7.1 对象的引用计数.....	239	7.7 有序集合（NSOrderedSet 与 NSMutableOrderedSet）.....	290
6.7.2 对象所属权.....	241	7.8 字典（NSDictionary 与 NSMutableDictionary）.....	291
6.7.3 方法中的保留和释放.....	243	7.8.1 NSDictionary 的功能与用法.....	292
6.7.4 使用自动释放池.....	245	7.8.2 对 NSDictionary 的 key 排序.....	295
6.7.5 自动释放池的销毁时机与工作过程.....	246		
6.7.6 临时对象与事件循环中的自动释放池.....	248		
6.7.7 手动内存管理的规则总结.....	249		
6.8 自动引用计数.....	250		
6.8.1 自动引用计数概述.....	250		

7.8.3 对 NSDictionary 的 key 进行过滤	297	9.2.3 理解 iOS 的 MVC	346
7.8.4 使用自定义类作为 NSDictionary 的 key	298	9.2.4 掌握 UIViewController 控制器	347
7.8.5 NSMutableDictionary 的功能与用法	299	9.3 事件机制	349
7.9 谓词	301	9.3.1 程序获取控件的两种方式	349
7.9.1 定义谓词	301	9.3.2 事件处理的方式	353
7.9.2 使用谓词过滤集合	302	9.4 Storyboard 的改进之处	358
7.9.3 在谓词中使用占位符参数	303	9.5 代码控制 UI 界面	360
7.9.4 谓词表达式语法	305	9.5.1 不使用界面布局文件开发 UI 界面	360
7.10 小结	307	实例: 无界面布局文件开发 iOS 应用	360
第 8 章 文件 I/O	308	9.5.2 使用代码创建 UI 界面	362
8.1 使用 NSData 与 NSMutableData	309	实例: 动态添加、删除标签	362
8.2 使用 NSFileManager 管理文件和目录	310	9.5.3 自定义 UI 控件	364
8.2.1 访问文件属性和内容	311	实例: 跟随手指运动的小球	365
8.2.2 创建、删除、移动、复制	312	9.6 美化 iOS 应用	367
8.2.3 查看目录包含的内容	313	9.6.1 定制 iOS 应用图标	367
8.3 使用 NSPathUtilities.h 管理路径	314	9.6.2 设置 iOS 应用的启动画面	367
8.4 使用 ProcessInfo 获取进程信息	316	9.7 小结	369
8.5 使用 NSFileHandle 处理文件 IO	317	第 10 章 iOS 应用的界面编程	370
8.6 使用 NSURL 读取网络资源	319	10.1 UI 控件概述	371
8.7 使用 NSBundle 处理项目相关资源	320	10.1.1 活动、静态和被动控件	371
8.8 对象归档	322	10.1.2 UIControl 基类介绍	371
8.8.1 使用 NSKeyedArchiver 归档	322	10.2 按钮 (UIButton)	373
8.8.2 实现 NSCoder 协议	323	实例: 丰富多彩的按钮	375
8.8.3 使用 NSData 完成自定义归档	326	10.3 文本框 (UITextField)	377
8.8.4 使用归档实现深复制	329	实例: 用户登录界面	379
8.9 小结	330	10.3.1 完成输入后关闭键盘	380
第 9 章 iOS 应用开发入门	331	10.3.2 触碰背景关闭键盘	381
9.1 从 iOS 项目开始	332	10.4 多行文本控件 (UITextView)	382
9.1.1 iOS 项目包含的文件	332	10.4.1 UIScrollView 支持的属性	383
9.1.2 Interface Builder 简介	333	10.4.2 使用委托对象处理 UITextView 事件	385
9.1.3 向 nib 文件添加控件	336	实例 1: 导航按钮关闭虚拟键盘	385
9.1.4 修改控件属性	338	实例 2: 自定义键盘附件关闭虚拟键盘	387
9.1.5 UIView 支持的属性	339	10.4.3 自定义选择内容后的菜单	389
9.1.6 UILabel 支持的属性	340	10.5 开关按钮 (UISwitch)	390
9.2 MVC	342	实例: 使用开关控制“手电筒”	390
9.2.1 从程序入口开始	342	10.6 分段控件 (UISegmentedControl)	391
9.2.2 应用程序委托	343	实例 1: 通过分段控件控制背景颜色	393
		实例 2: 动态增加、删除分段	394

10.7 图像控件 (UIImageView)	395	10.16.8 使用 iOS 6 新增的 UIRefresh Controller 刷新表格	465
实例 1: 图片浏览器	396	10.17 搜索条 (UISearchBar)	467
实例 2: “幻灯片”播放器	398	10.17.1 使用搜索条	468
10.8 进度条	399	10.17.2 使用 UISearchDisplayController	471
10.8.1 UIProgressView	399	实例: 下拉列表显示搜索结果	471
实例: 定制进度条	400	10.18 导航条 (UINavigationController) 与导航 条控制器 (UINavigationController)	475
10.8.2 UIActivityIndicatorView	401	10.18.1 使用导航条	475
10.9 拖动条 (UISlider)	403	10.18.2 使用 UINavigationController	477
实例: 通过拖动滑块改变图片透明度	404	实例: 查看、编辑图书信息	479
10.10 警告框 (UIAlertView) 与 UIActionSheet	405	10.18.3 使用 segue 控制过渡	484
10.10.1 UIAlertView 的基本用法	405	10.18.4 使用 segue 控制导航	487
10.10.2 带输入框的 UIAlertView	407	10.19 网格 (UICollectionView) 与网格控 制器 (UICollectionViewController) ...	490
10.10.3 使用 UIActionSheet	409	10.19.1 UICollectionView 的功能与 用法	492
10.11 日期选择器 (UIDatePicker)	411	10.19.2 使用 UICollectionViewDelegate FlowLayout 定制布局	497
实例: 倒计时器	413	实例: 单元格大小不同的网格	497
10.12 选择器 (UIPickerView)	415	10.19.3 扩展 UICollectionViewLayout 定制布局	498
10.12.1 单列选择器	416	实例: 环形布局的 UICollectionView	499
10.12.2 多列选择器	417	10.19.4 多分区网格	502
10.12.3 相互依赖的多列选择器	419	实例: 带页眉、页脚的多分区网格 ..	502
10.12.4 自定义选择器视图	421	10.19.5 直接使用 UICollection ViewController	506
实例: “老虎机”游戏	421	10.20 标签条 (UITabBar) 与标签页 控制器 (UITabBarController)	509
10.13 微调器 (UIStepper)	426	10.20.1 使用 UITabBar	510
10.14 网页控件 (UIWebView)	428	10.20.2 使用 UITabBarController	511
实例: 迷你浏览器	430	10.20.3 在 Storyboard 中使用 UITabBarController	514
10.15 工具条 (UIToolbar)	431	10.21 页控件 (UIPageControl) 与页控 制器 (UIPageViewController)	515
实例: 自定义工具条控件	433	10.21.1 UIPageControl 与 UIScrollView 结合使用	515
10.16 表格 (UITableView) 与表格控制器 (UITableViewController)	435	10.21.2 使用 UIPageViewController	520
10.16.1 简单表格	437	10.22 iPad 应用开发	523
10.16.2 处理单元格的选中	440	10.22.1 使用 iPad 专用控件: UISplitViewController	524
实例: 编辑选中行	441	10.22.2 使用 UIPopoverController 实现浮动窗口	530
10.16.3 定制表格行	446		
实例 1: 继承 UITableViewCell 定制 表格行	446		
实例 2: 使用动态单元格原型定制 表格行	449		
实例 3: 利用 xib 文件定制表格行	451		
10.16.4 编辑表格	454		
10.16.5 多分区表格以及分区索引	459		
10.16.6 直接使用 UITableViewController	461		
10.16.7 静态单元格	464		

10.22.3 开发同时兼容 iPhone 与 iPad 的应用	533	12.3.2 坐标变换与路径结合使用	594
10.23 小结	536	实例: 雪花飘飘	594
第 11 章 管理 iOS 应用的运行	537	12.3.3 使用矩阵变换	596
11.1 自动旋转机制	538	12.4 控制叠加模式	599
11.1.1 配置应用支持的方向	538	12.5 处理填充	604
11.1.2 指定视图控制器支持的方向	540	12.5.1 渐变填充	605
11.2 旋转时重构用户界面	541	12.5.2 模式填充	609
11.3 旋转时切换视图	543	12.6 使用 Core Image 滤镜	614
11.4 iOS 应用的生命周期	546	12.7 动画	621
11.4.1 应用程序的状态	547	12.7.1 Core Animation 动画基础	621
11.4.2 应用程序启动过程	548	12.7.2 使用 CALayer	622
11.4.3 程序入口: main 函数	549	12.7.3 使用 CATransition 控制过渡动画	625
11.4.4 响应中断	550	12.7.4 使用属性动画	629
11.4.5 进入后台	552	12.7.5 控制移动路径	634
11.4.6 返回前台运行	553	实例: 绕圈游动的小鱼	634
11.4.7 应用程序终止	554	12.8 小结	636
11.4.8 Main Run Loop	554	第 13 章 疯狂连连看	637
11.5 小结	555	13.1 游戏简介	638
第 12 章 图形、图像与动画	556	13.2 开发游戏界面	639
12.1 使用 UIImage 和 CGImage 处理位图	557	13.2.1 开发界面布局	639
12.1.1 显示动画	557	13.2.2 开发游戏界面控件	639
实例: 蝴蝶飞舞	559	13.2.3 处理方块之间的连接线	644
12.1.2 CGImage 与 CGImageRef	559	13.3 游戏的状态数据模型	646
12.2 Quartz 2D 绘图	560	13.3.1 定义数据模型	646
12.2.1 Quartz 2D 绘图基础: CGContextRef	560	13.3.2 初始化游戏状态数据	647
12.2.2 绘制几何图形	563	13.4 加载界面的图片	651
12.2.3 点线模式	565	13.5 实现游戏视图控制器	653
12.2.4 绘制文本	569	13.6 实现游戏逻辑	655
12.2.5 设置阴影	572	13.6.1 定义 FKGameService 的接口代码	655
12.2.6 使用路径	573	13.6.2 实现 FKGameService	656
实例: 绘制任意多角星	577	13.6.3 获取触碰点的方块	657
12.2.7 绘制曲线	578	13.6.4 判断两个方块是否可以相连	658
实例: 使用曲线绘制多瓣花朵	579	13.6.5 定义获取通道的工具方法	659
12.2.8 在内存中绘图	580	13.6.6 没有转折点的横向连接	661
实例: 绘图板	583	13.6.7 没有转折点的纵向连接	661
12.2.9 绘制位图	587	13.6.8 一个转折点的连接	662
实例: 扩展 UIImage 的功能	588	13.6.9 两个转折点的连接	665
12.3 图形变换	593	13.6.10 找出最短距离	671
12.3.1 使用坐标变换	593	13.7 小结	673

第 1 章

iOS 应用与开发环境

本章要点

- ✎ Objective-C 发展历史与简介
- ✎ iOS 应用开发简介
- ✎ 下载并安装 Xcode 和 SDK
- ✎ 安装辅助工具和文档
- ✎ 使用文本编辑器开发 Objective-C 程序
- ✎ 使用 Xcode 工具开发 Objective-C 程序
- ✎ 熟悉 Xcode 开发工具
- ✎ 掌握创建 iOS 项目的步骤
- ✎ 熟悉 Xcode 导航面板
- ✎ 熟悉 Xcode 检查器面板
- ✎ 熟悉 Xcode 库面板
- ✎ 使用 Xcode 的帮助系统

在正式开始学习 iOS 应用开发之前, 首先必须拥有一台运行 OS X 系统的计算机 (俗称电脑), 通常建议购买一台 Apple 电脑即可, 直接购买的 Apple 电脑都安装有最新的 OS X 系统 (建议选择运行 OS X 10.8 系统的电脑)。如果经济条件有限, 也可使用普通电脑, 甚至使用虚拟机来安装 OS X 系统, 这样也可用于开发 iOS 应用。

Apple 公司为 iOS 应用开发者准备了强大的 IDE 工具: Xcode, Xcode 集成了功能强大的界面设计器——Interface Builder, 它可让初学者充分享受快速上手的快乐, 即使开发者没有任何编程基础, 也可通过拖、拉、拽设计出一个美观的界面。当然, 只是界面而已。如果需要让应用程序能与用户交互, 能按一定逻辑去执行, 最终还是需要通过编码来实现。

本章并不打算介绍任何编程语法, 只介绍如何下载、安装 Xcode 工具, 以及各种辅助工具和文档, 这些工具和文档是开发 iOS 应用的基础。

1.1 Objective-C 与 iOS 简介

iOS 是目前最主流的两种移动设备操作系统之一 (另一个为 Android), 开发者可以通过开发 iOS 应用来获取巨大的成功。而学习 iOS 应用开发, 通常需要掌握 Objective-C 编程语言。

1.1.1 Objective-C 简介

Objective-C 是一种以 C 为基础, 并结合 Smalltalk 特征扩充出来的面向对象语言, 该语言于 20 世纪 80 年代初由 Brad J.Cox 创建。1988 年, NeXT 公司获得了 Objective-C 语言的授权; 1996 年, Apple 公司收购了 NeXT 公司, 这样 Objective-C 语言就变成了 Apple 公司的“御用”编程语言。2007 年, Apple 公司发布了 Objective-C 的升级版, 并称之为 Objective-C 2.0, 这也是本书将要介绍的 Objective-C 的版本。

与 Java、C 等编程语言不同的是, Objective-C 的应用并不太广泛, 很长时间以来, Objective-C 在商用语言排行榜的位置并不靠前, 但由于苹果电子产品的广泛流行, Objective-C 目前是上升势头最快的编程语言, 在最新的商用语言排行榜上, Objective-C 已经排了第 3 位, 仅次于 Java 和 C 这两大巨头。

另外, Objective-C 是一种“专属”Apple 公司的编程语言, 实际上, 目前很少有公司或产品采用 Objective-C 语言。对于大部分学习者来说, 学习 Objective-C 的目的就是为了开发 iOS 应用, 而 iOS 应用开发则具有光明的前景。

需要指出的是, 虽然 Objective-C 是以 C 为基础扩展而来的, 也受到了 Smalltalk 的影响, 但读者无须恐惧, 即使你没有任何 C 语言的编程基础, 甚至根本不知道 Smalltalk 这种语言, 也不影响学习 Objective-C。



提示:

学习任何编程语言总有第一次, 如果你真正掌握过一门或几门编程语言, 再去学习其他编程语言, 总会有些帮助。但即使你现在没有任何编程语言的基础, 你依然可以很好地掌握 Objective-C。Objective-C 依然是一门独立的编程语言, 完全可以从头学习并掌握它。

从某些角度来看, Objective-C 之所以能得到流行, 关键就在于它是开发 iOS 应用的首选语言。虽然 Objective-C 也可用于开发 Mac OS 应用, 但 Objective-C 的流行, 其实主要得益于 iOS 应用的流行。下面我们先简单介绍一下有关 iOS 的一些知识。