

<<苹果开发之Cocoa编程>>

图书基本信息

书名：<<苹果开发之Cocoa编程>>

13位ISBN编号：9787111391364

10位ISBN编号：7111391365

出版时间：2012-9-1

出版时间：机械工业出版社华章公司

作者：Aaron Hillegass,Adam Preble

页数：365

译者：黄敏,郝刚

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<苹果开发之Cocoa编程>>

### 前言

前言如果你正在开发或者希望进行Mac应用开发，那么这本书就是你所需要的。

本书是否涵盖了所有Mac应用编程的相关知识？

当然不是，不过本书涉及了Mac开发接近80%的内容。

读者可以在Apple公司的在线文档中找到剩余的20%，而这20%正是独属于你的。

本书展示了应用开发的基础，包含了Objective-C语言和Cocoa的主要设计模式。

读者还可以熟悉最常用的两个开发工具：Xcode和Instruments。

阅读完本书之后，读者就可以理解并使用Apple公司的在线文档了。

本书包含很多代码。

通过这些代码，笔者将向读者讲解Cocoa社区的一些惯用法。

希望通过实践这些示例代码，读者不仅能够成为一名Cocoa开发人员，而且还能够成为一名优秀的Cocoa开发人员。

第4版涉及在MacOSX10.6和10.7中引入的一些技术，包括Xcode4、ARC（AutomaticReferenceCounting，自动内存管理技术）、blocks、view-basedtableviews和Mac应用商店。

本书专门安排一章用于讲述iOS的基本开发。

本书适合已经具备一些C语言基础且了解一些面向对象知识的开发人员。

假如读者并不知道C语言或面向对象的概念，那么读者首先应该学习

《Objective-CProgramming:TheBigNerdRanchguide》。

读者并不需要具备Mac编程经验。

这是一本实战类的图书，因此读者需要安装MacOSX系统和开发工具。

Xcode4.2、Apple的IDE都可免费下载、安装。

如果读者是Mac的付费会员，或者隶属iOS开发者联盟，也可以从Apple公司的开发者网站下载Xcode（<http://developer.apple.com/>）。

开发者注册成为上述成员，可将自己的应用程序提交到Apple公司的Mac应用商店或iOS应用商店。

笔者将尽可能地让本书更加实用。

Aaron HillegassAdam Preble

## <<苹果开发之Cocoa编程>>

### 内容概要

本书是在Aaron多年教学经验的基础上进行了总结提炼的精华之作，加之前面3版不断推敲和完善，已经成为学习Cocoa编程的入门级经典书籍。

本书系统介绍了Objective-C语言、Cocoa主要的设计模式、必要的开发工具Xcode和Instruments等，同时新引入了ARC、blocks、view-based table

views和Mac应用商店，并专门开辟一章用于介绍在iOS上的基础开发。

相比于之前的版本，本书在Cocoa框架新特性、新技术的基础上，更加侧重实战，也更加结合应用发展的最新趋势。

全书以具体的实例贯穿，引导读者通过实践学习Cocoa框架的基本思想。

在本书中，读者不仅能够看到技术细节，更能看到不断剖析、层层递进的解决问题的思路，甚至可以看到如何查找资料、获取帮助的方法，帮助读者建立规范、科学的设计开发习惯。

阅读本书不要求读者拥有任何的Mac编程经验，只需要了解基本的C语言，以及面向对象的知识即可。

相信本书能够推动苹果中文社区的进一步发展、壮大，也希望能够将Objective-C和Cocoa框架以及苹果产品背后一流的技术实力从细微处展现出来，帮助那些致力于创新、变革以及追求完美的人们。

## <<苹果开发之Cocoa编程>>

### 作者简介

#### Aaron

Hillegass, 资深苹果应用开发工程师、培训讲师和技术作家, 在使用Cocoa进行开发方面拥有15年工作经验, 具有非常高的知名度和非常大的影响力。

曾就职于NeXT和Apple公司, 目前在Big Nerd

Ranch主讲Cocoa编程, 第一套OpenStep ( Cocoa的前身 ) 教程便出自他和本书另一位作者之手。他还写过多本关于Cocoa、Objective-C、iPhone和iOS方面的图书。

#### Adam

Preble, Mac和iOS软件程序开发顾问、资深苹果应用开发工程师、培训讲师和技术作家, 在使用Cocoa、Objective-C以及C/C++进行开发方面拥有多年工作经验, 尤其在应用程序用户体验方面有独到的见解。

目前在Big

Nerd Ranch开发Mac和iOS应用, 也是Cocoa编程的主讲教师。

## <<苹果开发之Cocoa编程>>

### 书籍目录

译者序

前言

致谢

第1章 什么是Cocoa

1.1 历史简介

1.2 开发工具

1.3 编程语言

1.4 对象、类、方法和消息

1.5 框架

1.6 如何阅读本书

1.7 约定

1.8 常见错误

1.9 如何学习

第2章 起步

2.1 Xcode

2.1.1 新建项目

2.1.2 main函数

2.2 Interface Builder

2.2.1 utility区

2.2.2 空白窗口

2.2.3 页面布局

2.2.4 dock

2.2.5 创建一个类

2.2.6 创建一个实例

2.2.7 创建连接

2.3 回顾Objective-C

2.3.1 Objective-C中的类型和常量

2.3.2 查看头文件

2.3.3 编辑实现文件

2.3.4 编译和运行

2.3.5 awakeFromNib

2.4 文档

2.5 你做了什么

2.6 应用程序运行过程

第3章 Objective-C

3.1 创建和使用实例

3.2 使用已有的类

3.2.1 给nil发送消息

3.2.2 NSObject、NSArray、NSMutableArray和NSString

3.2.3 组合和继承

3.3 创建自己的类

3.3.1 创建LotteryEntry类

3.3.2 更改main.m文件

3.3.3 实现Description方法

3.3.4 编写初始化器initializer

## <<苹果开发之Cocoa编程>>

- 3.3.5 带参数的初始化器
- 3.4 调试器
- 3.5 你做了什么
- 3.6 静态分析器
- 3.7 补充知识：消息机制工作原理
- 3.8 挑战
- 第4章 内存管理
  - 4.1 手动内存管理
    - 4.1.1 创建没有内存泄漏的彩票程序
    - 4.1.2 dealloc函数
    - 4.1.3 自动释放对象
    - 4.1.4 retain计数规则
  - 4.2 Accessor方法
  - 4.3 使用ARC
    - 4.3.1 强引用
    - 4.3.2 弱引用
    - 4.3.3 ARC补充知识
- 第5章 Target/Action
  - 5.1 一些NSControl子类的常见用法
    - 5.1.1 NSButton
    - 5.1.2 NSSlider
    - 5.1.3 NSTextField
  - 5.2 开始SpeakLine示例
  - 5.3 XIB文件中的布局设置
  - 5.4 实现SpeakLineAppDelegate类
  - 5.5 补充：通过代码设置
  - 5.6 挑战
  - 5.7 调试建议
- 第6章 Helper对象
  - 6.1 委托
  - 6.2 NSTableView及其dataSource
  - 6.3 用户界面布局
  - 6.4 建立连接
  - 6.5 编辑SpeakLineAppDelegate.m文件
    - 6.5.1 在实现委托时的常见错误
    - 6.5.2 对象委托
  - 6.6 补充知识：委托是怎样工作的
  - 6.7 挑战1：创建委托
  - 6.8 挑战2：创建一个数据源
- 第7章 键-值编码和键-值观察
  - 7.1 键-值编码
  - 7.2 绑定
  - 7.3 键-值观察KVO
  - 7.4 使对象的键可观察
  - 7.5 Properties
  - 7.6 补充知识1：键路径
  - 7.7 补充知识2：键-值观察

## <<苹果开发之Cocoa编程>>

### 第8章 NSArrayController

- 8.1 从RaiseMan应用开始
- 8.2 键-值编码和nil
- 8.3 添加排序功能
- 8.4 补充知识：不使用NSArrayController进行排序
- 8.5 挑战1
- 8.6 挑战2

### 第9章 NSUndoManager

- 9.1 NSInvocation
- 9.2 NSUndoManager的工作机制
- 9.3 为RaiseMan添加撤销功能
- 9.4 键-值观察
- 9.5 撤销编辑操作
- 9.6 插入后立即编辑
- 9.7 补充知识：窗口和undo管理器

### 第10章 Archiving

- 10.1 NSCoder和NSCoding
  - 10.1.1 编码
  - 10.1.2 解码
- 10.2 文档架构
  - 10.2.1 Info.plist和NSDocumentController
  - 10.2.2 NSDocument
  - 10.2.3 NSWindowController
- 10.3 保存和NSKeyedArchiver
- 10.4 载入和NSKeyedUnarchiver
- 10.5 为文件类型设置扩展名和图标
- 10.6 补充知识1：避免死循环
- 10.7 补充知识2：创建Protocol
- 10.8 补充知识3：自动保存文档
- 10.9 补充知识4：没有undo功能的基于文档的应用程序
- 10.10 统一类型标识符

### 第11章 Core Data基本原理

- 11.1 NSManagedObjectModel
- 11.2 Interface
  - 11.2.1 基于视图的表格视图
  - 11.2.2 连接和绑定
  - 11.2.3 Core Data 如何工作
- 11.3 补充知识：基于视图和基于单元格的表格视图比较
- 11.4 挑战

### 第12章 NIB文件和NSWindowController

- 12.1 NSPanel
- 12.2 向应用程序添加面板
  - 12.2.1 设置菜单项
  - 12.2.2 AppController.m
  - 12.2.3 Preferences.xib
  - 12.2.4 PreferenceController.m
- 12.3 补充知识：NSBundle

## <<苹果开发之Cocoa编程>>

### 12.4 挑战

### 第13章 User Defaults

#### 13.1 NSDictionary和NSMutableDictionary

##### 13.1.1 NSDictionary

##### 13.1.2 NSMutableDictionary

#### 13.2 NSUserDefaults

#### 13.3 设置Defaults

##### 13.3.1 设置程序标识符

##### 13.3.2 为defaults的名字建立键

##### 13.3.3 注册defaults

#### 13.4 让用户编辑Defaults

#### 13.5 使用Defaults

##### 13.5.1 禁用自动创建未命名新文档

##### 13.5.2 设置表格视图的背景色

#### 13.6 补充知识1：NSUserDefaultsController

#### 13.7 补充知识2：从命令行读写Defaults

### 13.8 挑战

### 第14章 使用Notifications

#### 14.1 Notifications 是什么

#### 14.2 Notifications不是什么

#### 14.3 NSNotification

#### 14.4 NSNotificationCenter

#### 14.5 发布notification

#### 14.6 注册成为观察者

#### 14.7 当notification到达时怎么处理

#### 14.8 userInfo字典

#### 14.9 补充知识：Delegates和 Notifications

### 14.10 挑战

### 第15章 使用告警窗口

#### 15.1 让用户确认删除

### 15.2 挑战

### 第16章 本地化

#### 16.1 本地化NIB文件

#### 16.2 字符串表

##### 16.2.1 创建字符串表

##### 16.2.2 使用字符串表

#### 16.3 补充知识1：ibtool

#### 16.4 补充知识2：格式化字符串中符号的顺序

### 第17章 自定义视图

#### 17.1 视图层次

#### 17.2 绘制视图

##### 17.2.1 创建视图子类的实例

##### 17.2.2 Size Inspector

##### 17.2.3 drawRect

#### 17.3 使用NSBezierPath 绘图

#### 17.4 NSScrollView

#### 17.5 用程序创建视图



## &lt;&lt;苹果开发之Cocoa编程&gt;&gt;

- 17.6 补充知识1：单元格
- 17.7 补充知识2：isFlipped
- 17.8 挑战
- 第18章 图片和鼠标事件
  - 18.1 NSResponder
  - 18.2 NSEvent
  - 18.3 捕获鼠标事件
  - 18.4 使用NSOpenPanel
    - 18.4.1 修改XIB文件
    - 18.4.2 代码编辑
  - 18.5 在视图中合成图片
  - 18.6 视图的坐标系统
  - 18.7 Autoscrolling
  - 18.8 补充知识：NSImage
  - 18.9 挑战
- 第19章 键盘事件
  - 19.1 NSResponder
  - 19.2 NSEvent
  - 19.3 使用自定义视图创建一个新的工程
    - 19.3.1 界面布局
    - 19.3.2 创建连接
    - 19.3.3 编写代码
  - 19.4 补充知识：Rollovers
  - 19.5 毛边蓝框
- 第20章 绘制带属性的文本
  - 20.1 NSFont
  - 20.2 NSAttributedString
  - 20.3 绘制字符串和属性字符串
  - 20.4 显示字母
  - 20.5 让视图生成PDF数据
  - 20.6 补充知识：NSFontManager
  - 20.7 挑战1
  - 20.8 挑战2
- 第21章 粘贴板和Nil-Targeted Actions
  - 21.1 NSPasteboard
  - 21.2 在BigLetterView 中添加剪切、复制和粘贴功能
  - 21.3 Nil-Targeted Actions
  - 21.4 补充知识1：哪个对象发送了action消息
  - 21.5 补充知识2：UTI和剪贴板
  - 21.6 补充知识3：延迟复制技术
  - 21.7 挑战1
  - 21.8 挑战2
- 第22章 Categories
  - 22.1 在NSString 中添加方法
  - 22.2 补充知识：声明私有方法
- 第23章 拖放
  - 23.1 设置BigLetterView 为拖动作的源

## &lt;&lt;苹果开发之Cocoa编程&gt;&gt;

- 23.1.1 开始拖操作
- 23.1.2 释放以后
- 23.2 设置BigLetterView为拖操作的目的
  - 23.2.1 registerForDraggedTypes :
  - 23.2.2 添加高亮效果
  - 23.2.3 实现拖操作目的方法
  - 23.2.4 添加第二个的BigLetterView 对象
- 23.3 补充知识：Operation Mask
- 第24章 NSTimer
  - 24.1 界面布局
  - 24.2 建立连接
  - 24.3 为TutorController添加代码
  - 24.4 补充知识：NSRunLoop
  - 24.5 挑战
- 第25章 工作表
  - 25.1 添加sheet窗口
    - 25.1.1 添加outlet和action
    - 25.1.2 界面布局
    - 25.1.3 添加代码
  - 25.2 补充知识1：contextInfo
  - 25.3 补充知识2：模式化窗口
- 第26章 创建NSFormatters
  - 26.1 简单的格式化器类
    - 26.1.1 创建ColorFormatter.h
    - 26.1.2 编辑XIB文件
    - 26.1.3 NSColorList
    - 26.1.4 在字符串中搜索子串
    - 26.1.5 实现基本的格式化器方法
  - 26.2 NSControl类的Delegate
  - 26.3 检查部分字符串
  - 26.4 返回属性字符串的格式化器
  - 26.5 补充知识：NSValueTransformer
- 第27章 打印
  - 27.1 处理页面设置
  - 27.2 补充知识：是在屏幕上绘图吗
  - 27.3 挑战
- 第28章 Web Service
  - 28.1 RanchForecast项目
    - 28.1.1 NSURLConnection
    - 28.1.2 在ScheduleFetcher中添加XML解析功能
    - 28.1.3 界面布局
    - 28.1.4 编写控制器代码
  - 28.2 打开URL
  - 28.3 挑战：添加一个WebView
- 第29章 Blocks
  - 29.1 块语法
    - 29.1.1 块内部的内存和对象操作

## &lt;&lt;苹果开发之Cocoa编程&gt;&gt;

- 29.1.2 块的可用性
- 29.1.3 RanchForecast : 异步开发
- 29.1.4 接收异步响应
- 29.2 挑战 : 实现一个Delegate
- 第30章 iOS开发
  - 30.1 将RanchForecast移植到iOS
  - 30.2 RootViewController
  - 30.3 添加导航控制器
  - 30.4 ScheduleViewController
  - 30.5 UITableViewController
  - 30.6 视图控制器切换
  - 30.7 挑战
- 第31章 视图切换
  - 31.1 开始
    - 31.1.1 创建管理视图控制器类
    - 31.1.2 创建视图控制器及其XIB文件
  - 31.2 给MyDocument添加视图切换功能
  - 31.3 调整窗口尺寸
- 第32章 Core Data Relationships
  - 32.1 模型编辑
  - 32.2 创建自定义NSManagedObject类
    - 32.2.1 Employee
    - 32.2.2 Department
  - 32.3 界面布局
  - 32.4 事件及nextResponder
- 第33章 Core Animation
  - 33.1 Scattered
  - 33.2 隐式动画及action
  - 33.3 挑战1
  - 33.4 挑战2
- 第34章 并发性
  - 34.1 多线程
    - 34.1.1 巨大的分歧
    - 34.1.2 简单的Cocoa后台线程
  - 34.2 Scattered 升级版 : Instruments 中的Time Profiling
  - 34.3 NSOperationQueue
    - 34.3.1 Scattered的多线程解决方案
    - 34.3.2 线程的同步处理
  - 34.4 补充知识 : 更快的Scattered 程序
  - 34.5 挑战
- 第35章 Cocoa和OpenGL
  - 35.1 一个简单的Cocoa/OpenGL应用程序
  - 35.2 界面布局
  - 35.3 代码编写
- 第36章 NSTask
  - 36.1 ZIPspector
    - 36.1.1 同步读取

<<苹果开发之Cocoa编程>>

36.1.2 iPing

36.2 挑战：.tar和.tgz文件

第37章 发布应用

37.1 Build Configurations

37.2 创建Release版本

37.3 应用沙箱

37.3.1 权限

37.3.2 文件访问与权限的折中

37.4 苹果应用商店

结束语

## 章节摘录

版权页：插图：相较于复杂的NSOperation对象创建方式，这里使用了NSOperationQueue的addOperationWithBlock：，来创建NSBlockOperation对象，并将其添加到队列中。

请注意，这里代码更改还是比较少的，程序最主要的流程也没有进行改变。

一般情况下并不这样进行多线程的处理，但是可以看到，通过对域内变量的引用，锁定机制帮助开发者避免了很多复杂代码。

34.3.2线程的同步处理 在这个例子中还不需要担心race conditions，其原因在于，程序设计时就避免了与后台线程进行数据结构共享。

后台线程的工作只是进行文件夹计数、打开图片以及创建thumbnails等。

这个例子中唯一的共享数据就是Core Animation layers，但它也只是由main线程进行了修改而已。

避免了race conditions，多线程看起来就太简单了。

并不是所有的多线程都可以通过这种方法来解决，很多时候需要保护代码块（或者多个代码块），以确保某个时间点上只有一个线程能运行。

这通常通过互斥锁（互相排斥的锁定）来完成。

Objective—C提供了一个简单的方法来部署互斥锁——@synchronized指令。

@synchronized指令使用了互斥锁，这与传值对象不同。

在这个例子中，我们锁定images对象，就是一个NSMutableArray实例。

由于NSMutableArray对象并不是thread—safe（就是说该对象并不是用于多线程操作的），因此，在多线程环境中进行操作时推荐使用互斥锁。

使用@synchronized指令可以保证：当所有的@synchronized指令在一个确定的对象上操作时，只有一个线程在某个时间点上可以执行封装的代码块。

因此，如果有两个线程同时都要调用addImage：方法，第一个线程将获得锁，可以进行图片对象的添加；而第二个线程将被锁住，直到这个锁被释放。

读者可能对为什么NSMutableArray不是thread-safe感到困惑。

一个原因是互斥锁与它有负载关联，线程的安全性使得NSMutableArray对象的性能明显下降；另一个原因就是，通常情况下锁定某个代码块比进行一个单独的线程调用，更有用、也更普遍一些，这样看起来好像是将对象从一个数据结构移动到另一个数据结构。

Cocoa提供了一系列的工具用于线程同步，如NSLock和NSCondition。

对于这些工具，可以看看NSOperationQueue中的相关资料，Grand Central Dispatch在《Advanced MacOS XProgramming》中也进行了详尽的讨论。

## <<苹果开发之Cocoa编程>>

### 媒体关注与评论

Aaron的这本书是Mac OS X编程图书领域的标杆——措辞优雅、精雕细琢，是开发苹果应用的优秀图书。

—— Scott Stevenson经过几个版次的市场检验，本书已经被公认为从零开始学习Cocoa编程的首选。

Aaron是少数（很可能是唯一）全职的专业Cocoa讲师之一，他多年的教学经验都浓缩于本书中。

—— Tim Burks，资深软件工程师/ObjC编程语言创始人如果你是工作在UNIX或Windows平台上的程序员，希望为Mac OS X开发一些新应用，或者想将应用移植到Mac上，那么这本书是必备的参考书和培训资料。

—— Kevin H. Spencer，Apple认证技术协调员

## <<苹果开发之Cocoa编程>>

### 编辑推荐

《华章程序员书库:苹果开发之Cocoa编程(原书第4版)》编辑推荐：经典畅销书全新升级，全面且深入地讲解了Cocoa编程的各项知识，被誉为Mac OS X编程图书领域的标杆，是从零开始系统学习Cocoa编程的首选！

## <<苹果开发之Cocoa编程>>

### 名人推荐

Aaron的这本书是Mac OS X编程图书领域的标杆——措辞优雅、精雕细琢，是开发苹果应用的优秀图书。

——Scott Stevenson，the ocacao网经过几个版次的市场检验，本书已经被公认为从零开始学习Cocoa编程的首选。

Aaron是少数（很可能是唯一）全职的专业Cocoa讲师之一，他多年的教学经验都浓缩于本书中。

——Tim Burks，资深软件工程师/ObjC编程语言创始人如果你是工作在UNIX或Windows平台上的程序员，希望为Mac OS X开发一些新应用，或者想将应用移植到Mac上，那么这本书是必备的参考书和培训资料。

——Kevin H. Spencer，Apple认证技术协调员



版权说明

本站所提供下载的PDF图书仅提供预览和简介, 请支持正版图书。

更多资源请访问:<http://www.tushu007.com>