

<<学习iPhone编程>>

图书基本信息

书名：<<学习iPhone编程>>

13位ISBN编号：9787564124113

10位ISBN编号：7564124113

出版时间：2010-10

出版时间：东南大学出版社

作者：艾伦

页数：356

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>

<<学习iPhone编程>>

内容概要

获取为iPhone和iPod Touch编程所需的实战经验。

有了这本易学易用的指南，你将通过学习如何运用Xcode、Objective—C编程语言与核心框架，构建多个样例应用程序。

在你掌握它之前，你将不仅拥有开发应用的技能，还会知道如何顺利完成提交应用到iTunesApp store的流程。

无论你是一位Mac编程新手还是摩拳擦掌准备上手iPhone和iPodTouch的Mac开发行家，《学习iPhone编程》都将为你构建面向市场的iPhone应用开一个好头。

<<学习iPhone编程>>

作者简介

Alasdair Allan是一位Exeter大学资深研究员，他负责构建P2P望远镜网络。他还忙于制作开源硬件，还有一系列用于管理云服务和监控分布式传感器网络的iPhone应用。

书籍目录

Preface 1. Why Go Native? The Pros and Cons Why Write Native Applications? The Release Cycle Build It and They Will Come 2. Becoming a Developer Registering As an iPhone Developer Enrolling in the iPhone Developer Program The Apple Developer Connection Installing the iPhone SDK Preparing Your iPhone or iPod touch Creating a Development Certificate Getting the UDID of Your Development Device Creating an App ID Creating a Mobile Provisioning Profile Making Your Device Available for Development 3. Your First iPhone App'. Objective-C Basics Object-Oriented Programming The Objective-C Object Model The Basics of Objective-C Syntax Creating a Project Exploring the Project in Xcode Our Project in Interface Builder Adding Code Connecting the Outlets in Interface Builder Putting the Application on Your iPhone4. Coding in Objective-C Declaring and Defining Classes Declaring a Class with the Interface Defining a Class with the Implementation Object Typing Properties Synthesizing Properties The Dot Syntax Declaring Methods Calling Methods Calling Methods on nil Memory Management Creating Objects The Autorelease Pool The alloc, retain, copy, and release Cycle The dealloc Method Responding to Memory Warnings Fundamental iPhone Design Patterns The Model-View-Controller Pattern Views and View Controllers The Delegates and DataSource Pattern Conclusion5. Table-View-Based Applications Simplifying the Template Classes Creating a Table View Organizing and Navigating Your Source Code Connecting the Outlets Building a Model Adding Images to Your Projects Connecting the Controller to the Model Mocking Up Functionality with Alert Windows Adding Navigation Controls to the Application Adding a City View Edit Mode Deleting a City Entry Adding a City Entry The "Add New City..." Interface Capturing the City Data6. Other View Controllers Utility Applications Making the Battery Monitoring Application Tab Bar Applications Refactoring the Template Adding Another Tab Bar Item Finishing Up Modal View Controllers Modifying the City Guide Application The Image Picker View Controller Adding the Image Picker to the City Guide Application7. Connecting to the Network Detecting Network Status Apple's Reachability Class Embedding a Web Browser in Your App A Simple Web View Controller Displaying Static HTML Files Getting Data Out of a UIWebView Sending Email Getting Data from the Internet Synchronous Requests Asynchronous Requests Using Web Services8. Handling Data Data Entry UITextField and Its Delegate UITextView and Its Delegate Parsing XML Parsing XML with libxml2 Parsing XML with NSXMLParser Parsing JSON The Twitter Search Service The Twitter Trends Application Regular Expressions Introduction to Regular Expressions Storing Data Using Flat Files Storing Information in an SQL Database Core Data9. Distributing Your Application Adding Missing Features Adding an Icon Adding a Launch Image Changing the Display Name Enabling Rotation Building and Signing Ad Hoc Distribution Developer-to-Developer Distribution App Store Distribution Submitting to the App Store The App Store Resource Center Reasons for Rejection10. Using Sensors Hardware Support Determining Available Hardware Support Setting Required Hardware Capabilities Using the Camera The Core Location Framework Location-Dependent Weather Using the Accelerometer Writing an Accelerometer Application Using the Digital Compass Accessing the Proximity Sensor Using Vibration11. Geolocation and Mapping User Location Annotating Maps12. Integrating Your Application Application Preferences Accessing Global Preferences Custom URL Schemes Using Custom Schemes Registering Custom Schemes Media Playback Using the Address Book Interactive People Picking Programmatic People Picking13. Other Native Platforms PhoneGap Download and Installation Building a PhoneGap Project MonoTouch Download and Installation Building a MonoTouch Project14. Going Further Cocoa and Objective-C The iPhone SDK Web Applications Core Data Push Notifications In-App Purchase Core Animation Game Kit Writing Games Look and Feel Hardware Accessoriesindex

章节摘录

插图：I added a custom PNG icon for the Refresh button to the project, setting the UIButton type to Custom and the image to point at my refresh icon (you will need to drag your icon into your Xcode project before it will be available as a custom image). I resized the Refresh button to be the same size as the Info button provided by the template, setting the View Mode to "Scale to Fill" in the Attributes tab of the Inspector window. When connecting the UIButtons to the received actions—for example, when dragging the refresh View: action to the Refresh button—choose Touch Up Inside from the drop-down menu of events that Interface Builder will present to you when you make the connection. With this number of UI elements to play with, it's going to be easy to get confused. What's more, we are not going to connect all of the labels to our code, as some of them aren't going to be updated (e.g., section headers and the "Temp: ", "Humidity: ", and "Wind: " labels). So, for the elements you will connect to an IBOutlet, use the Identity Inspector's Interface Builder Identity section to change the Name attribute of the element to be the same as the variable in the Main View Controller interface file. Figure 7-11 shows the assignments. While this doesn't make it easier to connect the outlets to the UI elements, it does make it easier to check whether we've made an incorrect connection. If you click on File's Owner and switch to the Connections tab of the Inspector window, as Figure 7-12 shows, you can quickly check that each outlet is connected to the correct UI element since the name on each side of the connection should be the same. Although we've written the interface for the view controller and built and connected our view to the interface, we haven't implemented it yet. Let's hold off on that until we've built our data model.

<<学习iPhone编程>>

媒体关注与评论

“《学习iPhone编程》涵盖了一位有追求的程序员在iPhone OS平台上开始工作所需的一切内容。我认为它全面而且非常可读。

力荐！

” ——Fraser Speirs总监，Connected Flow有限公司，FlickrExport的作者

<<学习iPhone编程>>

编辑推荐

《学习iPhone编程(英文)(影印版)》：立即开始使用Xcode，并且学习如何操作Interface Builder，利用模型-视图-控制器（MVC）架构和Objective-c，构建一个数据条目界面，学习如何解析和存储所收到的数据，在构建大量富有挑战的样例应用时，解决一些典型的问题，理解App Store和ad hoc发布的要求及相关细节。

使用iPhone的加速表、临近传感器、GPS、数字罗盘和摄像头，将你的应用iPhone偏好设置、媒体回放等功能集成。

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>