



IOS的APP开发

2017/4/8

丁碧云

CONTENTS

- 1、开发环境
- 2、开发流程
- 3、APP设计需求
- 4、进展
- 5、参考资料

1. 开发环境

1.1 iOS系统架构

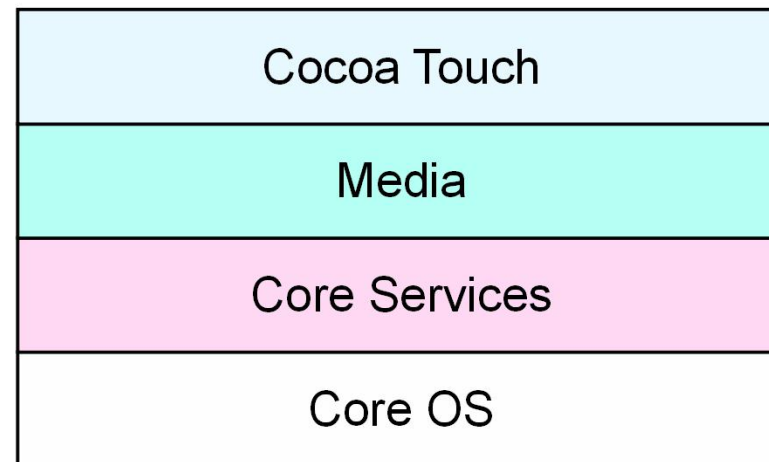
1.2 开发语言

1.3 开发环境

1.1 iOS系统架构

iOS的系统框架分为四个层次：

- 核心操作系统层（Core OS）
- 核心服务层（Core Services）
- 媒体层（Media）
- 可触摸层（Cocoa Touch）



1.2 开发语言

1) Object-C

根据C语言所衍生出来的语言，继承了C语言的特性，是扩充C的面向对象编程语言。完全兼容C语言，后缀为.m类型。

2) swift

苹果于2014年WWDC（苹果开发者大会）发布的新开发语言，可与Objective-C*共同运行于Mac OS和iOS平台，用于搭建基于苹果平台的应用程序。

1.2 开发语言（2）——swift



- Swift在天朝译为“雨燕”，右上角的图标就是它的LOGO
- 跟Objective-C一样，可以用于开发iOS、Mac应用程序
- 苹果公司从2010年7月开始设计Swift语言，耗时4年打造

Swift是一门完全面向对象的语言，它抛弃了C/C++的历史包袱，就是基本的类型如char、Int、Long等类型都是一个结构体对象，引入在Java、C++、Python中使用的操作符重载、泛型、名字空间、闭包等特性。

本质上讲：Swift语言是对OC语言的优雅包装，它的底层还是使用Cocoa Touch，Foundation框架，只是在语言层面上让开发变得更简单了。



Swift的首席架构师：
克里斯·拉特纳（Chris Lattner）

1.2 开发语言（2）——swift的面向对象

概念

面向对象（OOP）是现代流行的程序设计方法，是一种主流的程序设计规范。其基本思想是使用对象、类、继承、属性、方法等基本概念来进行程序设计，从现实世界中客观存在的事物出发来构建软件系统，并且在系统构造中尽可能运用人类的自然思维。

基本特性

封装性：就是尽可能隐蔽对象的内部细节，对外界形成一个边界，只保留有限的对外接口使之与外界发生联系。

继承性：一些特殊的类能够具有一般类的全部属性和方法，这称作特殊类对一般类的继承。

多态性：对象的多态性是指父类中定义的属性或方法被子类继承后，可以使同一个属性或方法在父类及其各子类中具有不同的含义，这称为多态性。

1.2 开发语言（2）——swift的面向对象类型

swift语言中的面向对象类型包括：类（class）、结构体（struct）和枚举（enum）。

面向对象类型	特性	语法格式	数据类型	类型成员	实例后
类	封装性 继承性 多态性	[修饰符] ¹⁾ class 类名 {定义类的成员}	引用类型 ⁴⁾	属性（存储属性 ²⁾ 、计算属性）、方法、下标、构造器和嵌套类型	对象
结构体	封装性	[修饰符] struct 结构名 {定义结构体的成员}	值类型 ⁵⁾	属性（计算属性 ³⁾ ）、方法、下标、构造器和嵌套类型	实例
枚举	封装性	[修饰符] enum 枚举名 {枚举的定义}	值类型	属性（存储属性、计算属性）、方法、下标、构造器和嵌套类型	实例

1.2 开发语言（3）——Swift与OC比较

苹果公司声称新的Swift语言拥有快速、现代、安全、互动的新特性，并且性能全部优于OC。

- 根据WWDC的展示
 - 在进行复杂对象排序时
 - ✓ Objective-C的性能是Python的2.8倍，Swift的性能是Python的3.9倍
 - 在实现 RC4加密算法时
 - ✓ Objective-C的的性能是Python的127倍，Swift的性能是Python的220倍
- 有持怀疑态度的国外程序员，也对Objective-C和Swift的性能进行了测试
 - <http://www.splasmata.com/?p=2798>

swift与OC语言在编程时的区别

1. 不像 C语言 和 O-C语言 一样都必须有一个主函数 `main ()` 作为程序入口，swift程序从第一句开始向下顺序执行。一直到最后。
2. 每个语句结束后可以不加分号，但是多条语句不能写在同一行
3. 在声明常量或者变量的时候直接初始化可以省略其类型，否则需要在变量名称后跟冒号加类型。实际开发中建议全都加上，以免出现错误。

1.3 开发环境

软件环境：

- 1) 运行环境：iOS（苹果公司开发的移动操作系统）；
- 2) 开发环境：mac OS Sierra 10.12.1（swift语言开发最低要求为Mac OS X10.9.3版本以上）；
- 3) 开发工具：Xcode 7.3版本（要求Xcode 6.0以上）；

硬件环境：

- 1) Mac苹果电脑（MacBook Air）

2. 开发流程

2.1 开发者证书

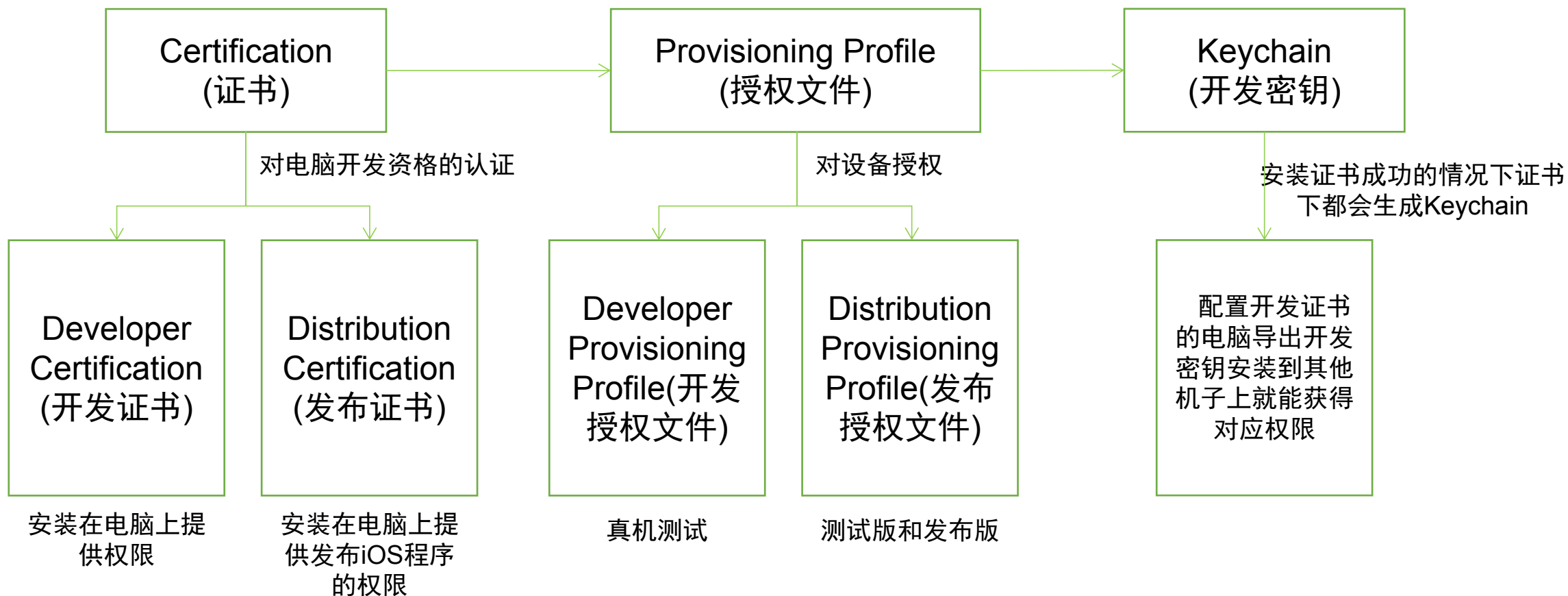
2.2 Cocoa框架

2.3 第三方库

2.4 设计架构

2.5 开发流程

2.1 开发者证书



2.2 Cocoa框架

Cocoa是创建Mac OS X和iOS程序的原生面向对象API，为这两者应用提供了编程环境。通常称为“Cocoa框架”，事实上Cocoa本身是一个框架的集合，它包含了众多子框架

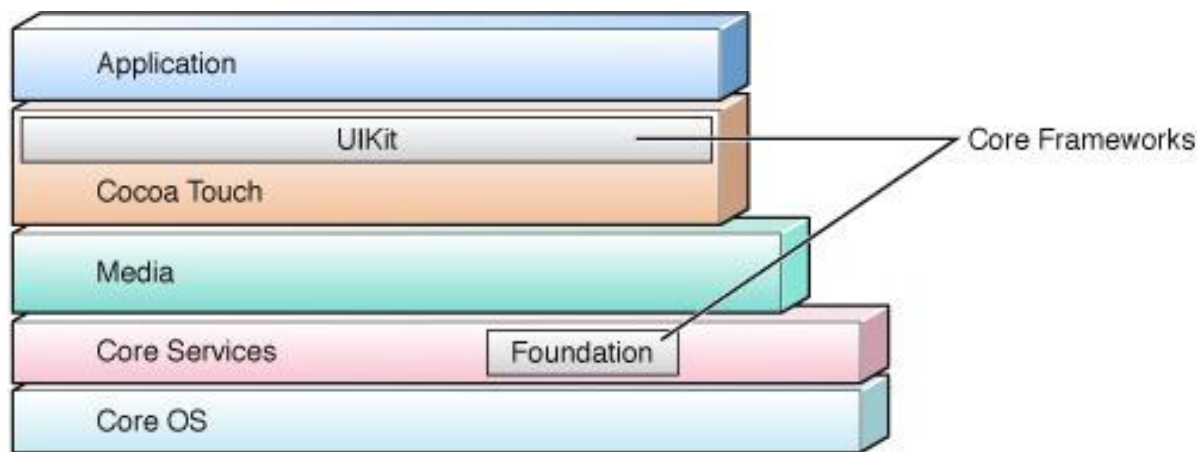
Cocoa中最核心的有两个：

(1) Foundation框架；

Foundation框架包含所有和界面显示无关的类。

(2) Application Kit (AppKit) 框架 (Cocoa Touch中叫UIKit框架)。

Application Kit 框架包含实现图形的、事件驱动的用户界面需要的所有对象:窗口、对话框、按键、菜单、滚动条、文本输入框----这个列表还在不断增加。



2.2 Cocoa框架（1）——Foundation框架

Foundation框架为所有的应用程序提供基本系统服务

使用Foundation可以:

- * 创建和管理集合，比如数组和字典
- * 访问存储在应用程序里的图片和其它资源
- * 创建和管理字符串
- * 提交和接收通知
- * 创建日期和时间对象
- * 自动发现IP网络上的设备
- * 操作URL流
- * 执行异步代码

2.2 Cocoa框架（2）——UIKit框架

UIKit框架提供创建基于触摸用户界面的类

所有的iOS应用程序都基于UIKit，你不能是应用程序脱离这个框架。UIKit提供了在屏幕上绘制的机制，捕获事件，和创建通用用户界面元素。UIKit也通过管理显示在屏幕上的组件来组织复杂的项目。

使用UIKit可以：

- * 构建和管理你的用户界面
- * 捕获触摸和基于移动的事件
- * 呈现文字和web内容
- * 优化你的多任务程序
- * 创建定制的用户界面元素

2.3 第三方库

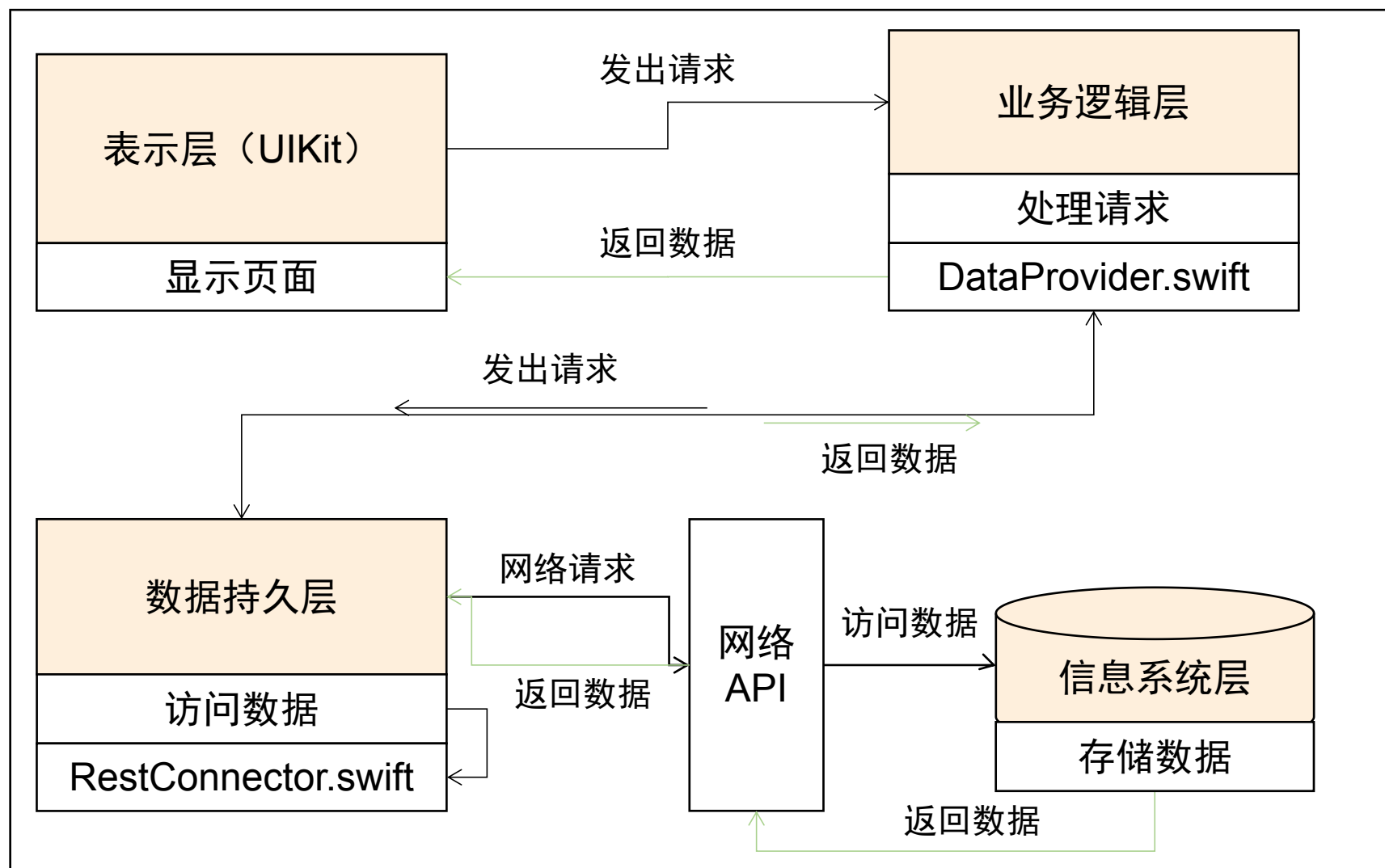
除了需要核心框架，在APP开发过程中还会用到一些辅助框架，如Alamofire（网络请求）、JLToast（动态弹出框）等。

在开发项目中，需要使用第三方库，Cocoapods是最有影响力的OS X 和iOS项目依赖管理工具，于2011年发布，经过多年的发展，他已经非常完善了。Cocoapods支持项目中采用Object-C或者swift语言。

Cocoapods会将第三方库的源代码编译为静态链接库.a文件或者是动态框架.framework文件的形式，并将它们添加到项目中，建立依赖关系。

CocoaPods被建立在Ruby上，而自从OS X 10.7之后的Mac OS X版本带有Ruby，这样使用起来非常方便。

2.4 设计架构



软件架构设计的原则是：可复用性和可扩展性。

2.5 APP开发流程(OC)

MainStoryboard.storyboard文件



从iOS5开始，storyboard文件用于描述软件界面，作为程序入口，选定view->object，从下面的框中选择需要的控件，选定控件后用来设置其属性。

绑定控件与方法（属性），以实现
对控件的操作

通过点击Editor的中间件，
将控件拉线连接到控制
的方法上，实现绑定。

UIViewController.h文件



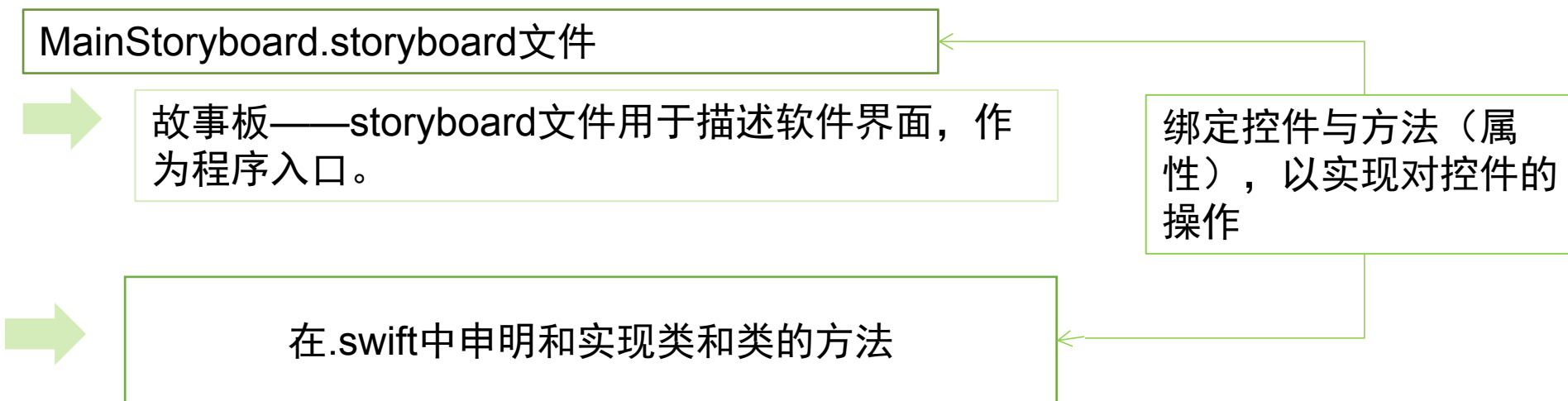
在这个文件只能声明方法（监听按钮或是其触发事件），声明控件属性

UIViewController.m文件



在这个文件中，写出方法的具体操作

2.5 APP开发流程(swift)



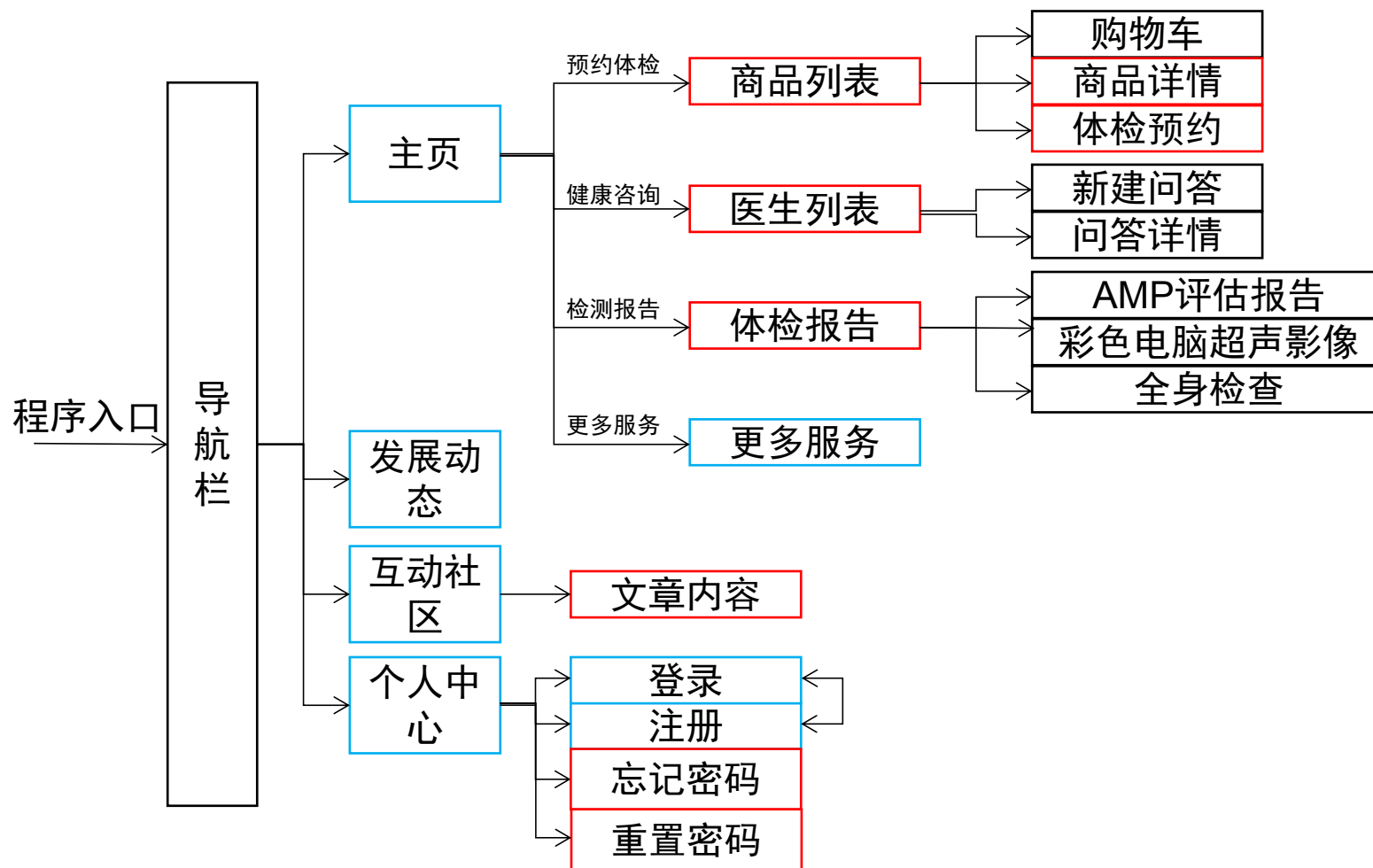
3. APP设计需求

参考原来APP进行修改和重构

3.1 原来的架构

3.2 根据需求设定的架构

4. 进展



备注：彩色框表示已设计的界面

5. 参考资料

- 1) 《iOS开发指南 从Hello World到App Store上架》（第4版）. 关东升 著；
- 2) 《从零开始学Swift》. 关东升 著；
- 3) 《swift开发实战 权威指南》 欧阳坚等 著；
- 4) 《疯狂Swift》 李阳 著。
- 5) 《iOS成长之路》



THANKS

