

实训学习总结

电子科技大学 计算机学院研一 周思成

一、 OC 基础学习

《OC 程序设计》一书作为基础，在已有其他程序语言的情况下，熟悉这门语言的使用不算太难。与其他 OOP 语言（或有 OOP 思想的类 C 语言，cpp）有异曲同工之妙。例如：

1. 函数调用会用“[]”，与其他 oop 的“.”和“->”有些许差异；
2. OC 中的指针“*”与 cpp 和 c 中的概念有差异，OC 中用这个符号类似引用和 oop 中的对象引用。
3. 其中的单继承，协议等思想看似是新的，其实也是 oop 语言的共有之处，不过换了名字，包括“块”之类的也在比较新的程序语言中也有类似特性。

二、 iOS 开发学习

iOS 开发可跟着《OC 程序设计》一书后面部分作为入门，之后在跟着极客帮的朱德权老师讲授的课程学习，可以基本学习到一整套 iOS 开发流程和范式，很有用；但是这次由于研究生假期时间的“不可预估性”，我也没有完成整个课后作业的开发流程，也没有完全看完这个课，但之后我会在其余时间完成这个课程学习和开发。但这次实训也有很多收获：

1. iOS 的 UI 设计与 Qt 中的设计思想类似，看布局文件也有一股 Qt 味道。但是与 Qt 的槽函数相比，在 Xcode 中使用 IDE 方便的手动连接（函数）虽然方便了，但是总感觉缺少安全感（虽然后面也知道了可以使用纯代码布局等）。
2. 在 Android 开发，IOS 开发，Qt 开发中的大概思路其实很像，MVC 模式 +delegate，这一部分是这些开发的精髓和难点，也需要很多训练和积累才能进步。

三、 iOS-Cocoapods 的学习

Cocoapods 是 iOS 依赖管理工具，例如 Java 语言的 Maven，Gradle 之类的，提供了很多开源库和私有库的管理方法。在构建大型工程的时候难免会用到其他开源库，例如在这个课后项目中的 AFNetworking 库等。

这个依赖管理工具促使开源工具的更广泛应用以及给 iOS 包管理提供很大

的便利，但以我在其他编程语言上对于依赖管理工具使用情况来看，cocoapods 虽然提供便利，但之后的版本管理和依赖冲突解决等也可以是个问题。

四、 设计与解决

首先由于个人的原因，的确没有完成这次课后小作业，开始的时候以为很简单，想多设计点细节和内容，结果真正做的时候工作量也不会太小，再加上研究生假期要做很多事，没有办法按时完成实训，确实是我的问题，向 ByteDance 的老教师们道歉。但在已有工作过程的还是有所收获：

1. 本来设计的 Tabbar 和 NavagationBar 的透明和半透明(毛玻璃)效果，由于 Xcode 和 iOS 版本的不断更新，很多以前的设计代码在新版本上需要修改和调试，这时候就要去多去吸收其他人的设计思想以及在简书，csdn，StackOverflow 等网站到处搜寻。
2. 使用 AFNetworking 来请求网站 API 来获取服务以及反序列化等工作也是借鉴了很多人的代码和范式。
3. (未完成)TableView 的设计本来是想在 Cell 嵌套 TableView 来完成表格中其他表格的滑动设计，但在实现的时候发现在数据管理的时候有点难，在加上这个假期导师的项目比较 push 就没有完成 table 的展示设计，但这个点我也可能需要更多参考别人的方案设计。

五、 经验之谈

这次实训的收获还是挺多的，从 OC 入门到慢慢掌握 iOS 开发流程，以及更大项目的依赖管理等内容，除了有些遗憾没有完成整个项目的课后小作业，在这里我也有些体会和理解：

1. 对于大型共享项目来说，Git 操作其实挺重要的；就拿这次来说，很多同学之间在 main 上 push 和 pull 其实是并不好的，因为我看到我个人的文件夹在后期就被其他同学的 push 所覆盖，把我的之后的改动给手动 rollback 了；我还是建议大家还是得新建一个自己的 branch，在自己的 branch 上修改，提交过后再 merge，这是一种比较合理的方案，也是常用方案。
2. iOS 开发要多去学习别人的布局解决方案，也要再参考别人代码过程中思考适合自己的解决方案，而且 iOS 更新的速度也挺快的，也要在新版

本的特性和老版本解决方案中取舍得到自己所需要的方案。