

下午内容回顾

一.闭包

1.Block的回顾

2.闭包的基本使用

- 闭包类型的写法: (参数列表) -> (返回值类型)

3.Block的循环引用

- 问题: 有没有形成循环引用
- 需求: 如何可以形成循环引用 httpTool类型定义block
- 解决: `__weak ViewController *weakSelf = self;`

4.闭包的循环引用

- 问题: 有没有形成循环引用
- 需求: 如何可以形成循环引用 httpTool类型定义闭包
- 解决: 1. `weak var weakSelf = self` 2. `{[weak self]` 3. `{[unowned self]`

5.尾随闭包

- 如果闭包是方法的最后一个参数,可以写成尾随闭包 `() {}`

二.懒加载

- lazy
- 用到时再加载
- 多次使用只会加载一次

三.常见注释

1.单行注释 //

2.多行注释 `/**/` 可以嵌套使用

3.文档注释: `///`

4.分组注释: `// MARK:-`

四.访问权限

- internal : 当前模块中都可以访问
- private : 在同一个源文件中可以访问
- public : 可以跨模块访问

五.异常处理

1.异常处理的三种方案

- try : `do {} catch {}`
- try? : 返回值是可选类型.
- try! : 没有异常. 注意: 有异常,则崩溃

2.自定义自己的异常

- throws : 表示该方法会抛出异常
- enum Type : ErrorType {case 异常一 case 异常二}
- throw 异常

六.OC和Swift相互调用

1.swift调用OC

- 创建.h的桥接文件
- 将需要用到的OC类.h导入到桥接文件中
- 配置桥接文件: 项目 -> buildSettings -> bridging -> 配置.h文件

2.OC调用Swift

- 项目名称不能乱起
- 导入“项目名称-Swift.h”
- Swift中类/属性/方法前需要加上public

七.tableView初体验

- 代码规范