

冰凌天 Lv2

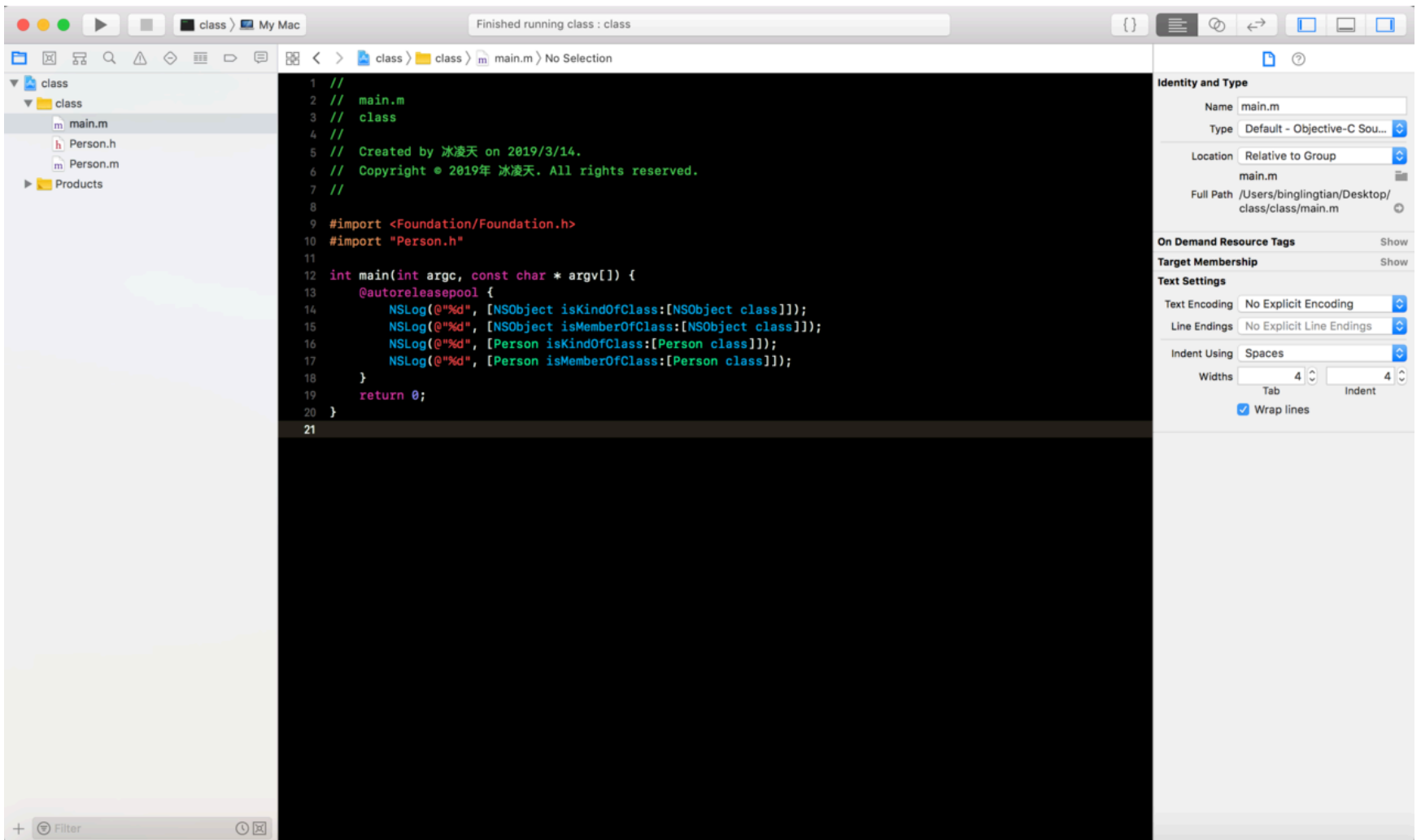
2019年03月14日 阅读 52

[关注](#)

小码哥iOS学习笔记第十五天: isKindOfClass:和isMemberOfClass:

一、面试题

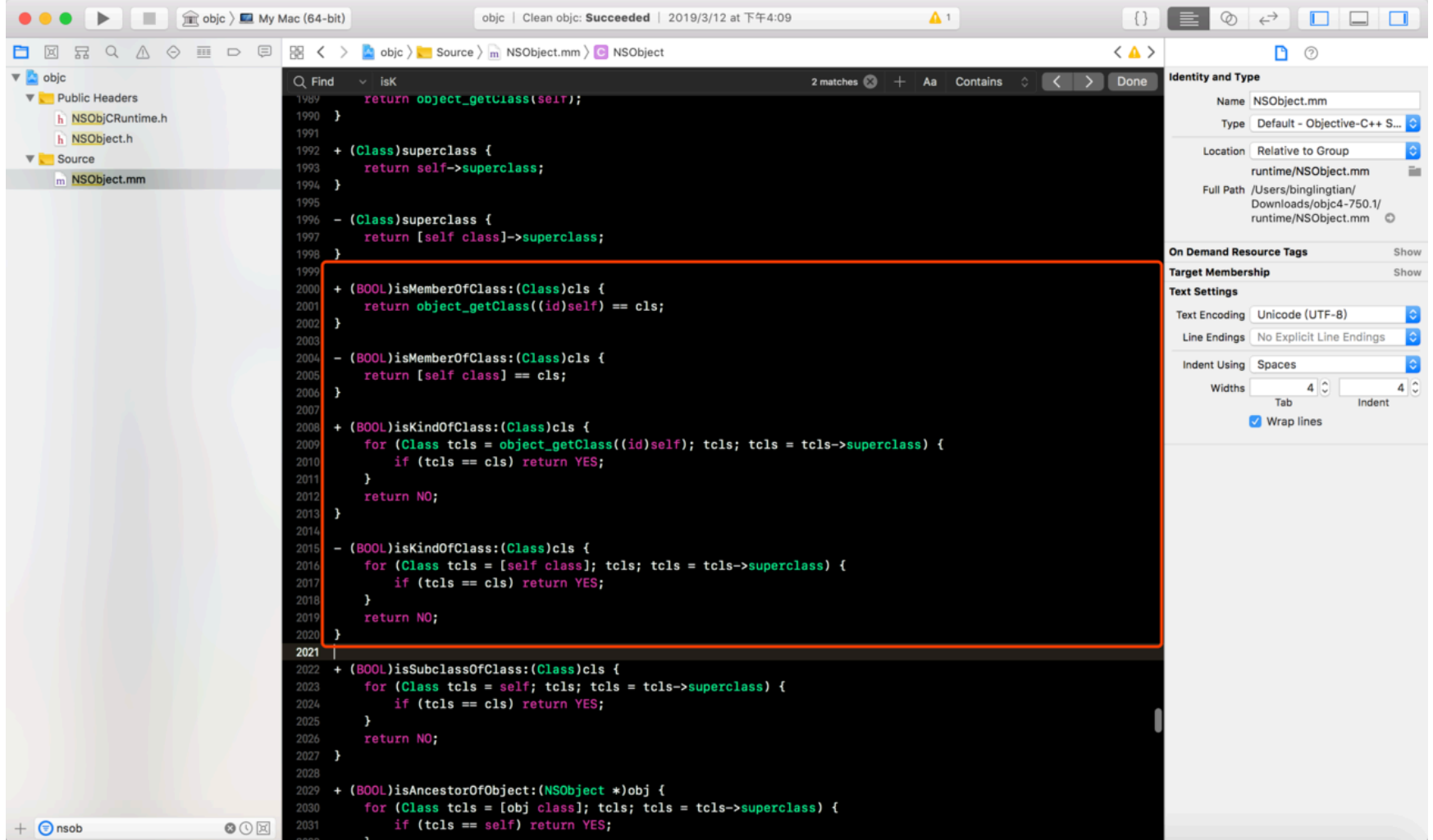
- 面试题: `Person` 继承自 `NSObject`, `main` 函数中代码如下



- 此时打印是什么? 为什么是这种打印?

二、isKindOfClass:和isMemberOfClass:的底层源码

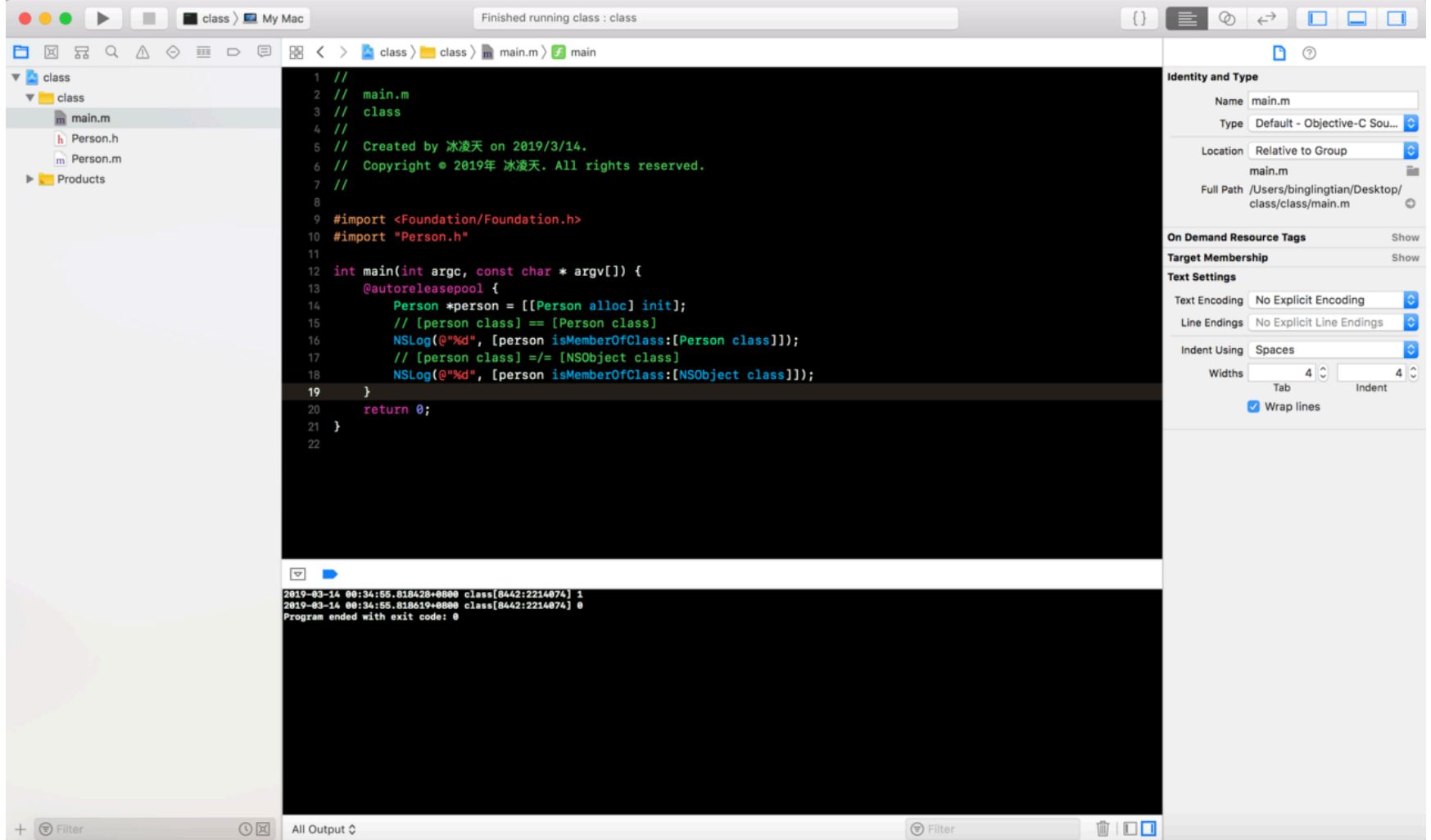
- 关于 `isKindOfClass:` 和 `isMemberOfClass:` 两个方法, 底层代码是开源的



1、- (BOOL)isMemberOfClass:(Class)cls

```
- (BOOL)isMemberOfClass:(Class)cls {
    return [self class] == cls;
}
```

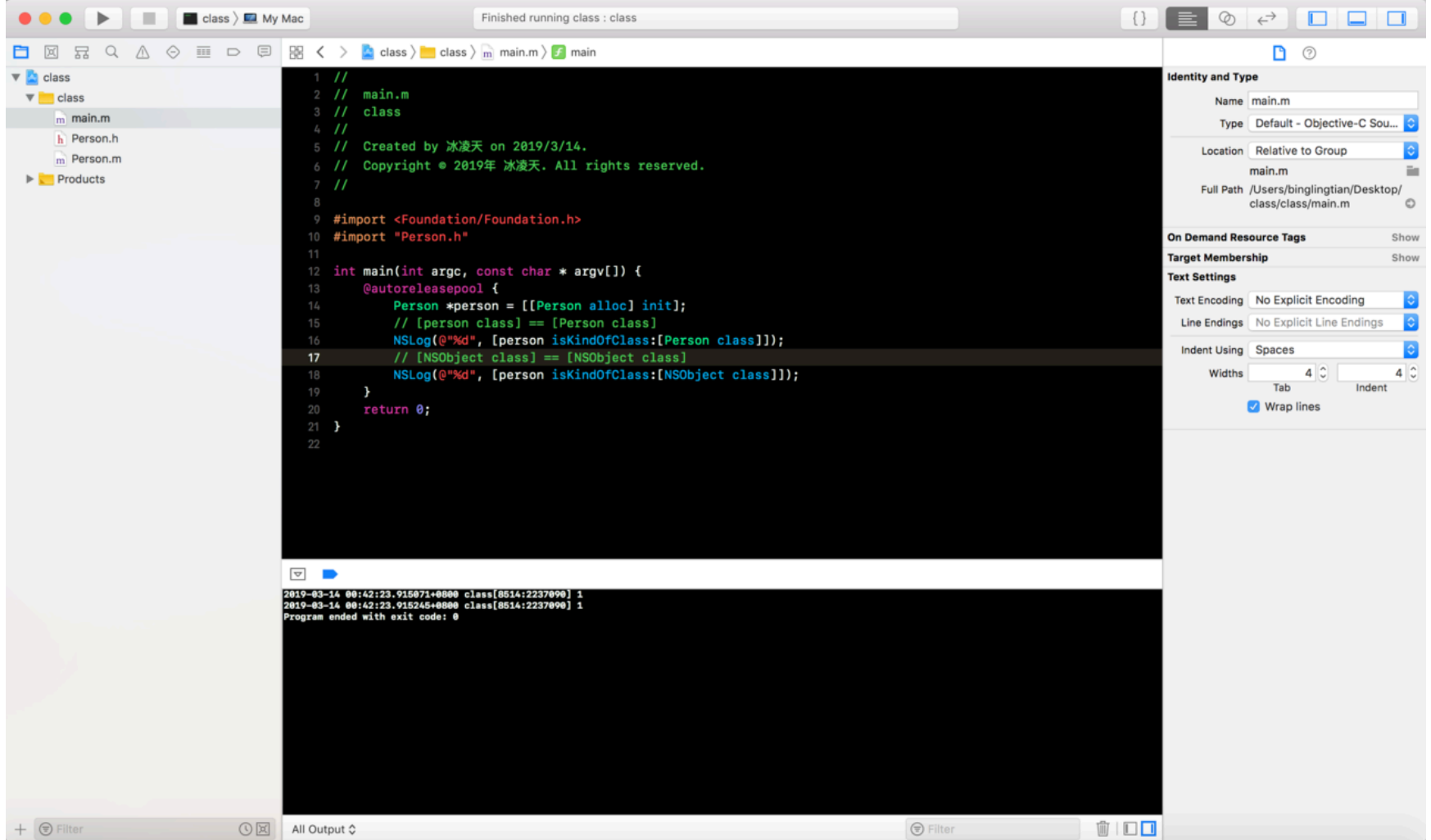
- 可以看到方法的实现中, 是将 `self` 的类方法, 与传入的 `Class` 进行对比, 如果相同返回 `YES` , 否则返回NO



2、- (BOOL)isKindOfClass:(Class)cls

```
- (BOOL)isKindOfClass:(Class)cls {  
    for (Class tcls = [self class]; tcls; tcls = tcls->superclass) {  
        if (tcls == cls) return YES;  
    }  
    return NO;  
}
```

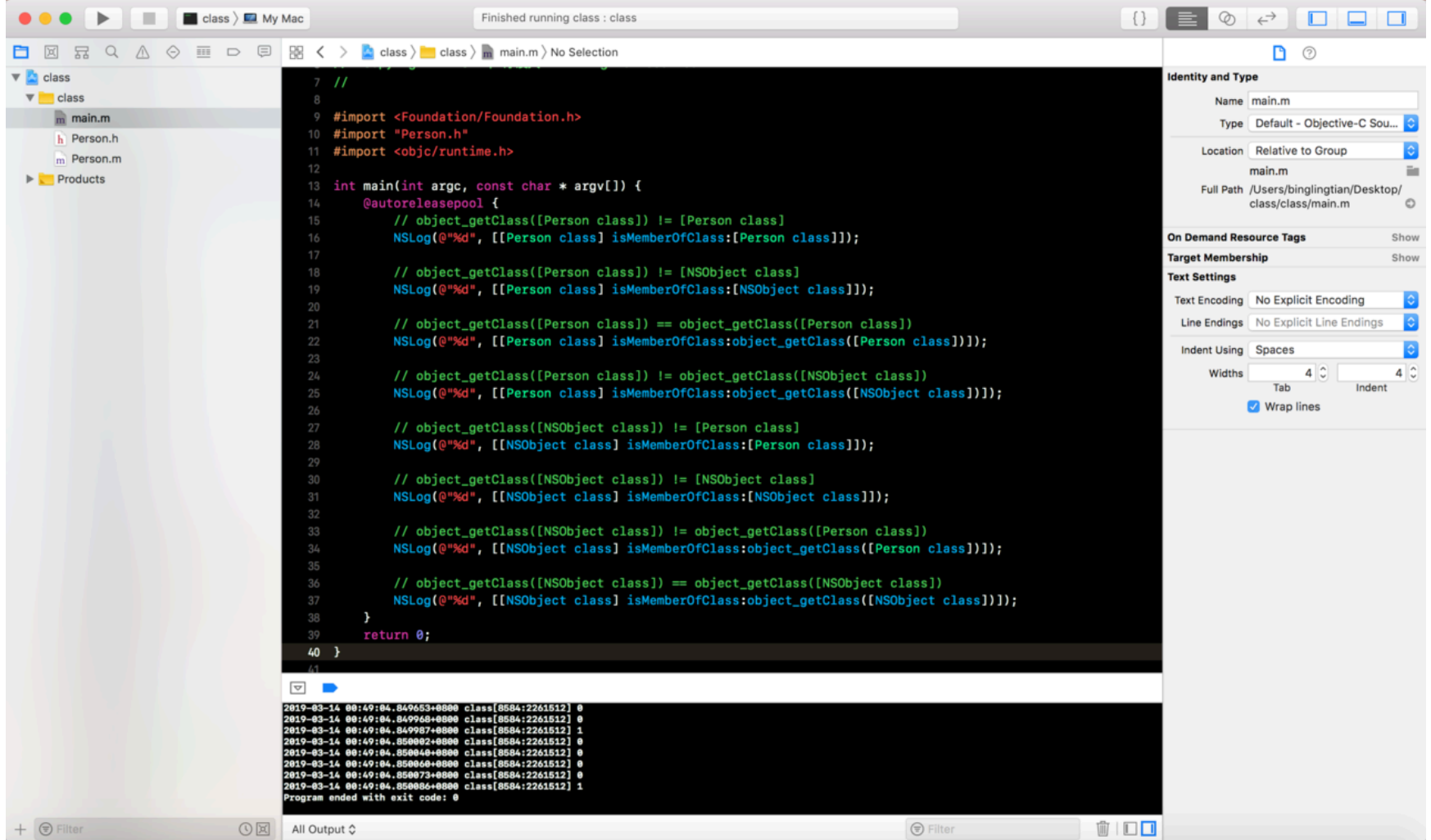
- 可以看到方法的实现中, 是先将 `[self class]` 与传入 `cls` 对比,
 - 如果相同返回YES
 - 如果不同就会取出 `[[self class] superclass]` 与 `cls` 对比
- 只要这个过程中有相同的情况就会返回 YES , 如果最后也没有匹配成功就会返回 NO



3、+ (BOOL)isMemberOfClass:(Class)cls

```
+ (BOOL)isMemberOfClass:(Class)cls {  
    return object_getClass((id)self) == cls;  
}
```

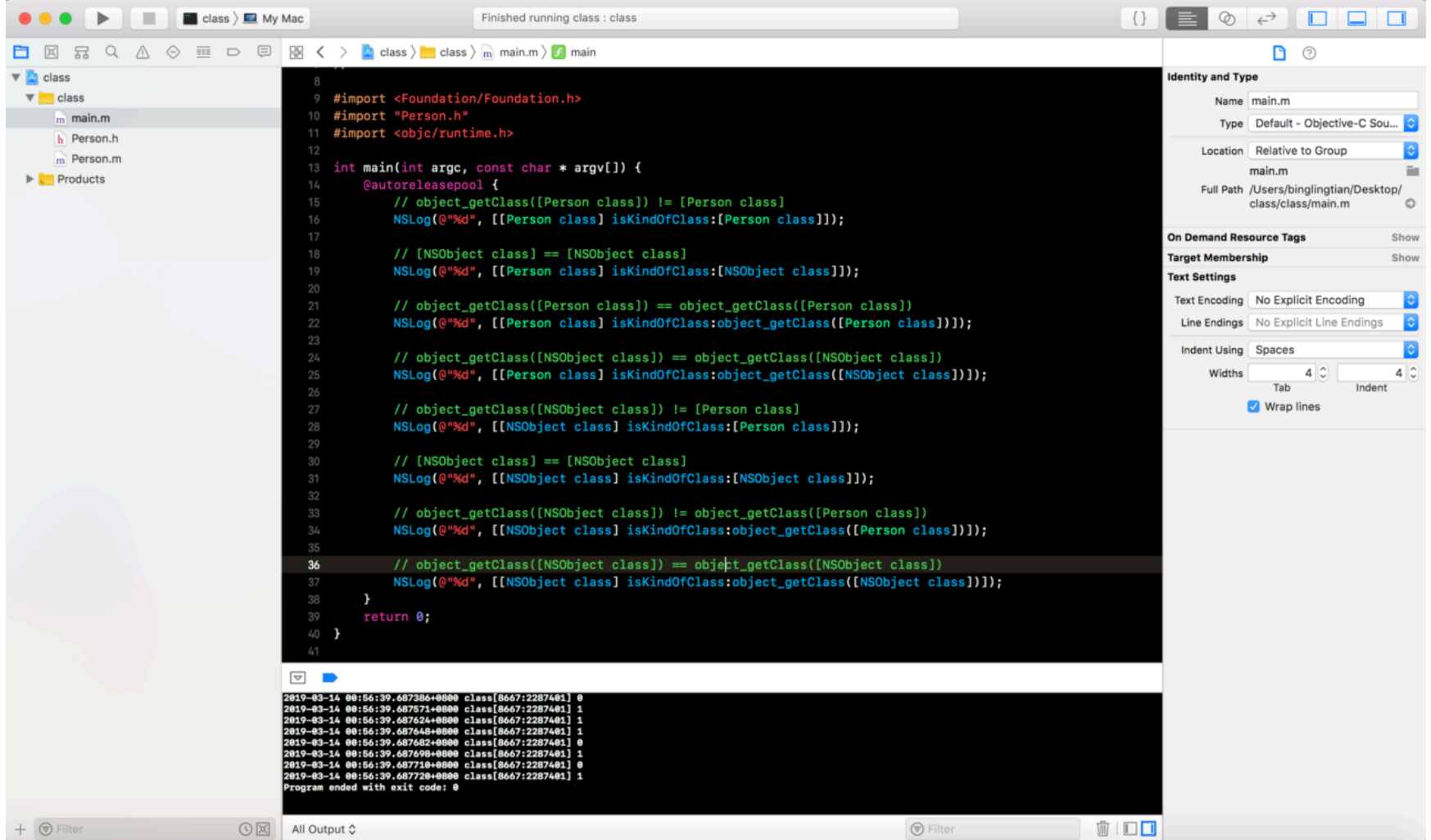
- 可以看到方法的实现中, 获取了 `self` 的 元类对象 与传入的 `cls` 进行对比, 如果相同返回 `YES`, 否则返回 `NO`



4、+ (BOOL)isKindOfClass:(Class)cls

```
+ (BOOL)isKindOfClass:(Class)cls {  
    for (Class tcls = object_getClass((id)self); tcls; tcls = tcls->superclass) {  
        if (tcls == cls) return YES;  
    }  
    return NO;  
}
```

- 可以看到方法的实现中, 使用 `self` 的 **元类对象** 与传入的 `cls` 进行对比
 - 如果相等返回 **YES**
 - 如果不相等, 就回取出 `object_getClass(self)->superclass` 与 `cls` 对比, 依次类推
- 只要这个过程中有相同的情况就会返回 **YES** , 如果最后也没有匹配成功就会返回 **NO**



- 注意: 在上面的代码中, 有两句的结果是 1

ObjectiveC

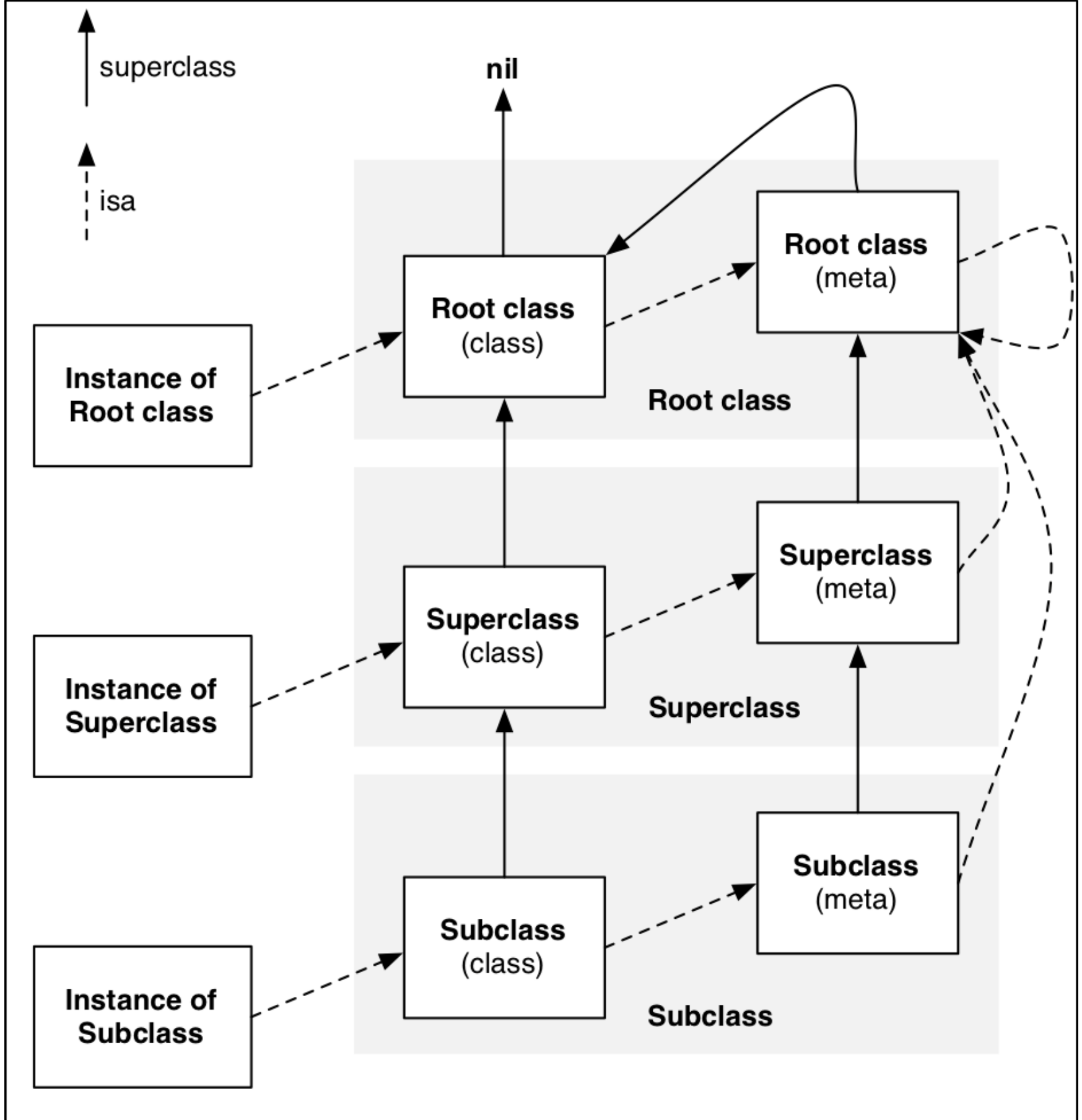
```
// [NSObject class] == [NSObject class]
NSLog(@"%d", [[Person class] isKindOfClass:[NSObject class]]);

// [NSObject class] == [NSObject class]
NSLog(@"%d", [[NSObject class] isKindOfClass:[NSObject class]]);
```

- 在这两句中, 可以看到传入的 cls 参数都是 [NSObject class]

那么为什么, 这两句代码的结果相等呢?

- 在使用 superclass 查找父类的过程中, 如果是通过 元类对象 查找, 根类 NSObject元类对象 的 superclass 是 [NSObject class]



- 所以对于任何的元类对象, 只要一直查找 元类 的 `superclass`, 最终都会找到 `[NSObject class]`, 所以上面两句代码的结果是相等的

总结:

只要是 `isKindOfClass:` 方法的 `cls` 参数是 `[NSObject class]`, 那么调用者只要是 `NSObject` 或者 `NSObject` 的子类, 那么结果都是YES

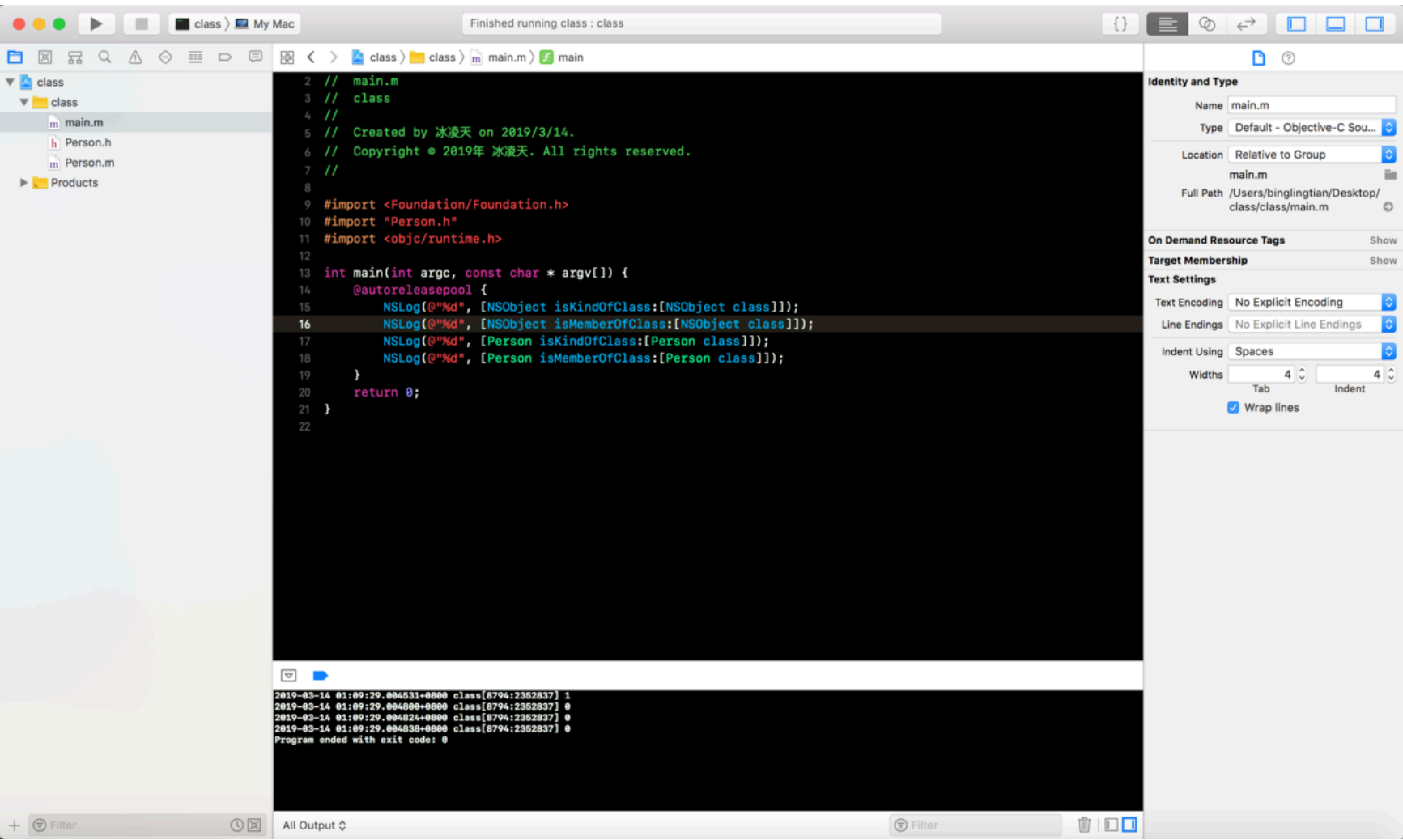
三、解答面试题


```
// 元类对象最后会通过`superclass`找到[NSObject class]，所以结果相等，结果是1
NSLog(@"%d", [NSObject isKindOfClass:[NSObject class]]);

// objc_getClass([NSObject class]) != [NSObject class] 结果是0
NSLog(@"%d", [NSObject isKindOfClass:[NSObject class]]);


// objc_getClass([Person class]) != [Person class] 结果是0
NSLog(@"%d", [Person isKindOfClass:[Person class]]);

// objc_getClass([Person class]) != [Person class] 结果是0
NSLog(@"%d", [Person isKindOfClass:[Person class]]);
```



关注下面的标签，发现更多相似文章

iOS



冰凌天

获得点赞 189 · 获得阅读 9,937

Lv2

iOS

关注

安装掘金浏览器插件

打开新标签页发现好内容，掘金、GitHub、Dribbble、ProductHunt 等站点内容轻松获取。快来安装掘

评论

输入评论...

相关推荐

[专栏](#) 知识小集 · 1天前 · iOS

值得收藏的 5 个 iOS 库

 43  4

[专栏_森宇_](#) · 4天前 · iOS

iOS 13 适配要点总结

 72  16

[专栏](#) 知识小集 · 7天前 · iOS

Xcode 11 的那些新东西

 49 

[荐](#) · 知识小集 · 10天前 · iOS

[专栏](#)

iOS 13 正式发布，来看看有哪些 API 变动

 68  7

[专栏](#) 宋卓 · 19小时前 · iOS

编程范式 --- 函数式编程（Funtional Programming，简称FP）

 1  1

[专栏](#) by在水一方 · 2天前 · iOS

iOS 端 h5 页面秒开优化实践

 16 

专栏GSC · 1天前 · iOS


iOS开发：集成的SDK不支持模拟器调试怎么办？


 3

 4

专栏iOS桃子 · 2天前 · iOS

iOS | 面试知识整理 - 数据持久化(八)

 2



专栏QiShare · 6天前 · iOS

iOS13 DarkMode适配（一）

 25

 16

专栏Henry_Jeannie · 2天前 · iOS

iOS -- 问题杂记

 16

