



冰凌天 Lv2

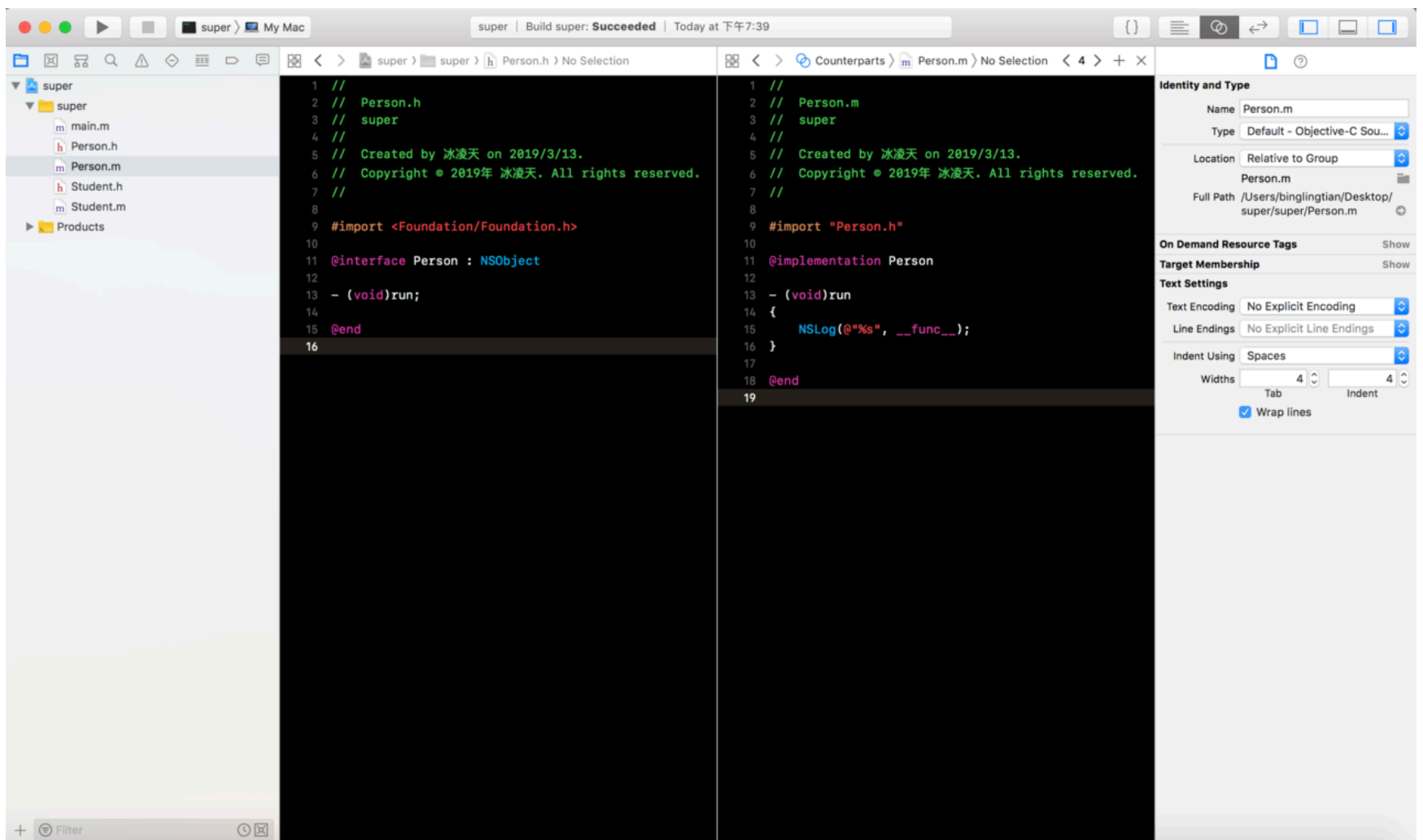
2019年03月13日 阅读 35

[关注](#)

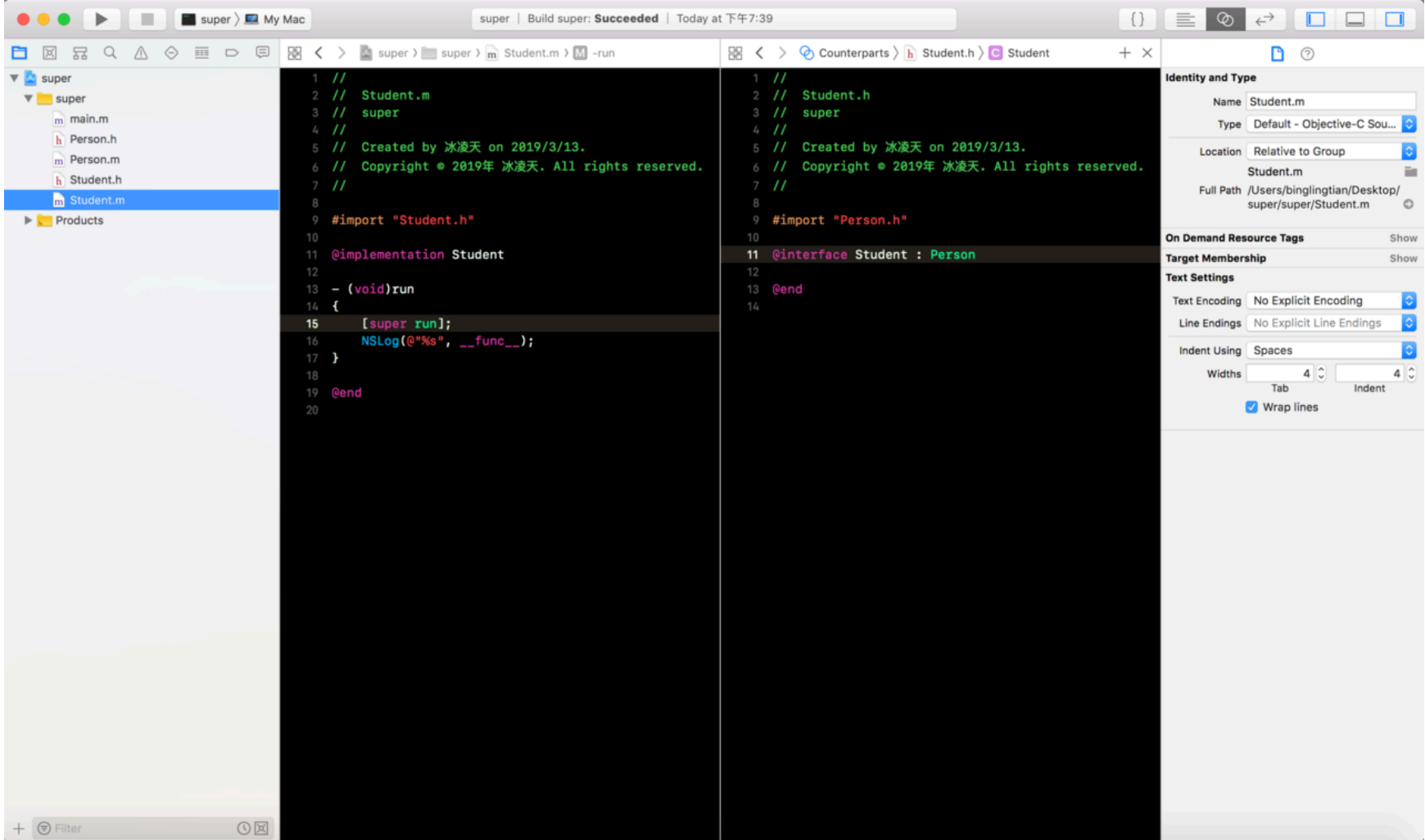
# 小码哥iOS学习笔记第十四天: super

## 一、准备代码

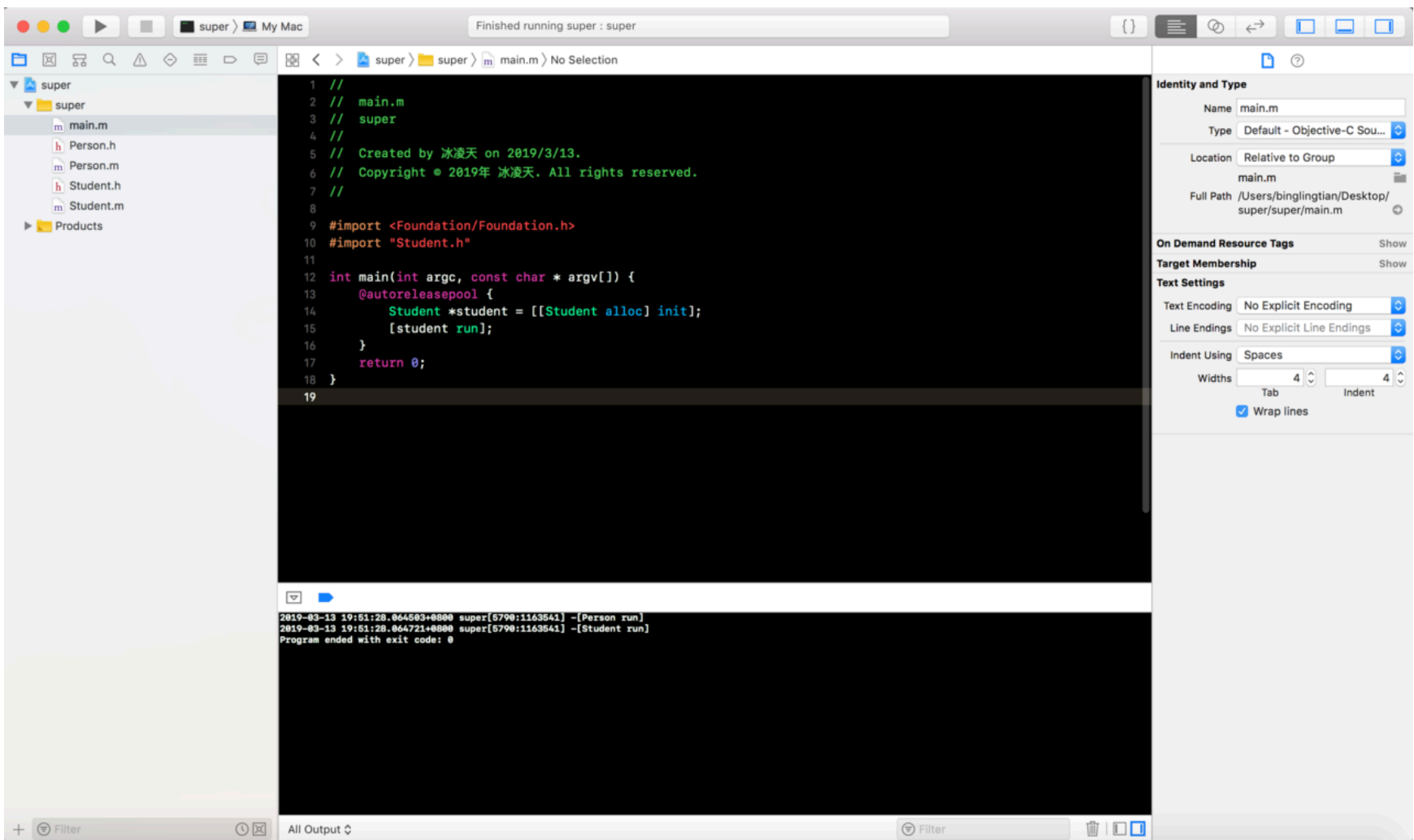
- `Person` 继承自 `NSObject` , 并实现 `-run` 方法



- `Student` 继承自 `Person` , 并重写 `-run` 方法



- 在 `main` 函数中, 执行下面的代码, 可以看到 `Person` 和 `Student` 方法中的 `run` 都有打印结果



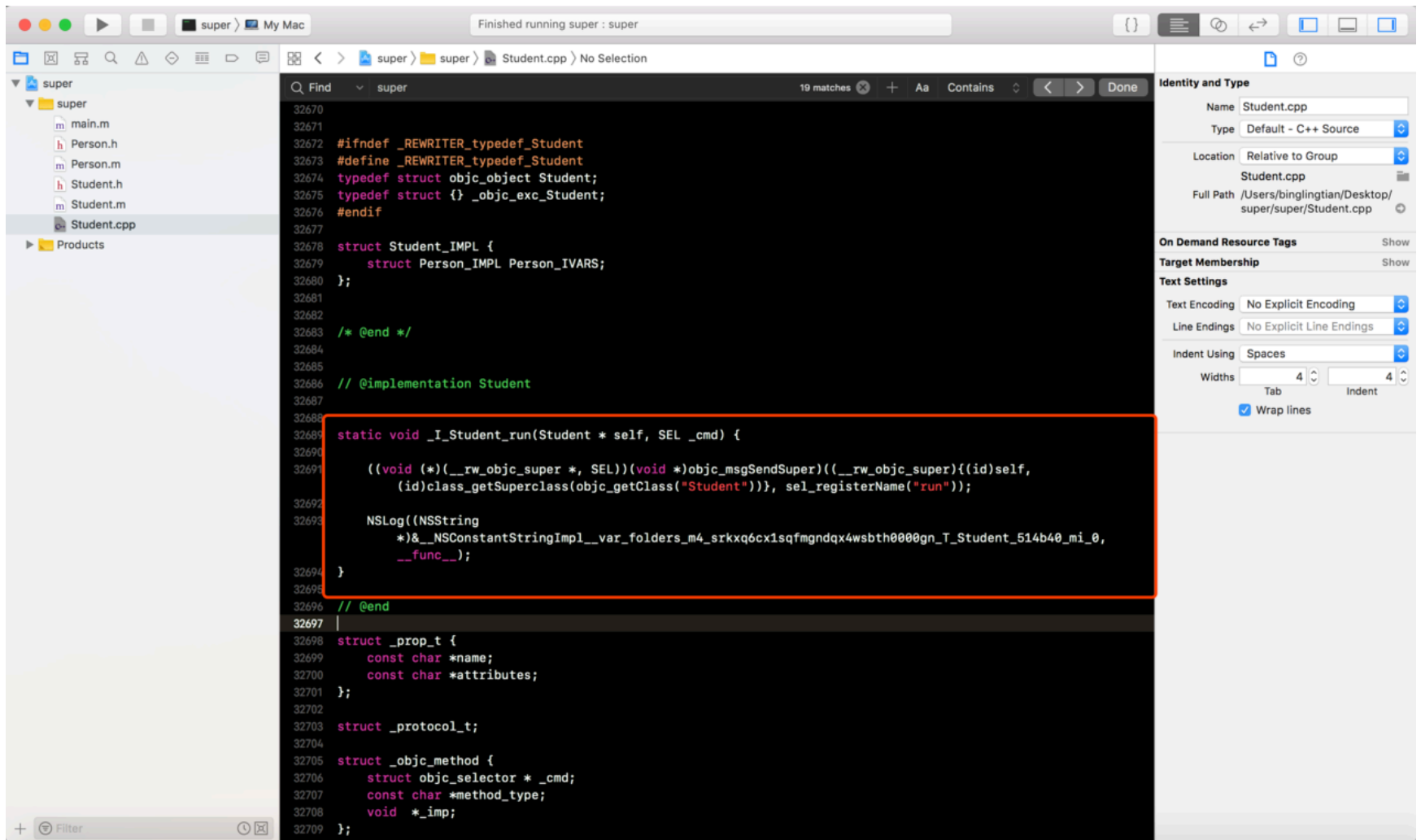
- 那么 `super` 在底层是什么样的呢?

## 二、super

- 使用终端, 执行下面的命令, 生成 `Student.cpp` 文件

```
xcrun -sdk iphoneos clang -arch arm64 -rewrite-objc Student.m
```

- 可以在 `Student.cpp` 文件中找到 `-run` 方法的底层代码



- 下面一句, 就是 `[super run];` 代码部分

```
((void (*)(__rw_objc_super *, SEL))(void *)objc_msgSendSuper)((__rw_objc_super){(id)self
```

- 去掉类型转换

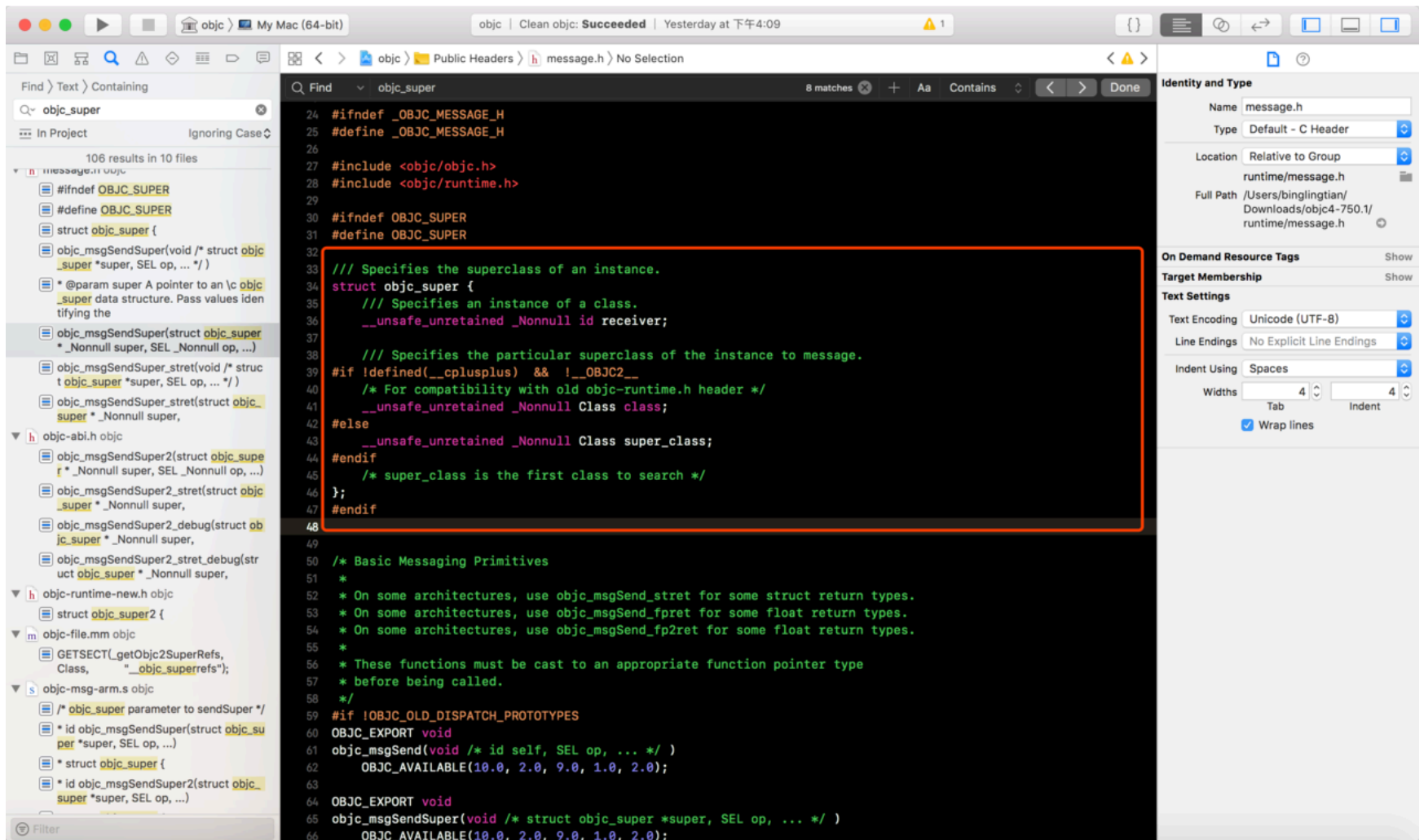
```
objc_msgSendSuper((__rw_objc_super){(id)self, (id)class_getSuperclass(objc_getClass(\"Student\"))}, sel_registerName(\"run\"));
```

- 将第一个参数抽出

```
__rw_objc_super arg = {
    self,
    class_getSuperclass(objc_getClass(\"Student\"))
};
```

```
objc_msgSendSuper(arg, sel_registerName("run"));
```

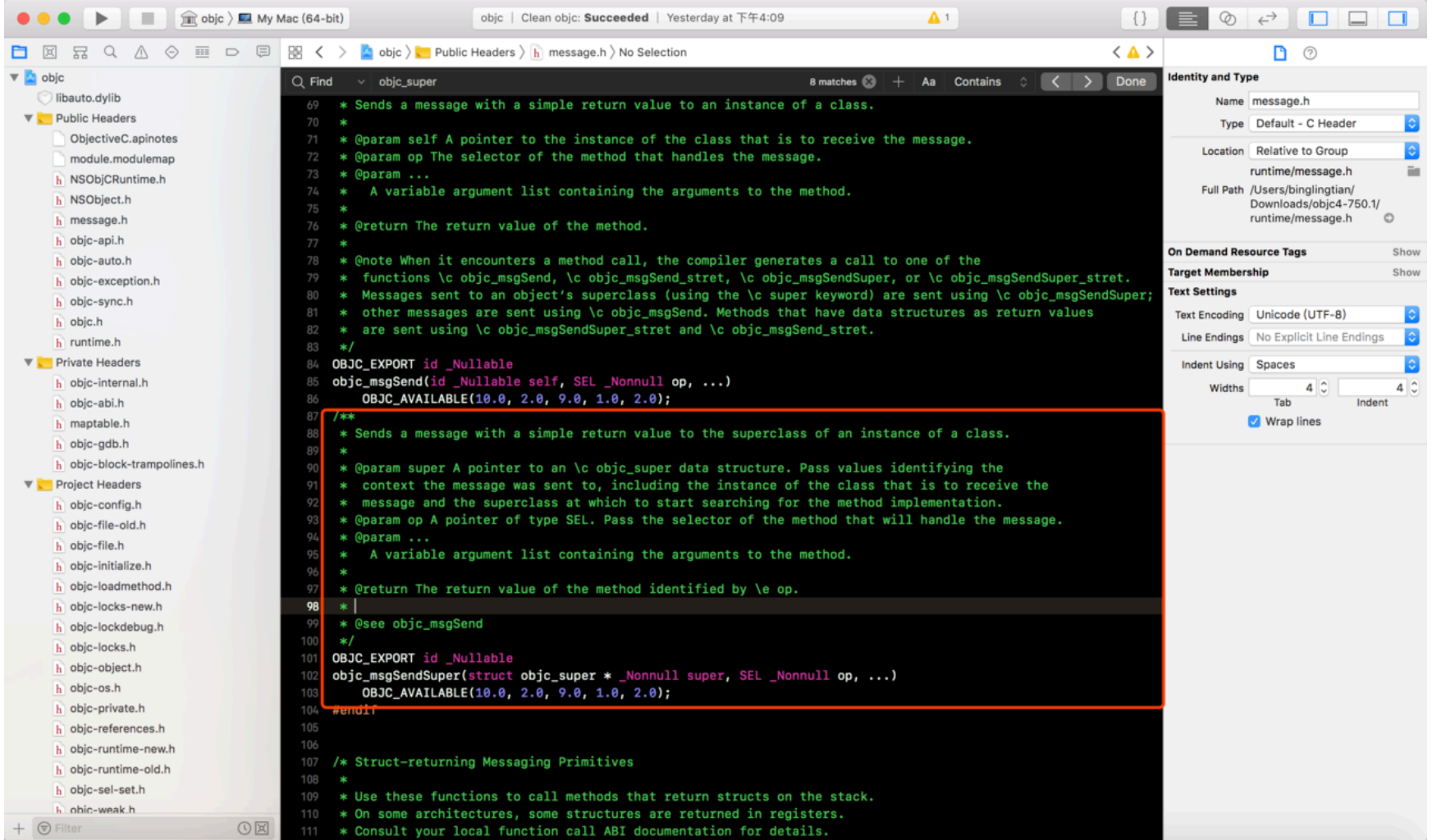
- 可以看到 `super` 的底层调用了 `objc_msgSendSuper` 方法, 并传入两个参数
  - `__rw_objc_super`: 结构体
  - `sel_registerName("run")`: 方法SEL
- 查看源码中 `objc_super` 结构体



- 去掉没有用的代码, 整理如下

```
struct objc_super {
    __unsafe_unretained _Nonnull id receiver;
    __unsafe_unretained _Nonnull Class super_class;
};
```

- 可以看到, 第一个成员变量是 `receiver`, 第二个成员变量是 `super_class`
- 继续查看 `objc_msgSendSuper` 函数的定义



- 在注释中可以看到对 `objc_super` 两个成员变量的解释
  - `receiver` : 消息接收者
  - `super_class` : 从 `super_class` 开始查找调用的方法
- 从实际代码中可以看到, 这两个成员变量分别传入了 `self` 和 `[Person class]`

```
__rw_objc_super arg = {
    self,          // receiver
    class_getSuperclass(objc_getClass("Student"))        // super_class
};
```

- 所以消息接收者是 `self`, 从 `[Person class]` 中开始查找方法
- `objc_msgSendSuper` 的第二个参数是 `sel_registerName("run")`, 即 `@selector(run)`
- 所以 `[super run]`; 在底层代码的原意是: 给 `self` 发送一条消息, 消息名称是 `run`, 这条消息从 `[Person class]` 开始查找

总结:

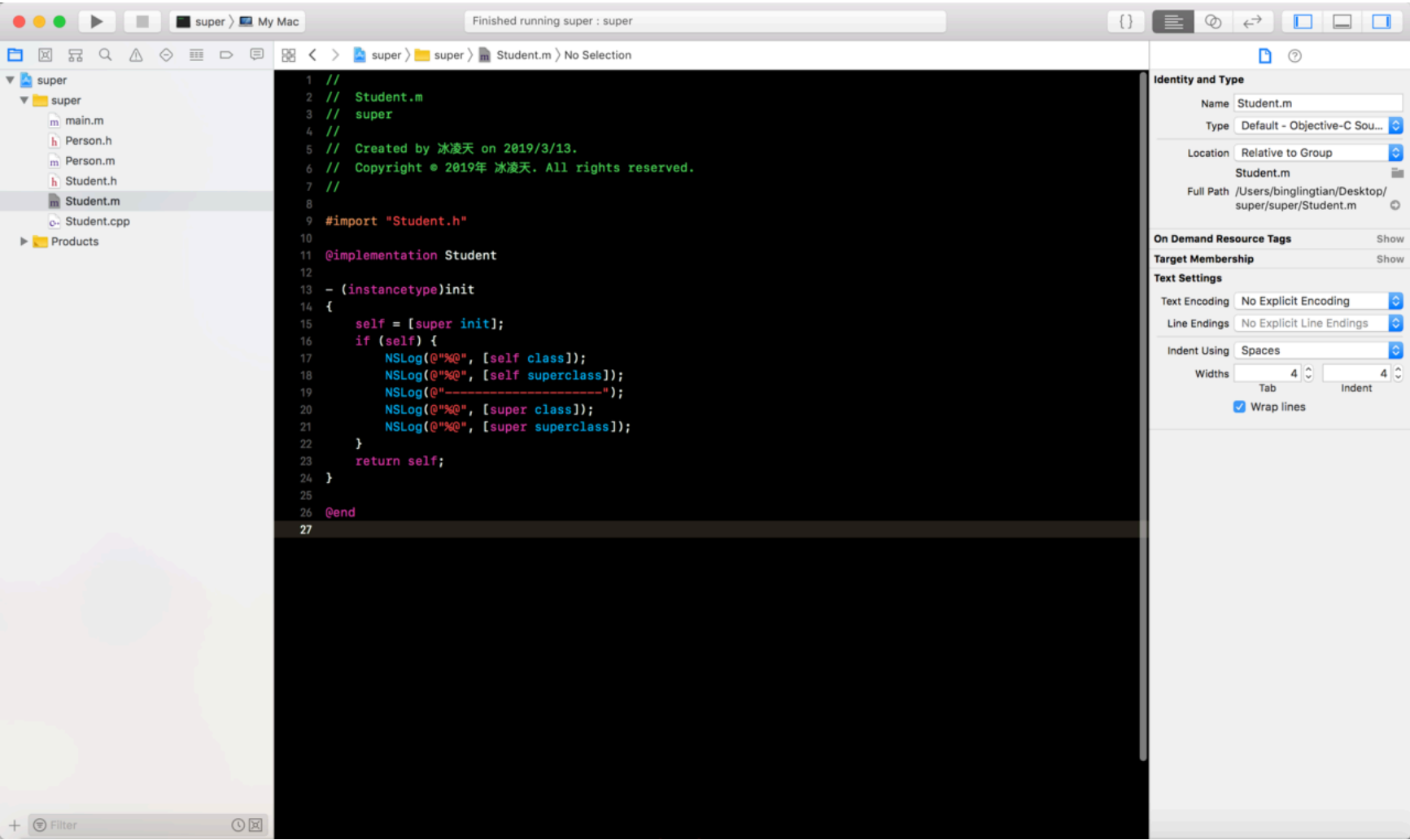
super的含义是, 查询方法的起点是父类, 不是本身的类对象

消息接收者是self, 不是父类对象

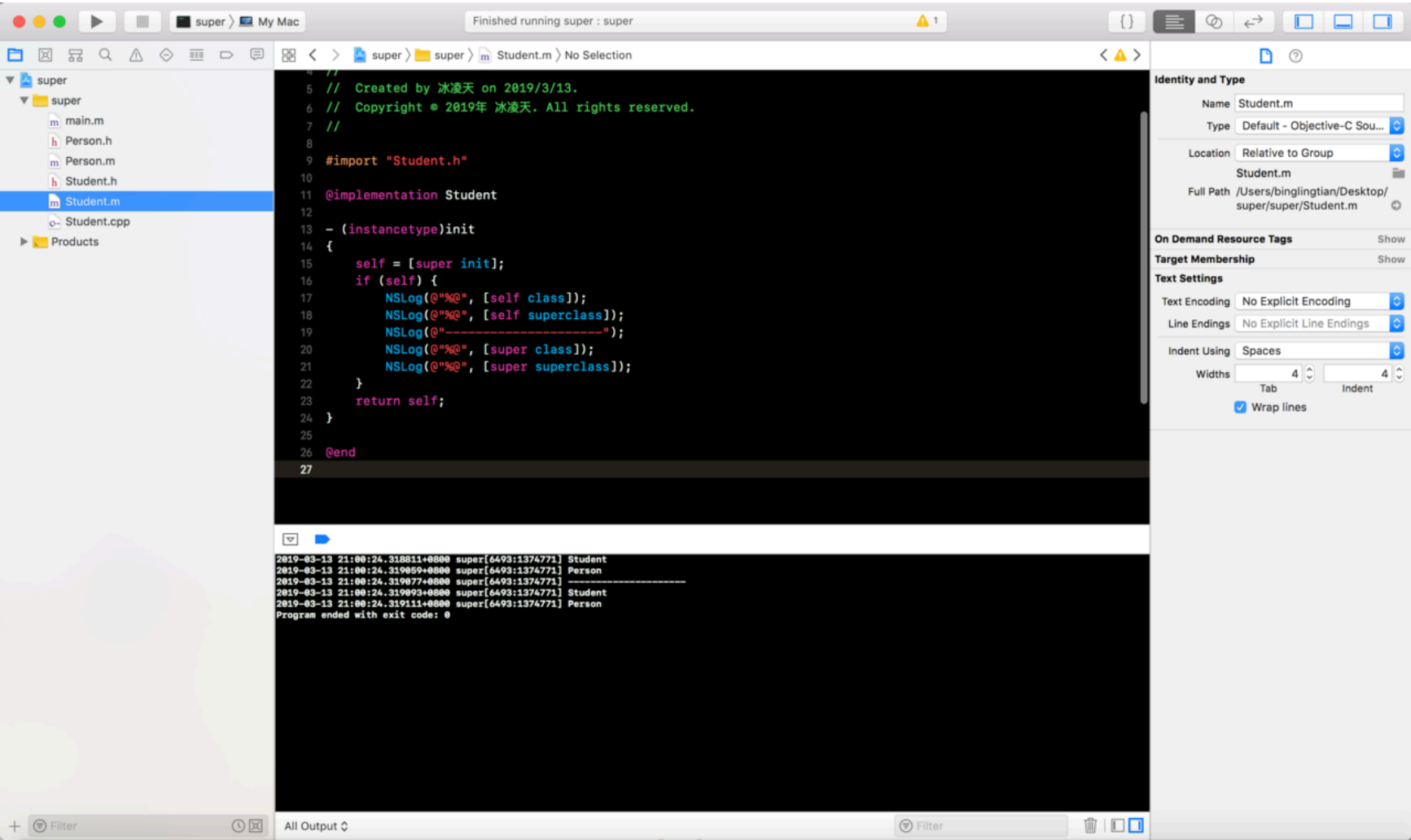
发送的消息是调用的方法

### 三、面试题

- 下面的代码, 运行结果是什么, 为什么



- 运行程序, 打印如下

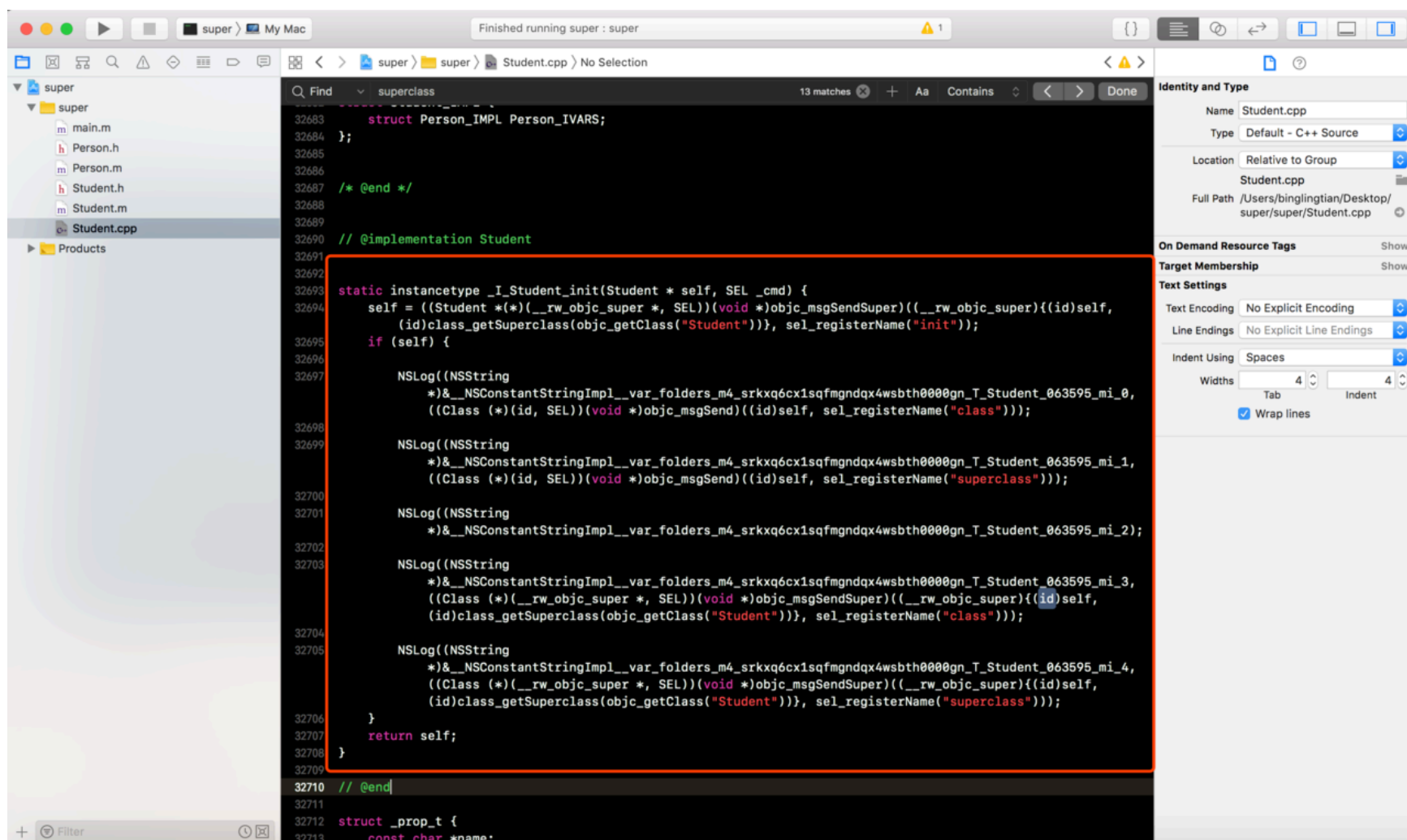




```
2019-03-13 21:00:24.318811+0800 super[6493:1374771] Student
2019-03-13 21:00:24.319059+0800 super[6493:1374771] Person
2019-03-13 21:00:24.319077+0800 super[6493:1374771] -----
2019-03-13 21:00:24.319093+0800 super[6493:1374771] Student
2019-03-13 21:00:24.319111+0800 super[6493:1374771] Person
```

## 打印分析

- 查看底层源码



- 去掉类型转换, 底层代码如下

```
objc_msgSend(self, sel_registerName("class"))
objc_msgSend(self, sel_registerName("superclass"))
objc_msgSendSuper((__rw_objc_super){self, class_getSuperclass(objc_getClass("Student"))})
objc_msgSendSuper((__rw_objc_super){self, class_getSuperclass(objc_getClass("Student"))})
```

- 前两个方法是给 `self` 发送消息, 消息名称是 `class` 和 `superclass`, 结果很明显就是查看 自己的类型 和 父类的类型
- 后两个方法也是给 `self` 发送消息, 只不过是从 父类 开始查询 `class` 和 `superclass` 方法, 而这两个方法都存在于 `NSObject` 中
- 消息接收者同样是 `self`, 调用的方法也是相同, 所以结果就是 自己的类型 和 父类的类型

关注下面的标签，发现更多相似文章

iOS



冰凌天

Lv2

iOS

获得点赞 189 · 获得阅读 9,936

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。快来安装掘金浏览器插件获取高质量内容吧！

## 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



71



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

 15



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

