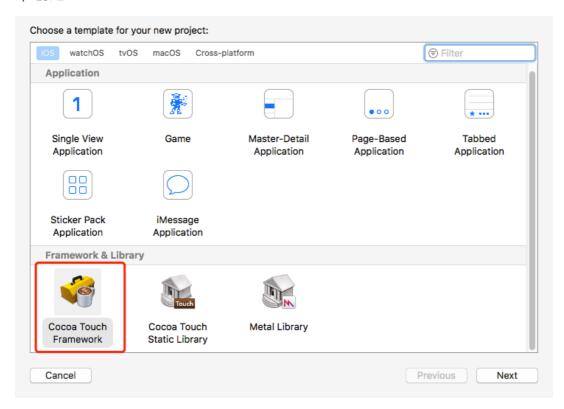
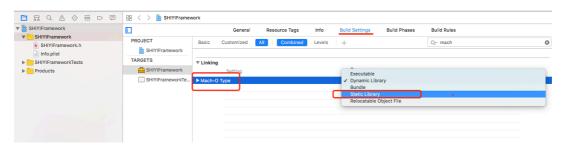
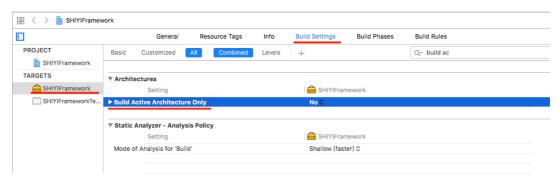
- 一: 制作静态库framework
- 1, 创建framework



2, 设置mach-O Type为 静态库 Static Library



3, 设置编译的时候, 不能只编译活跃的架构:

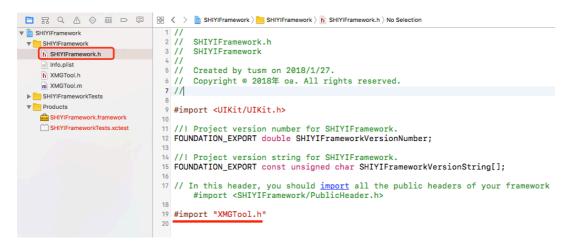


4, 创建XMGTool工具类,并编写类方法log

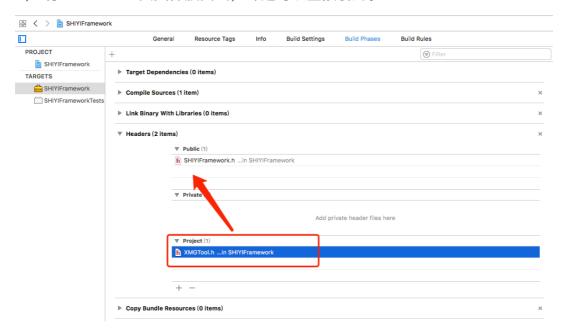
```
    □
    □
    □
    □
    □
    □
    SHIYIFramework )
    □
    SHIYIFramework )
    □
    XMGTool.m ) No Selection

                                  1 //
                                   2 //
                                         XMGTool.m
▼ NHIYIFramework
                                  3 //
                                         SHIYIFramework
    h SHIYIFramework.h
                                  4 //
     Info.plist
                                  5 // Created by tusm on 2018/1/27.
    h XMGTool.h
                                  6 // Copyright © 2018年 oa. All rights reserved.
   m XMGTool.m
                                  7 //
▶ SHIYIFrameworkTests
▼ Products
                                  9 #import "XMGTool.h"
   A SHIYIFramework.framework
                                 10
   SHIYIFrameworkTests.xctest
                                 11 @implementation XMGTool
                                 13 +(void)log{
                                 14
                                         NSLog(@"静态库打印");
                                 15 }
                                  17 @end
```

5, 在公共头文件中引入:



6, 将XMGTool.h 头文件放出去,外边可以直接引用。

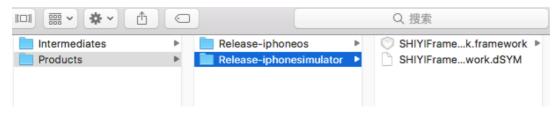


直接拖拽到Public目录中:

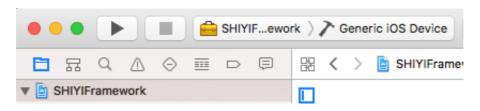


这样打成framework以后,外边就可以直接看到.h文件了。

7,选择真机和模拟器编译。(设置模式为release)



选择真机编译:



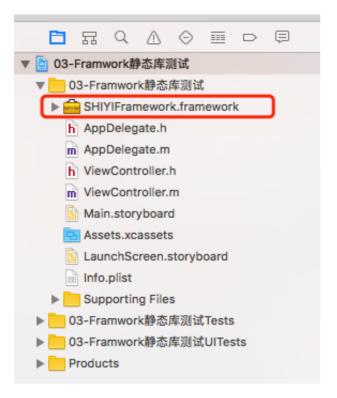
8, 查看.framework支持的架构:



注意: 在查看支持的架构时,直接去找framework里边的可执行文件

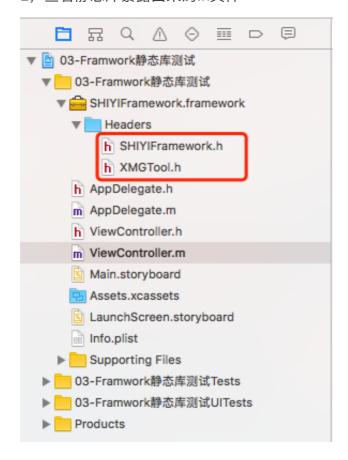
二: 创建项目测试静态库

1, 创建测试项目, 并直接拖进去framework



直接编译,通过

2, 查看静态库暴露出来的.h文件



3, 静态库的使用:

直接引入:

```
9 #import "ViewController.h"
#import <SHIYIFramework/SHIYIFramework.h>
12 @interface ViewController ()
13
14 @end
15
16 @implementation ViewController
17
18 - (void)viewDidLoad {
19
       [super viewDidLoad];
20
       [XMGTool log];
21
22 }
23
24
25 @end
```

直接运行,运行成功:

