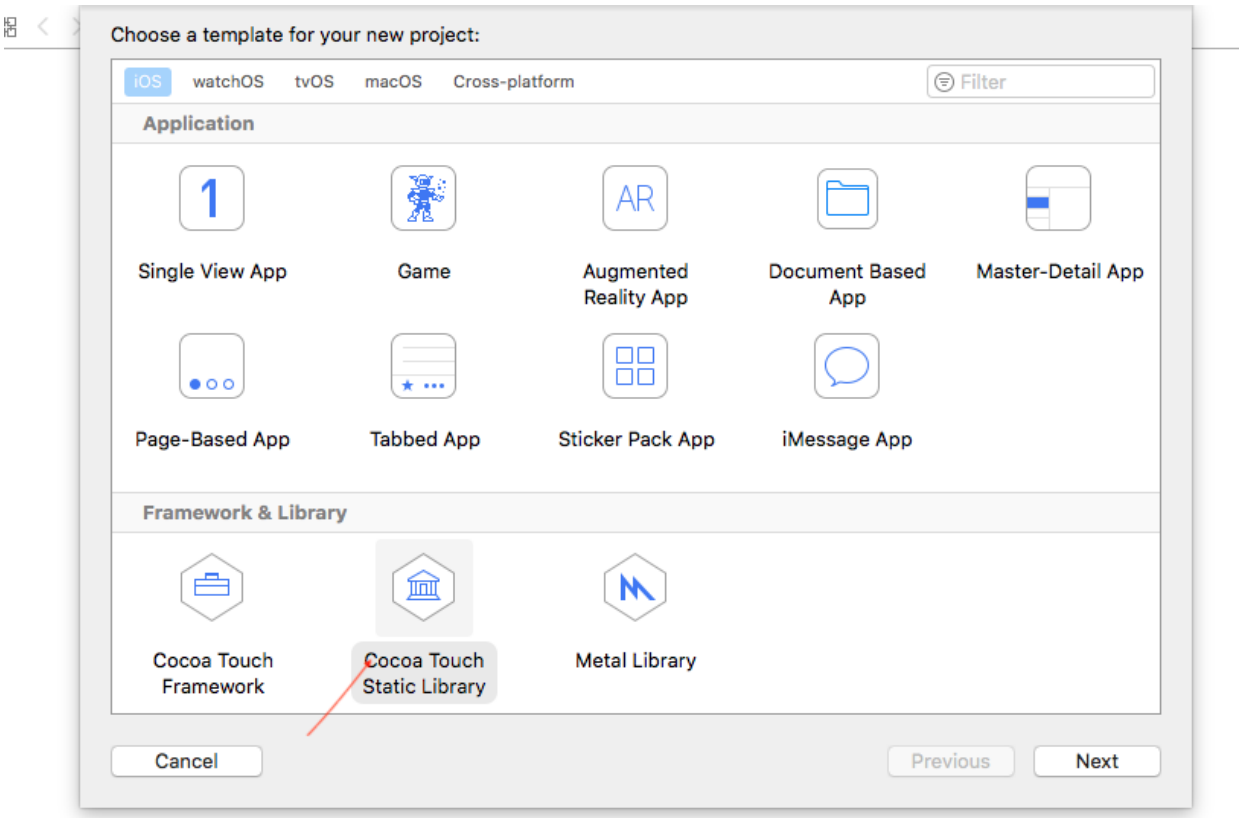
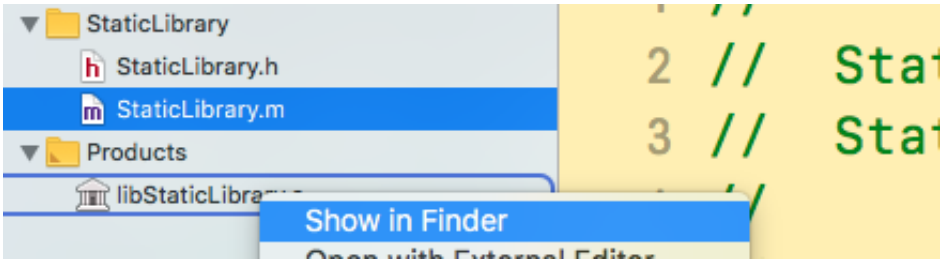
打包动态库和静态库

打包注意，真机下打包的静态库只能在真机下运行，模拟器下打包的静态库只能在模拟器下运行

1. *打包.a静态库*



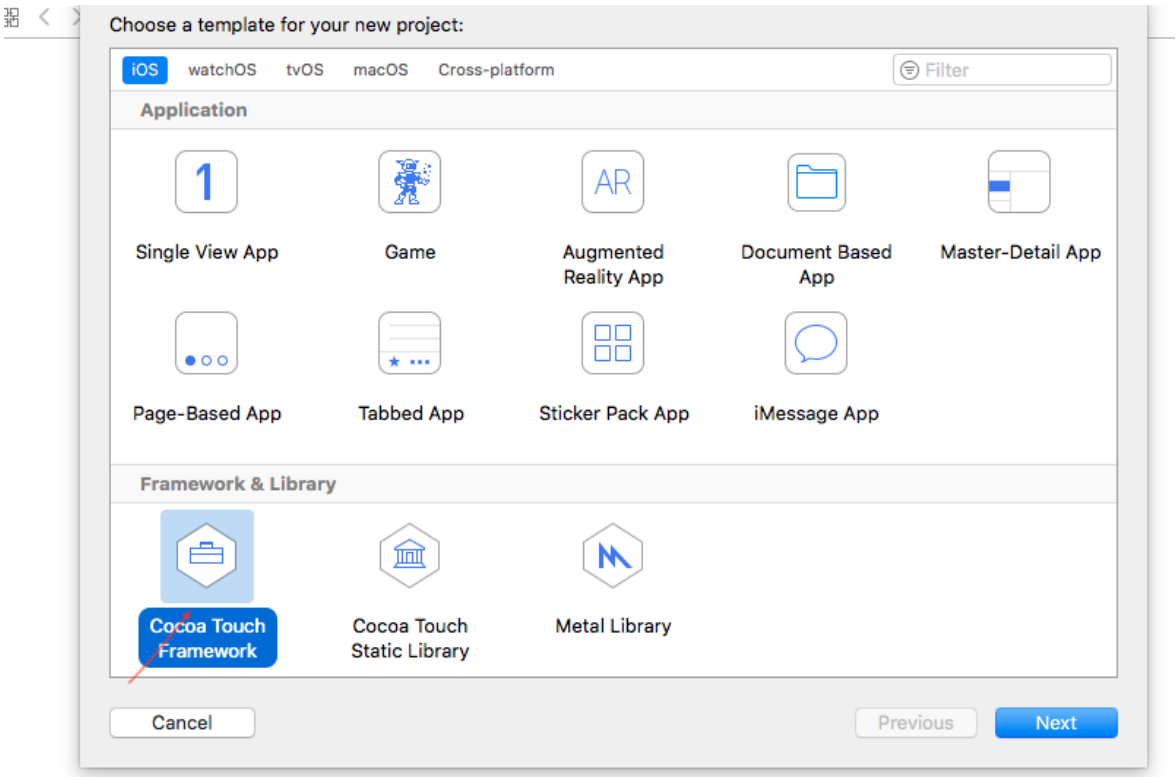
在.h和.m中写入方法，按command+b编译成功



鼠标右键点击，点击Show in Finder 查看静态库

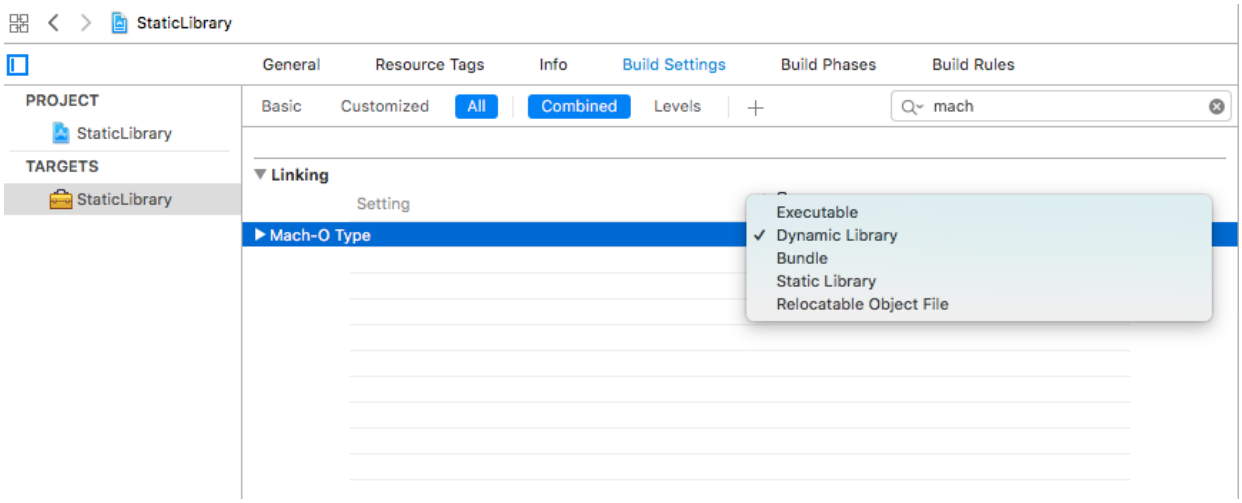
.a文件拖进项目使用

2. *打包.framework静态库*

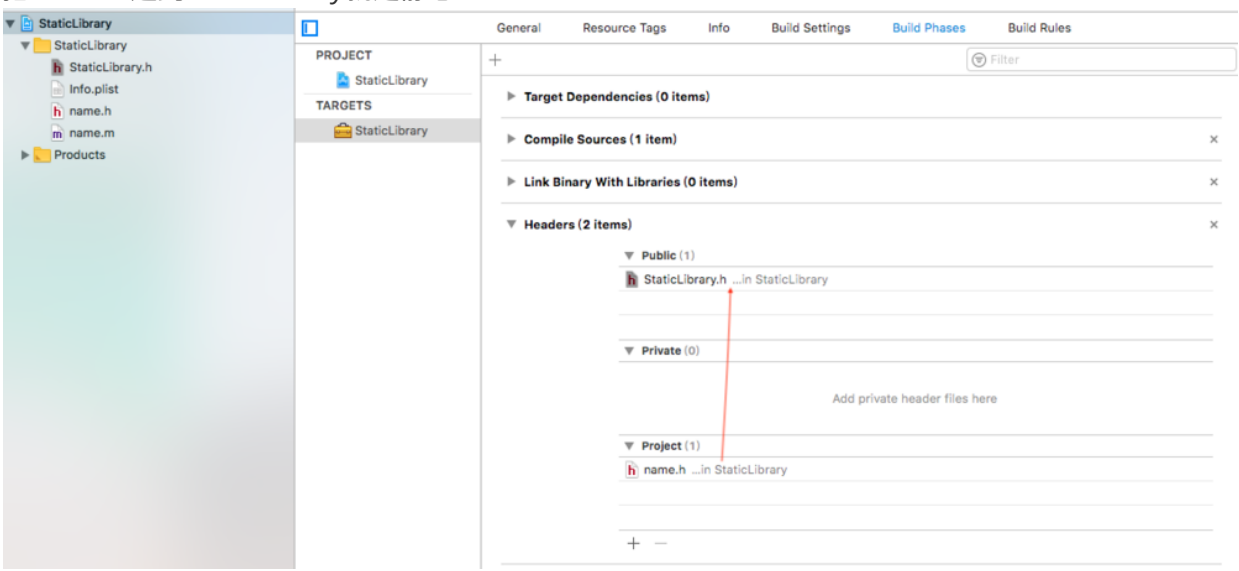


添加类方法，并把类方法的头文件导入到静态库.h里面

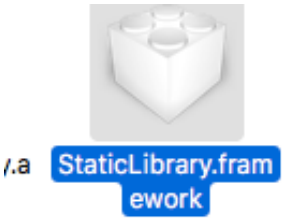
.framework默认是动态库



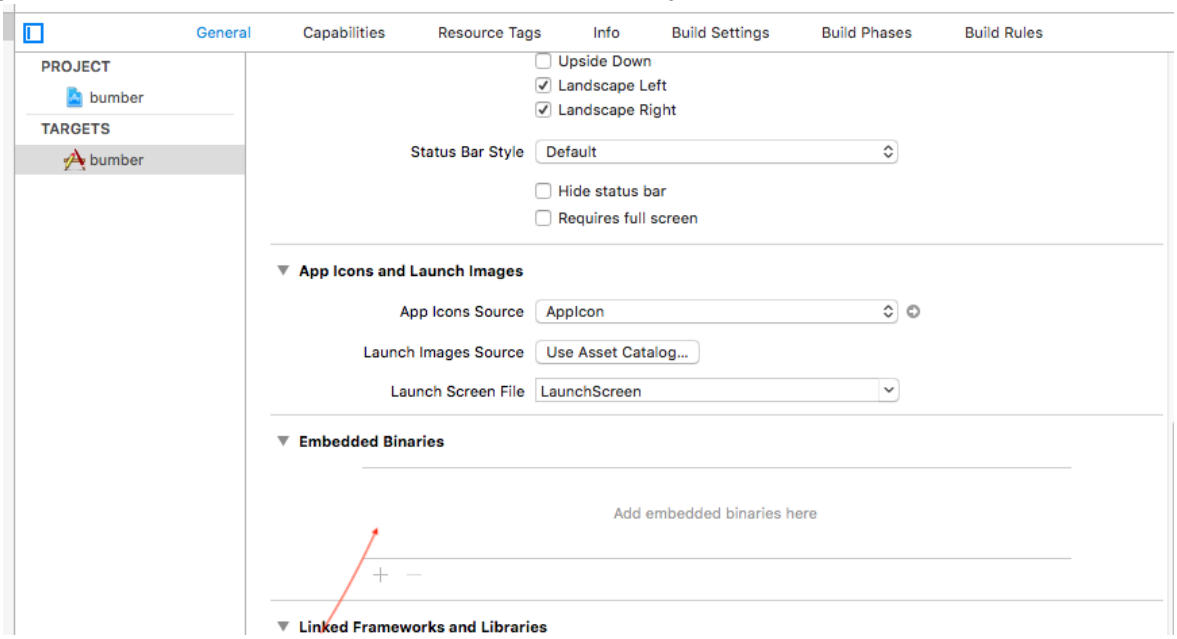
把mach-o 选到 static library就是静态



把类的.h拖到这里公开 然后command+b编译成功



3. *framework动态库 把mach-o 默认是动态库 使用方法跟.framework一样，区别*



使用时需要在这里添加静态库

4. *真机模拟器两用包*

   将真机包和模拟器包使用命令行合到一起，命令格式为lipo -create dic/xxx.framework/xxx dic2/xxx.framework/xxx -output xxx,其中dic和dic2代表生成framework的两个目录，一个是iphones一个是iphonesimulator，而xxx.framework其实就是我们在build过后生成的framework包了，最后output后边的xxx 其实就是最后合成生成的文件，最后将文件覆盖到iphones里边，就会替换原有的xxx文件，具体目录结构如图

上图红色箭头所指部分为生成合成文件将要覆盖的文件，覆盖完成后可以直接将Release-iphones里边将framework文件拿来直接用了，可以用于真机和模拟器的framework动态包就出世了。

用法和其他framework用法完全一致，注意事项就是在引用生成的framework的同时需要在工程中引入生成的framework的相关其他引用即系统以及第三方的framework的引入以及静态库的引入，还有一个设置是在other linker flags下设置-ObjC，整个过程就是这样。