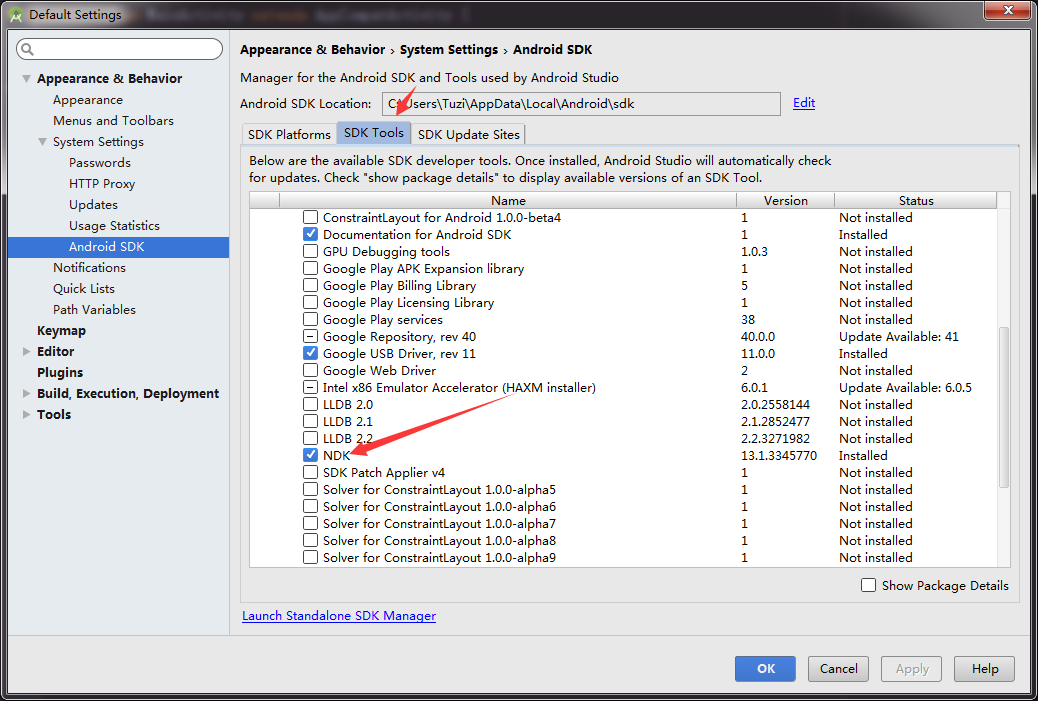
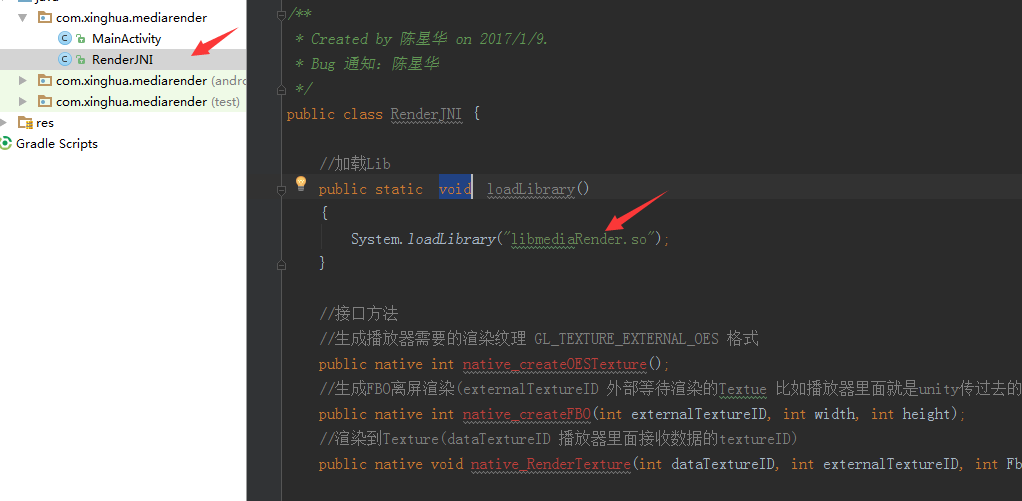
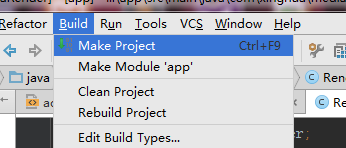
1. 使用android studio生成一个新工程：名称 mediaRender（后面使用opengl做视频渲染）
2. 配置NDK环境



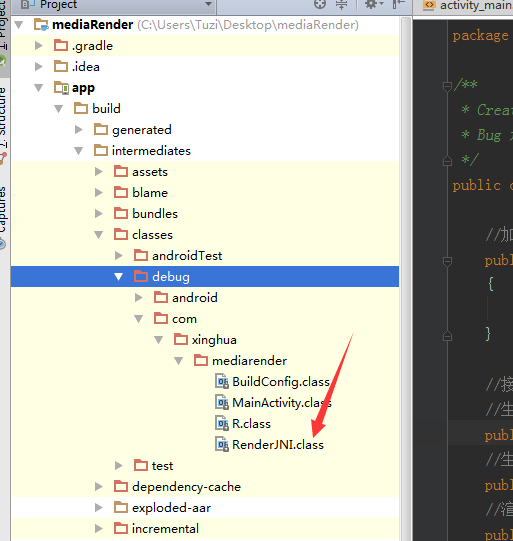
3.编写java就是和c++通信接口



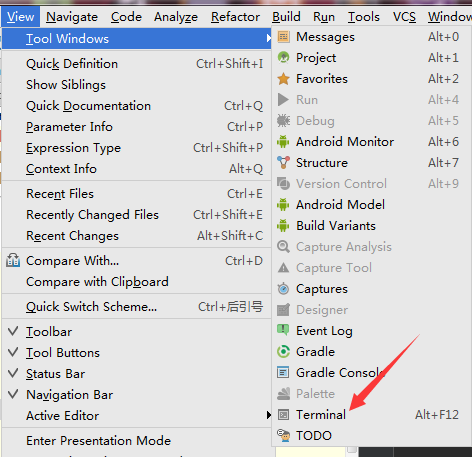
4.make project

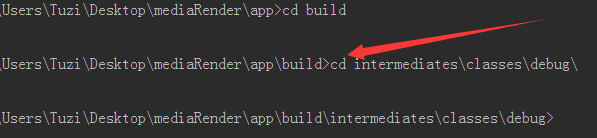


在debug目录下可以看到

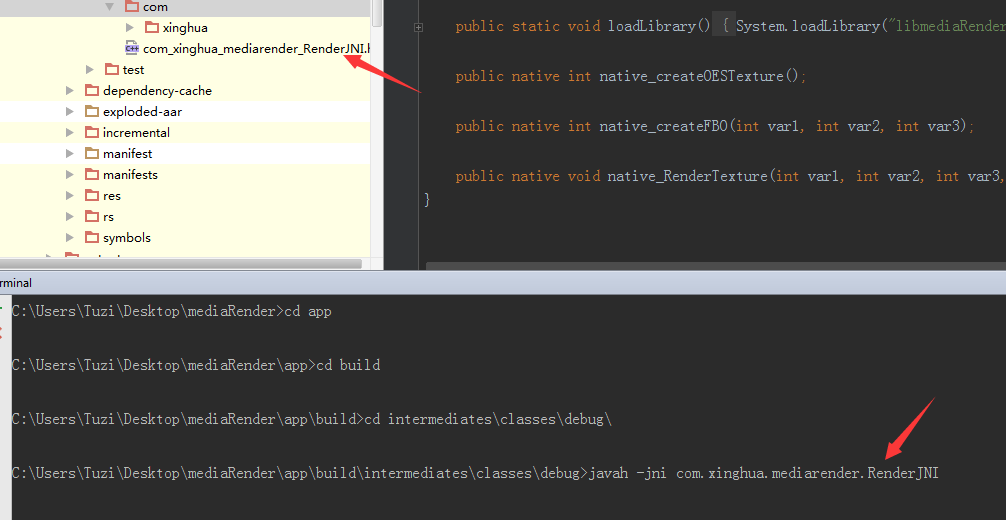


5.打开控制台





输入javah ---------生成接口.h



## Javah 使用时报错：找不到类

使用javah的时候，直接cd到class所在目录包名的上一级，比如class文件在D:/Practice/hhhh/com/test/hello.class

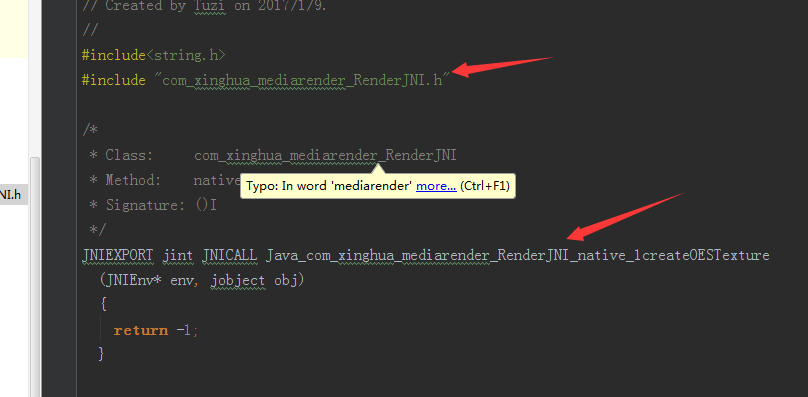
需要cd   
D:/Practice/hhhh

然后javah   
com.test.hello

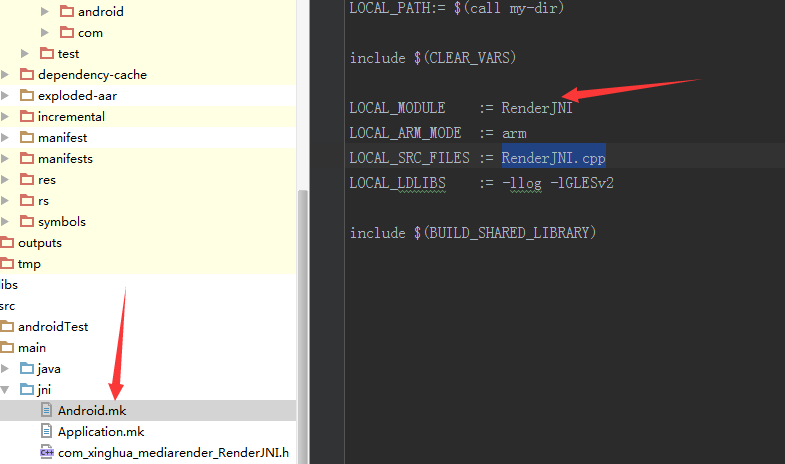
就在D:/Practice/hhhh下生成了需要的.h文件

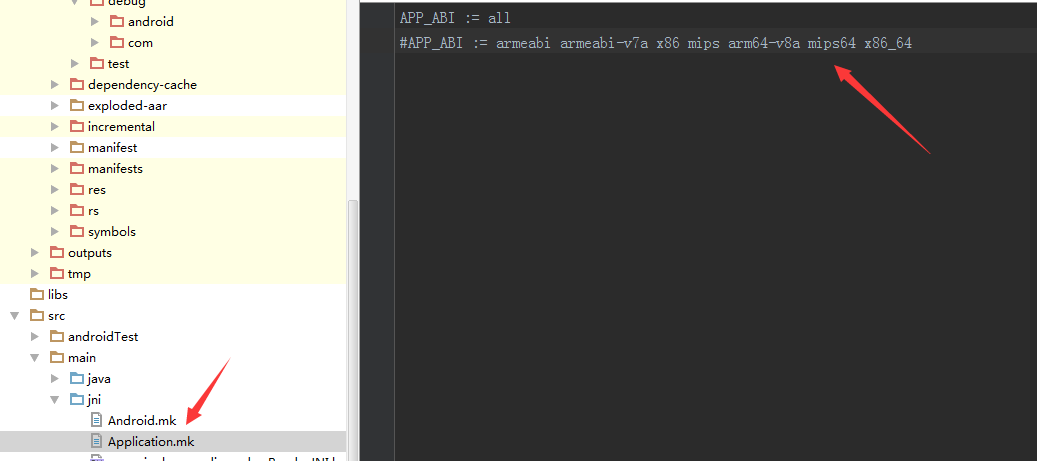
classpath什么的就不需要了。以上。

6.生成jni文件夹和生成 RenderJNI.cpp并且实现里面的方法（刚刚生成的renderJNI.h里面的方法）

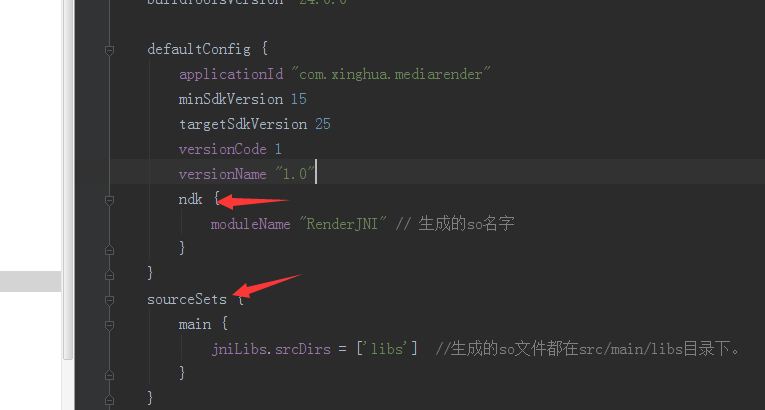


# 7. 创建Android.mk和Application.mk文件

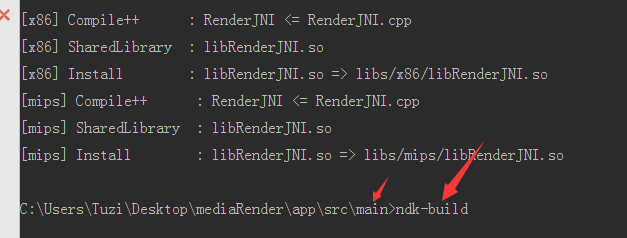


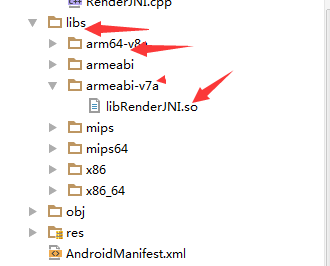


# 8. 配置gradle

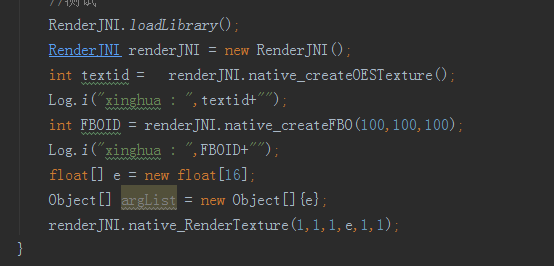
9

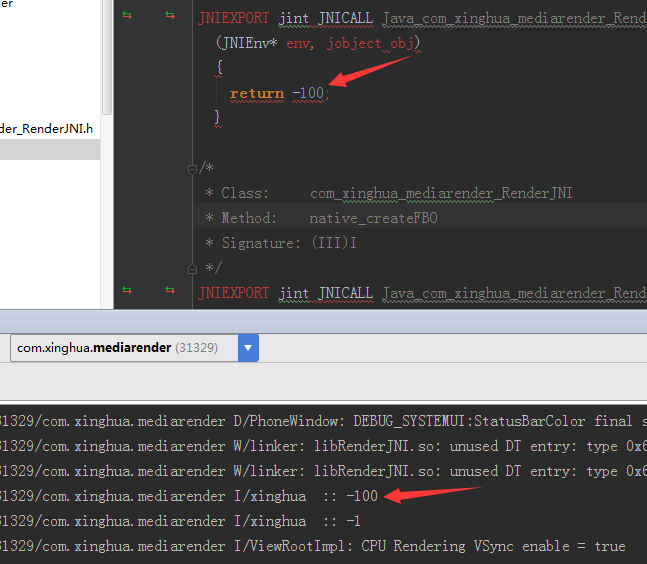
9.build





10.测试





注意：

Ndk-build 时候发现#include <GLES3/gl3.h>找不到，是ndk版本不对查看ndk的路径，发现android-18之前都不支持，修改Application.mk指定编译的版本APP\_PLATFORM = android-21

注意c++和java NDK打印:

//注意要使用JAVA的啊打印1.必须加入头文件<android/log.h>  
//2.Android.mk加入LOCAL\_SHARED\_LIBRARIES +=\ libcutils  
#define LOG\_TAG "Xinghua"  
#define LOG(...) \_\_android\_log\_print(ANDROID\_LOG\_INFO, LOG\_TAG, \_\_VA\_ARGS\_\_)

//c++ java ndk 数组转换(最后记得释放)  
textureidArr = env->GetIntArrayElements(Attr, 0);  
ArrLength = env->GetArrayLength(Attr);