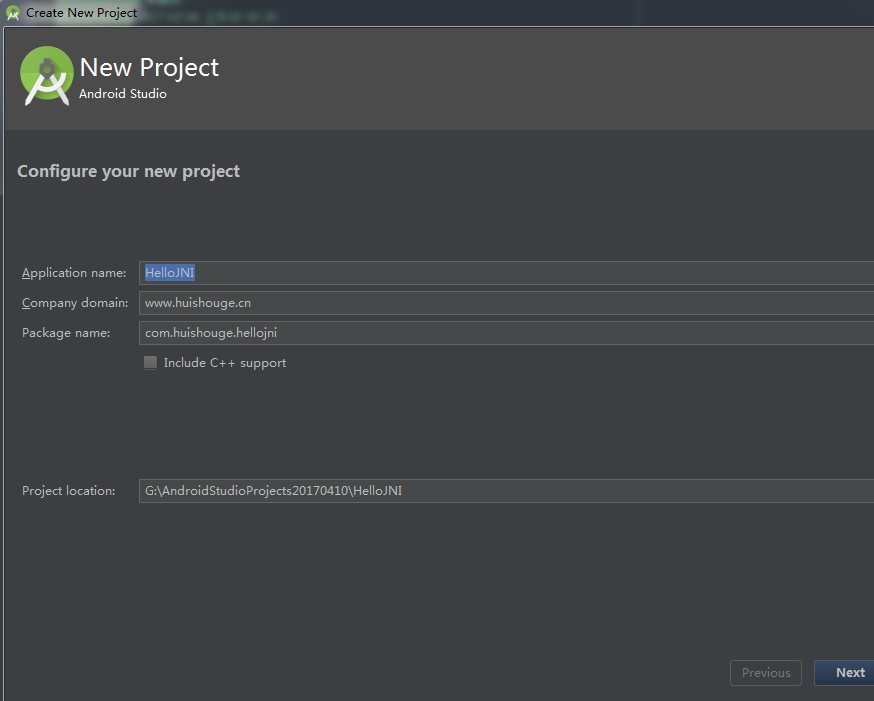
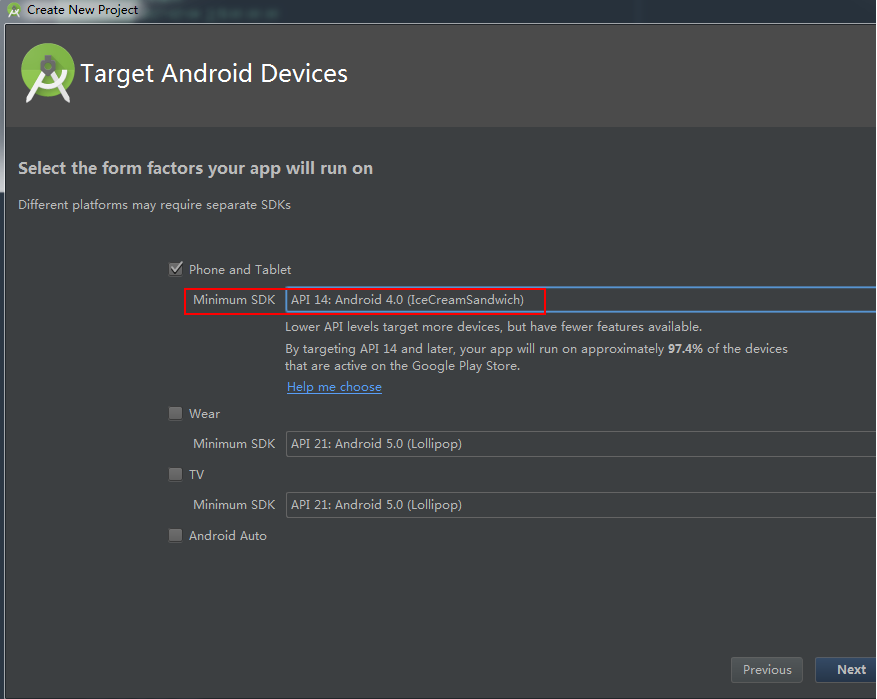
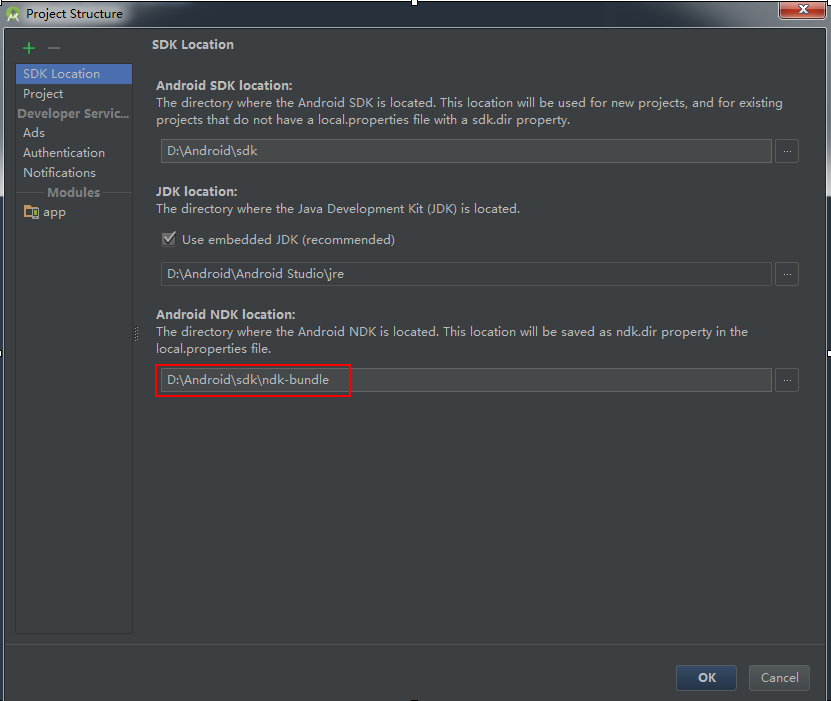
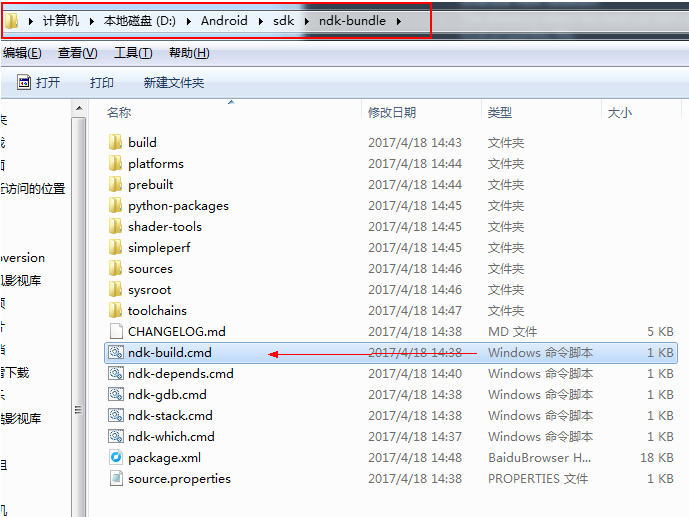
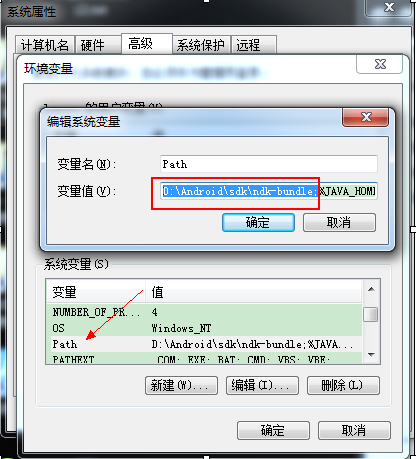
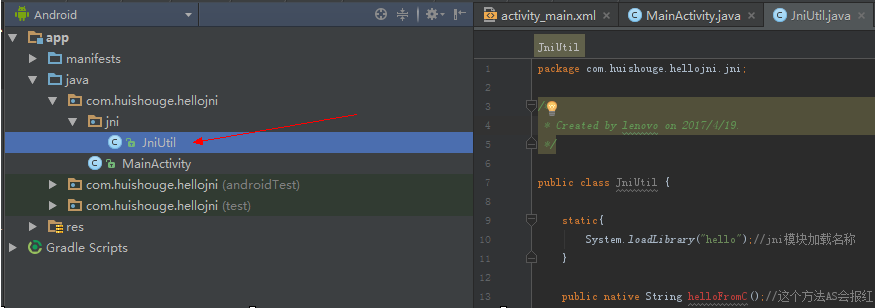
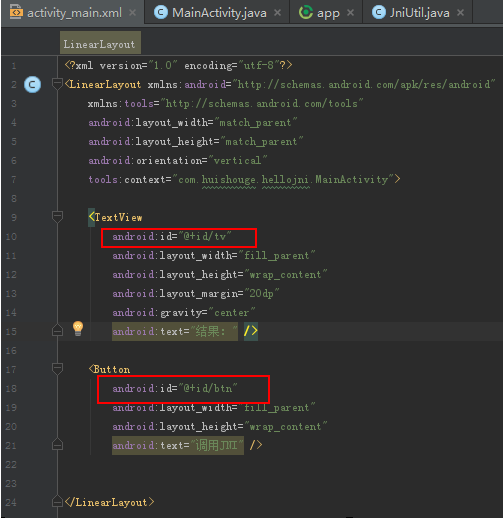
首页 创建项目HelloJNI

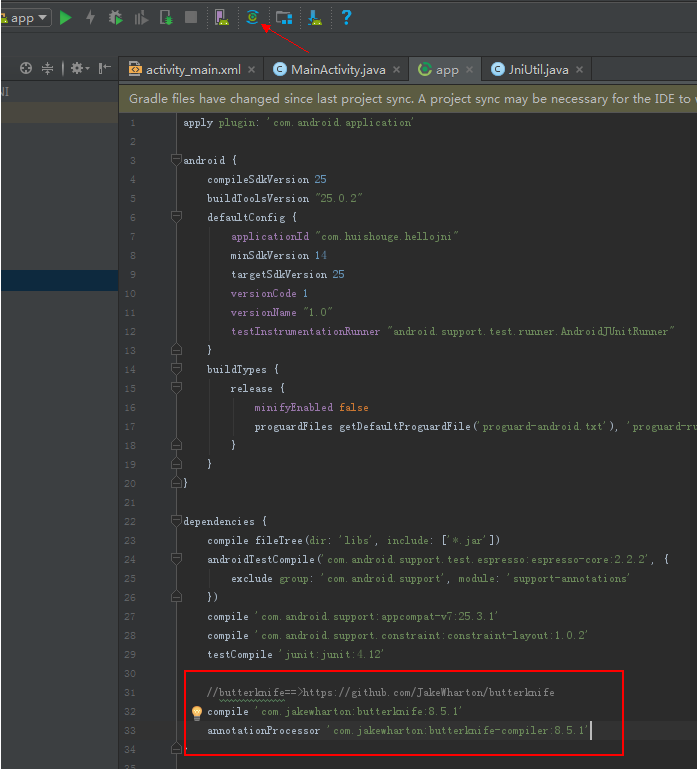


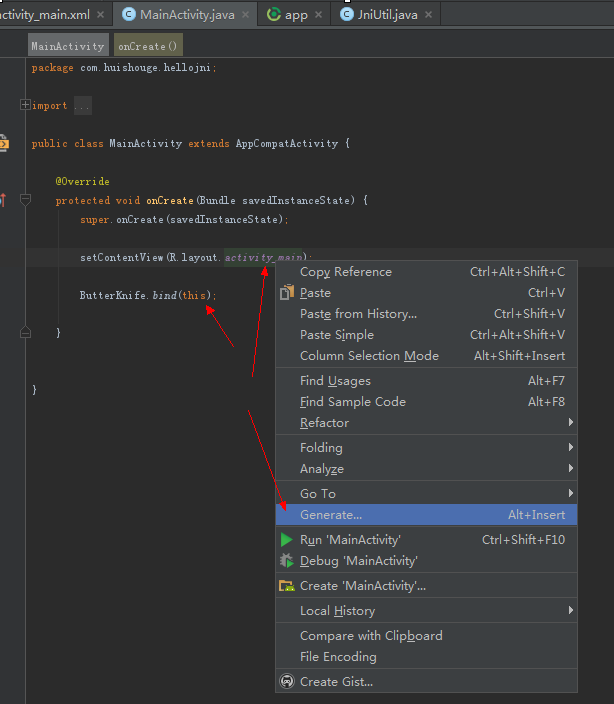
最小SDK根据实际需要，这时我们选择4.0即API14

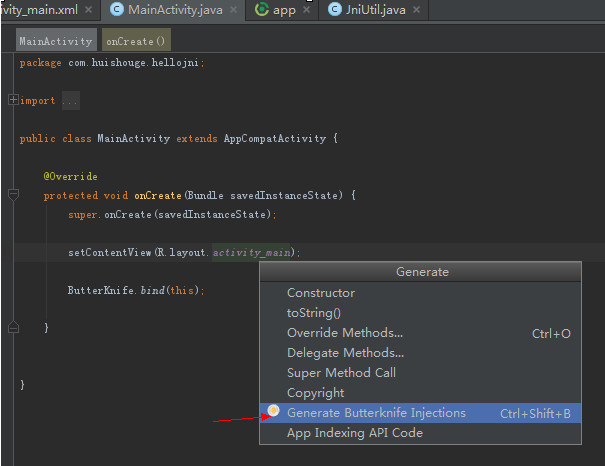


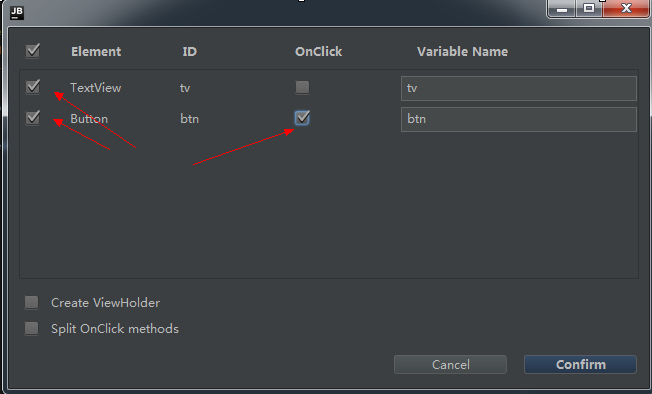
配置NDK环境，如果没有开发配置过NDK，默认NDK目录是空的，路径框下面会有个下载按钮可以在线下载NDK环境，通过这里下载默认是下载在你的NDK目录下的ndk-bundle下

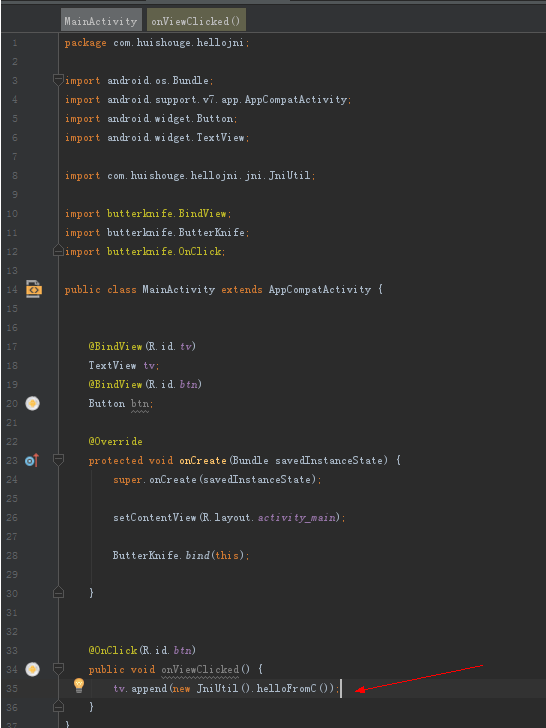
  
ndk-build.cmd是编译NDK要使用到的命令  
  
配置NDK环境变量到path  
  
创建jni包专门用来存在jni相关，创建JniUtil类  
初始jni的so库，hello就是so  
static{  
System.*loadLibrary*("hello");//jni模块加载名称  
}  
  


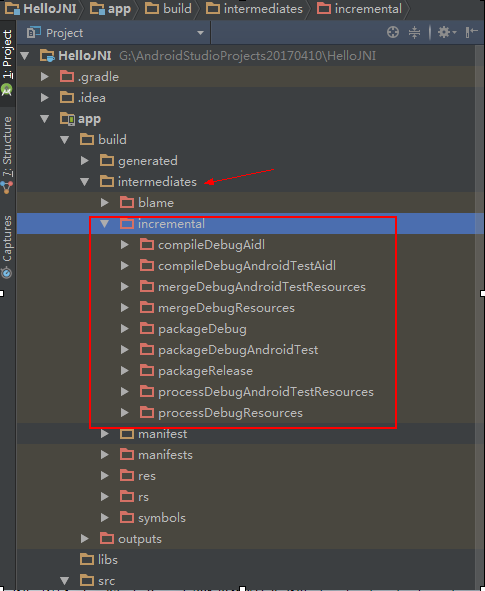


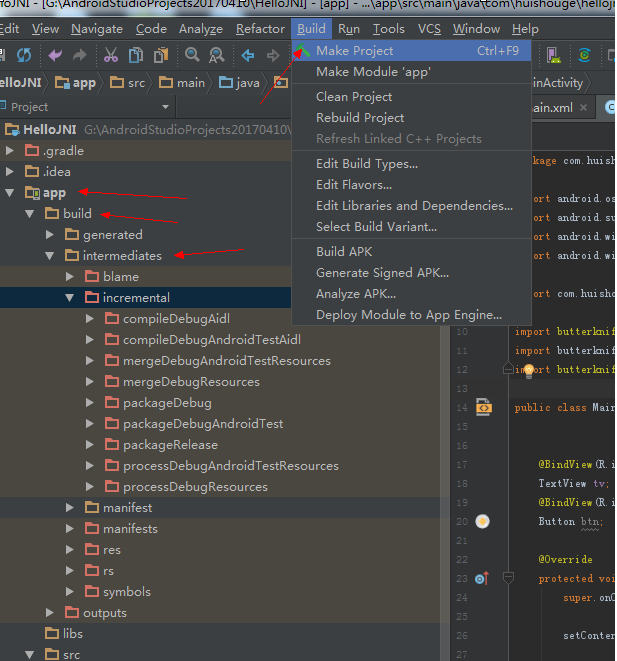


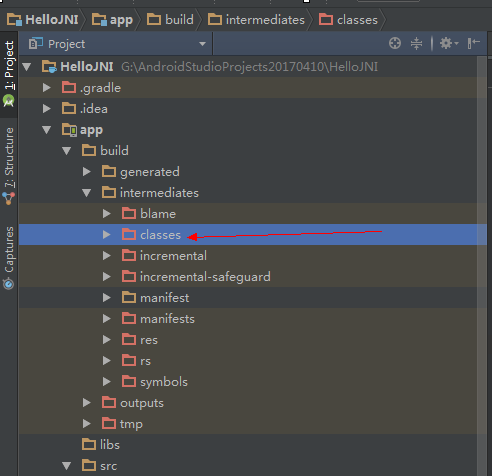


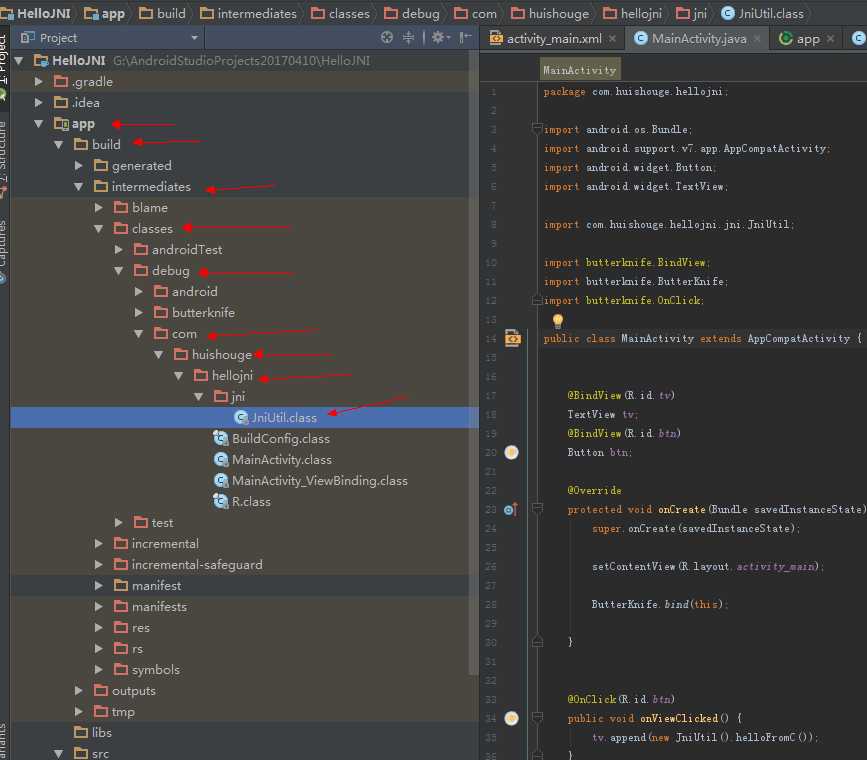


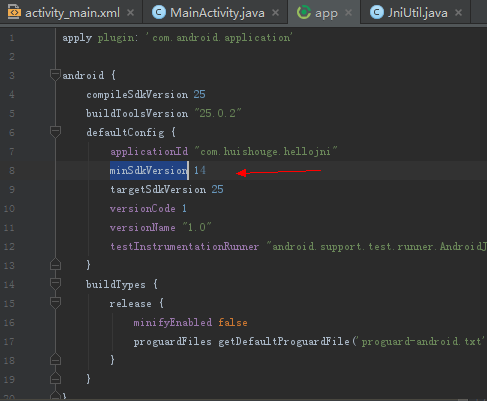






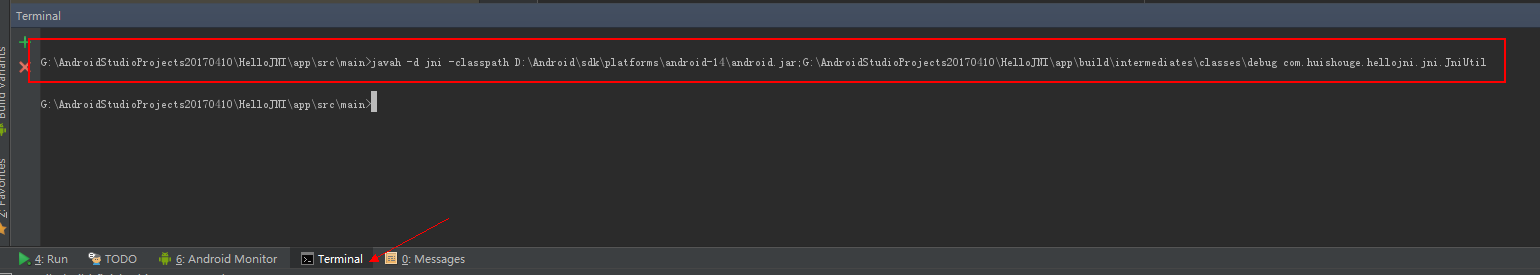




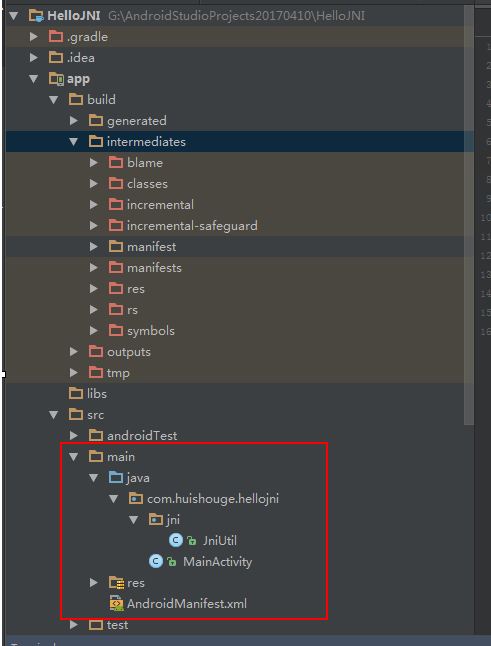


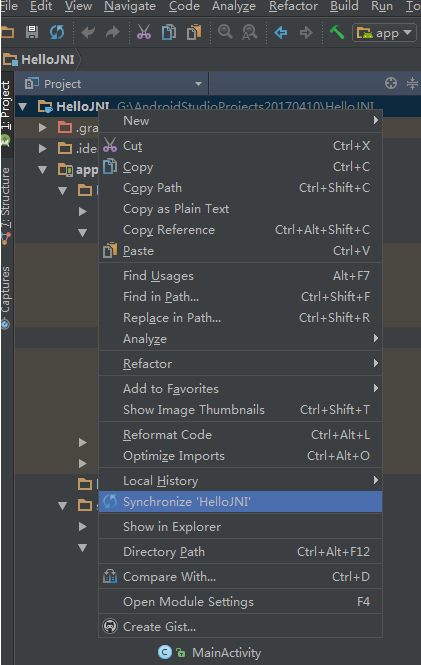
命令如下：(命令中的android.jar使用的是app中的最小sdk路径,后面分号加上生成jni头的绝对路径)

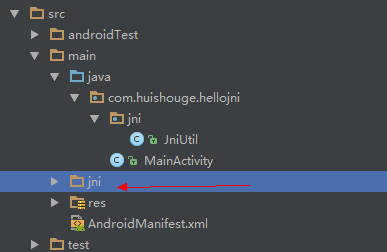
javah -d jni -classpath D:\Android\sdk\platforms\android-14\android.jar;G:\AndroidStudioProjects20170410\HelloJNI\app\build\intermediates\classes\debug com.huishouge.hellojni.jni.JniUtil

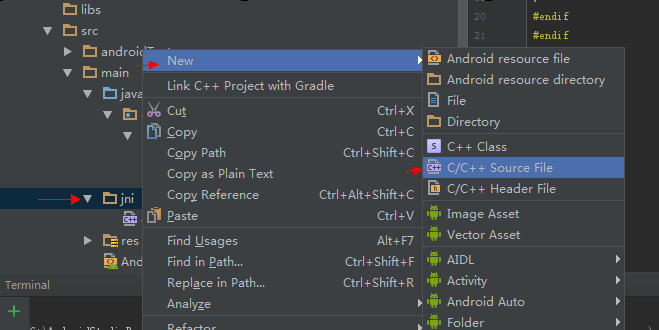


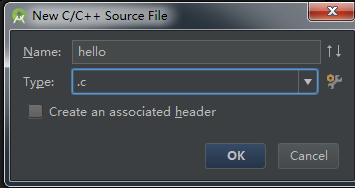
本来main目录是没有jni目录的执行以上命令后会自动生成jni目录，注意要刷新：

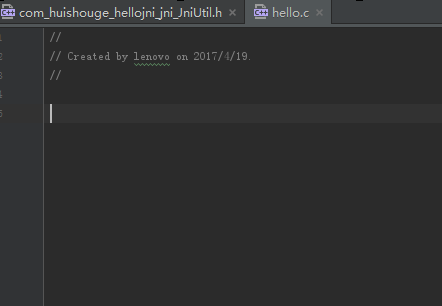






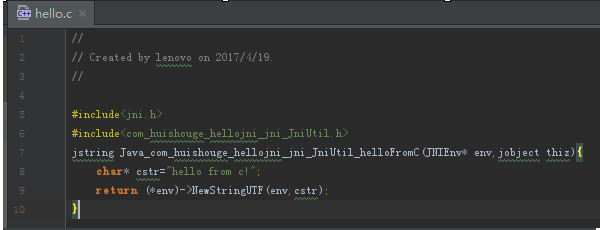


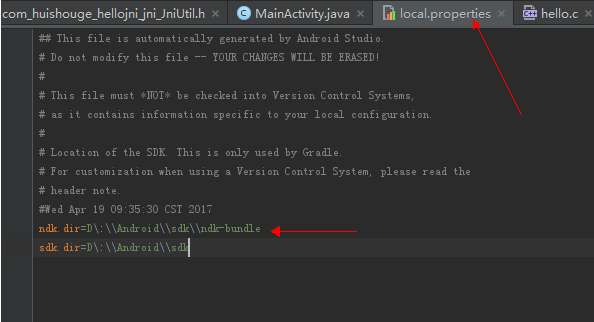




.c文件内容如下：

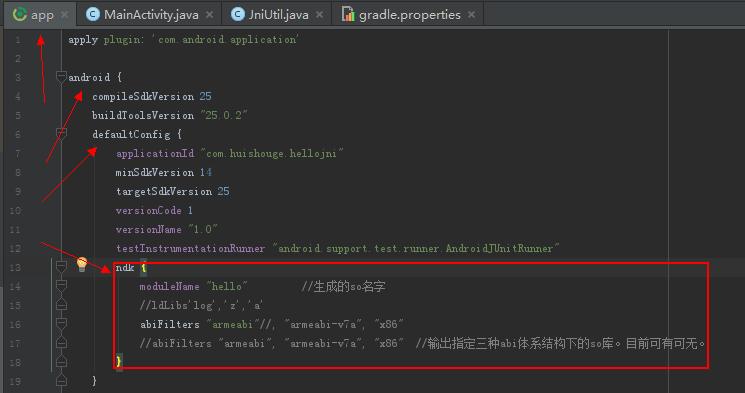
#include<jni.h>  
#include<com\_huishouge\_hellojni\_jni\_JniUtil.h>  
jstring Java\_com\_huishouge\_hellojni\_jni\_JniUtil\_helloFromC(JNIEnv\* env,jobject thiz){  
 char\* cstr="hello from c!";  
 return (\*env)->NewStringUTF(env,cstr);  
}





App中的build.gradle中配置如下：

ndk {  
 moduleName "hello" //生成的so名字，注意与System.loadLibrary("hello");中的名称要一致  
 //ldLibs'log','z','a'  
 abiFilters "armeabi"//, "armeabi-v7a", "x86"  
 //abiFilters "armeabi", "armeabi-v7a", "x86" //输出指定三种abi体系结构下的so库。目前可有可无。  
}



Gradle.properties中配置如下：

android.useDeprecatedNdk=true

