# 1.androi ndk 开发

第一步：

创建对应的native方法；

第二步：

创建jni文件夹目录

第三部：

创建jni头文件，使用dos命令：javah -classpath bin/classes -d jni com.example.ndk.NdkNativeJava

生成com\_example\_ndk\_NdkNativeJava.h头文件

第四部：

创建 c文件,导入类库

#include<stdio.h>

#include<stdlib.h>

#include"com\_example\_ndk\_NdkNativeJava.h"

将新创建的jni头文件中的方法在c文件中实现

第五步：

创建Android.mk文件

LOCAL\_PATH := $(call my-dir)

include $(CLEAR\_VARS)

LOCAL\_MODULE := hello

LOCAL\_SRC\_FILES := hello.c

include $(BUILD\_SHARED\_LIBRARY)

第六步：

编译生成.so文件(libhello.so)

dos命令：ndk-build

第七部：

加载libhello.so文件

static{

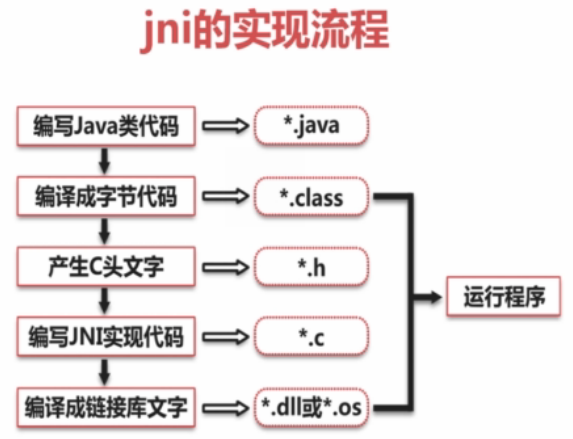
System.loadLibrary("hello");

}

扩展：Android Studio ndk开发命令

javah -classpath build\intermediates\classes\debug -d jni com.example.zhaofeng.myapplication.ndk.NdkNativeJava

# 2.ndk的概念



1.什么是ndk

官网：https://developer.android.com/tools/sdk/ndk/index.html

帮助android开发c/c++的工具

2.什么时候用到ndk

1.代码的保护，由于apk的java层代码很容易被反编译，而c/c++好一点

2.在ndk中调用第三方c/c++库，因为大部分的开源库都是用c/c++代码编写的

3.便于移植，用c/c++写的库可以方便在其他的嵌入式平台上再次使用

3.什么是交叉编译

简单的说，就是在一个平台上生成另一个平台上可执行的代码

在pc(电脑)上开发编译成可在arm(嵌入式处理器)执行的过程，即为交叉编译

4.jni是什么

Java Native Interface(JNI)标准成为java平台的一部分，它允许java代码和其他语言写的代码进行交互

5.什么是链接库

1.静态链接库

std

2.动态链接库

6.make文件介绍

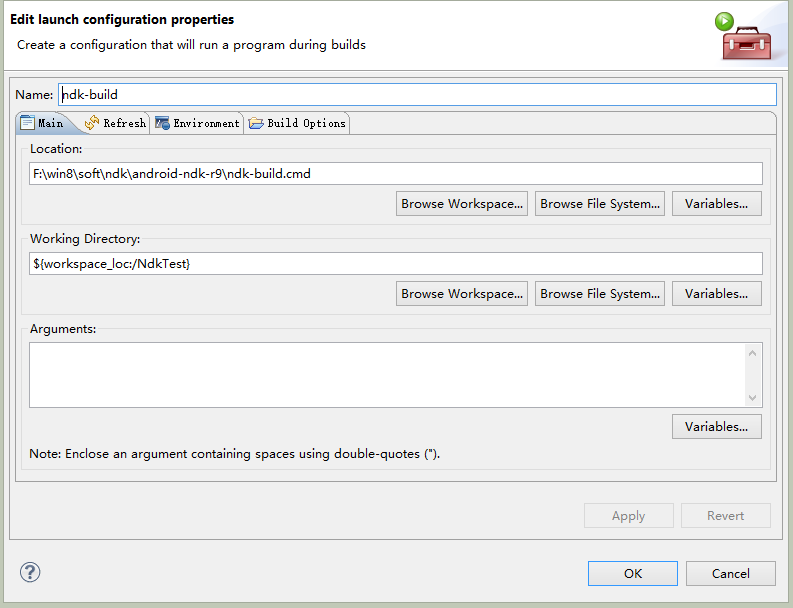
Android.mk文件，进行一些配置

7.ndk开发包目录介绍

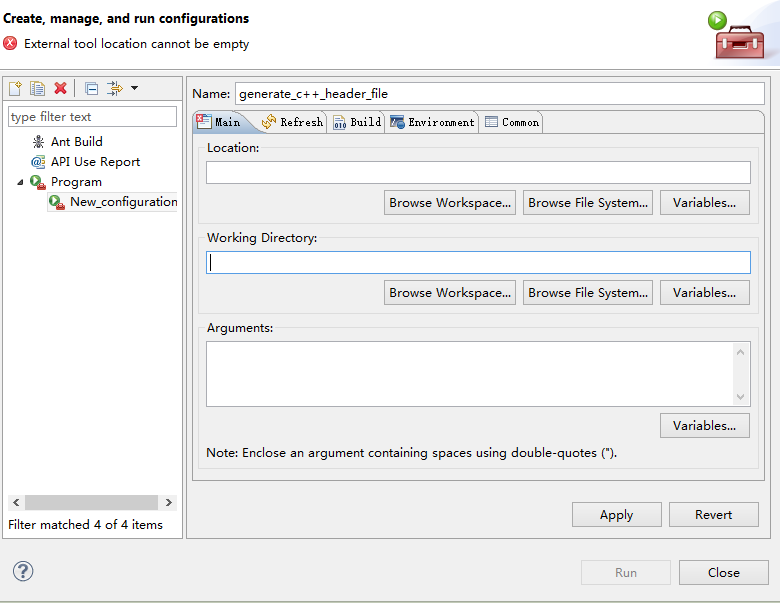
# 3.ndk eclipse 环境搭建

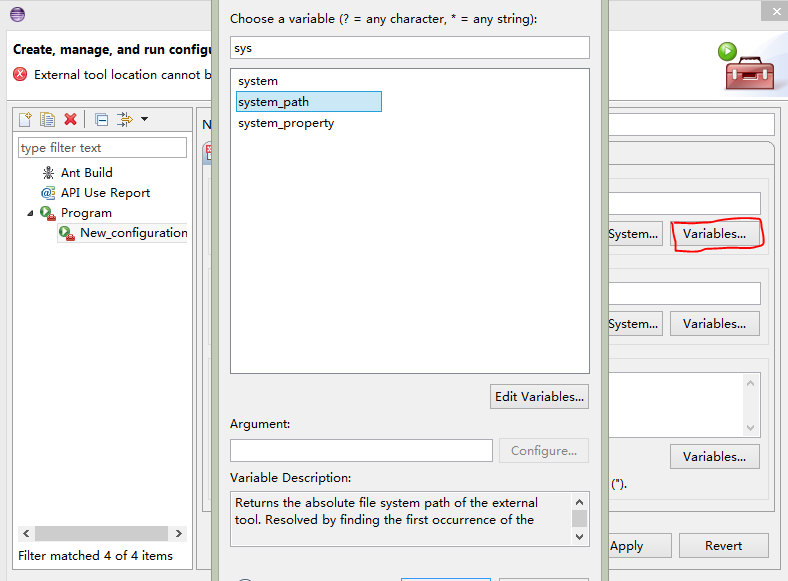
1.window——>Preferences——>Android——>NDK选择对应的目录

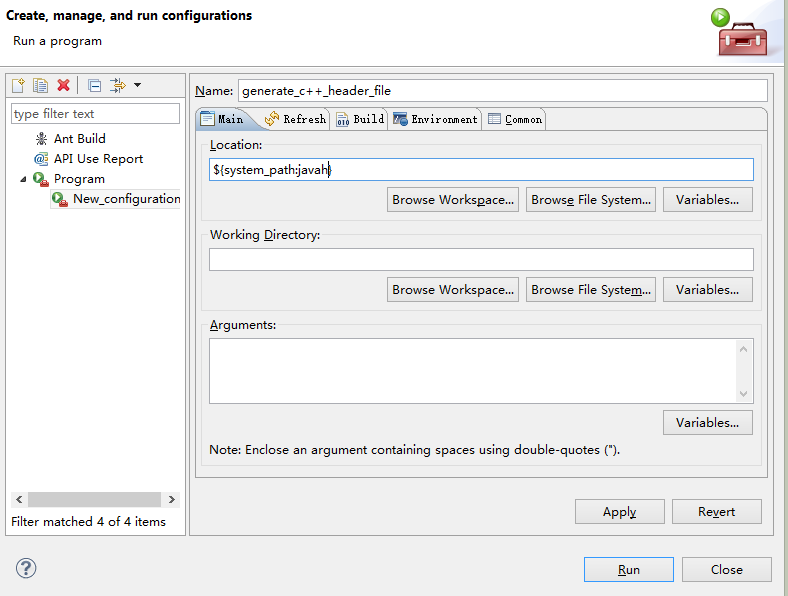
2.选择ndk项目，Properties——>Builder下新建 ndk-build(名称随意)，具体配置看图:

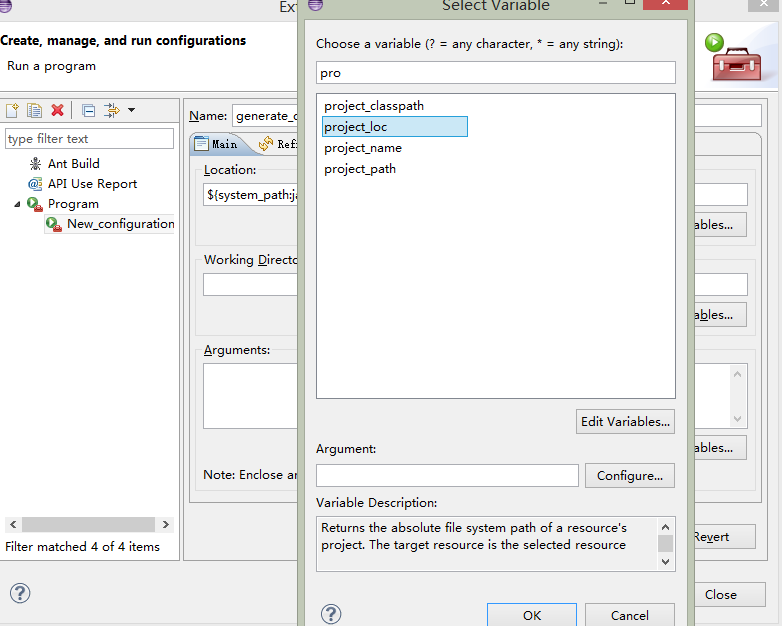


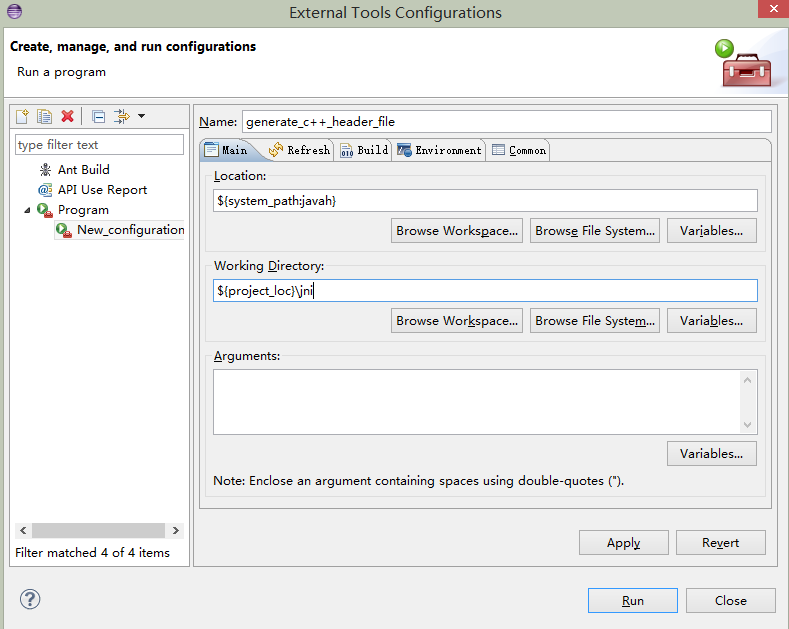
3.配置生成javah选项

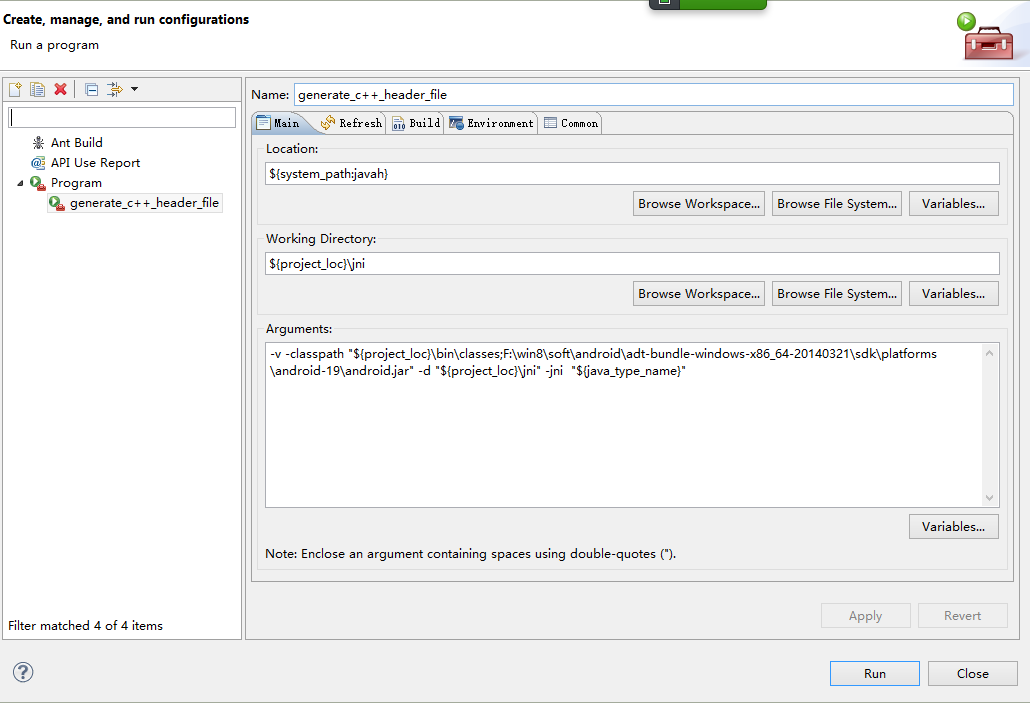
第一步：







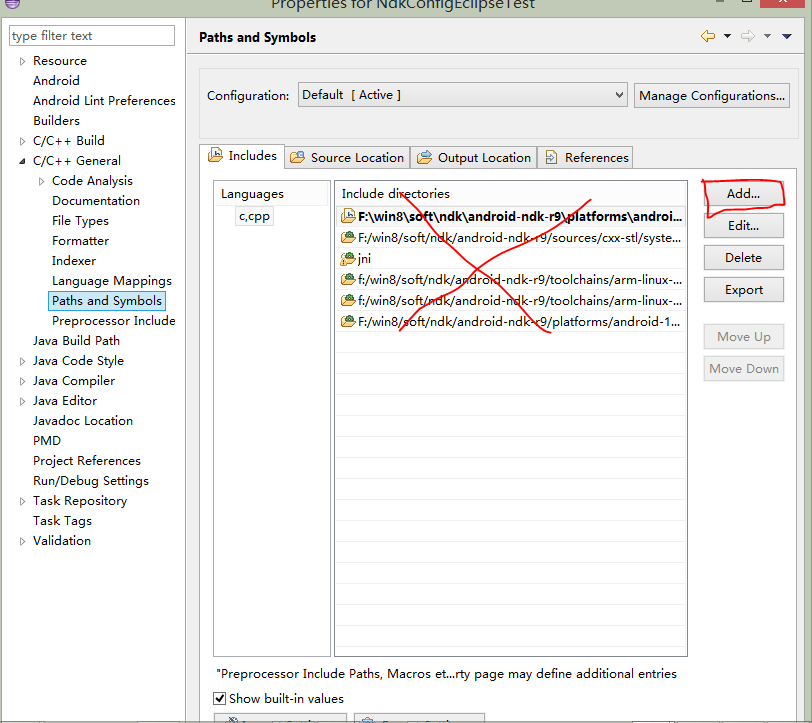




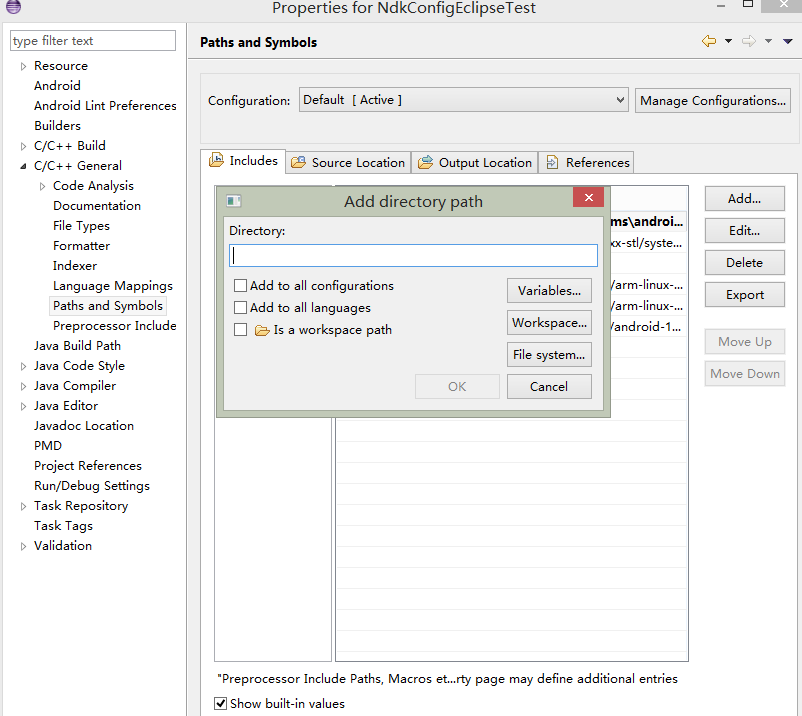
4.配置eclipse添加native支持jni方法代码提示

(1)右击选择项目->Android Tools->Add Native Suport...

(2)右击选择项目-Propreties->...

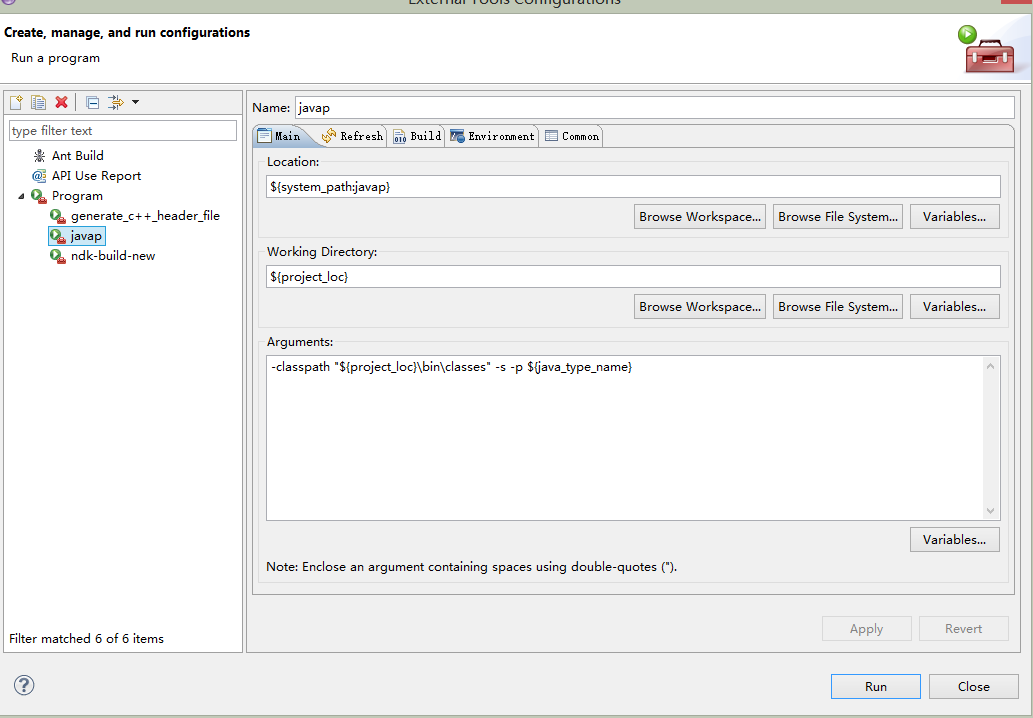


下面选择ndk对应的include文件夹



5.配置生成javap选项

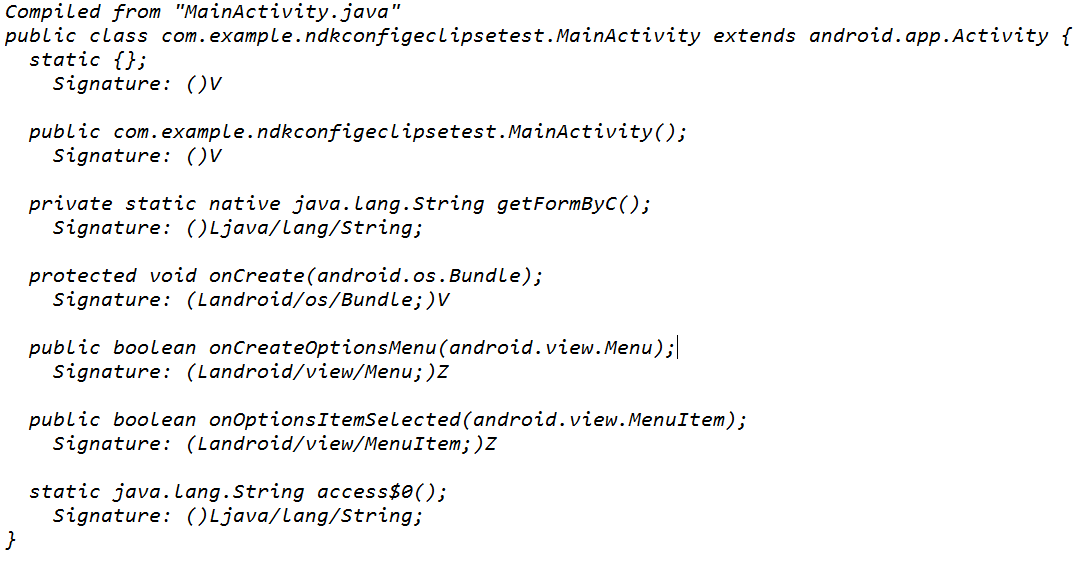
回调方法签名，主要用于c/c++回调java方法



-s:表示打印结果；

-p:表示打印出类中-private的方法

运行结果：



# 4.JNI的交互处理

1.java层和native层进行字符串的交互处理

2.数组的处理：

（1）第一种方式是生成native层的数组拷贝

（2）第二种方式是直接调用数组指针进行操作