

# NDK JNI

说到 NDK 开发，其实是为了有些时候为了项目需求需要调用底层的一些 C/C++ 的一些东西；另外就是为了效率更加高些。

但是很多时候能不用就不用；这个是啥原因？个人感觉有些时候是觉得麻烦，首先要配置 NDK 还要 下载 Cygwin ，配置 Cygwin ,然后需要编译生成，哎太麻烦了。如果你说使用 Eclipse，的确这样在编译上是能简单很多，但是前期也是需要配置一大堆；哎，配置错了还伤不起。

现在谷歌主力推动 Android Studio 这个是个不错的工具，至少现在来看在 Eclipse 上能做到的 这边都有，这边有的Eclipse上不一定有。

下面来说说 在 Android Studio 中怎么进行 NDK 的开发。

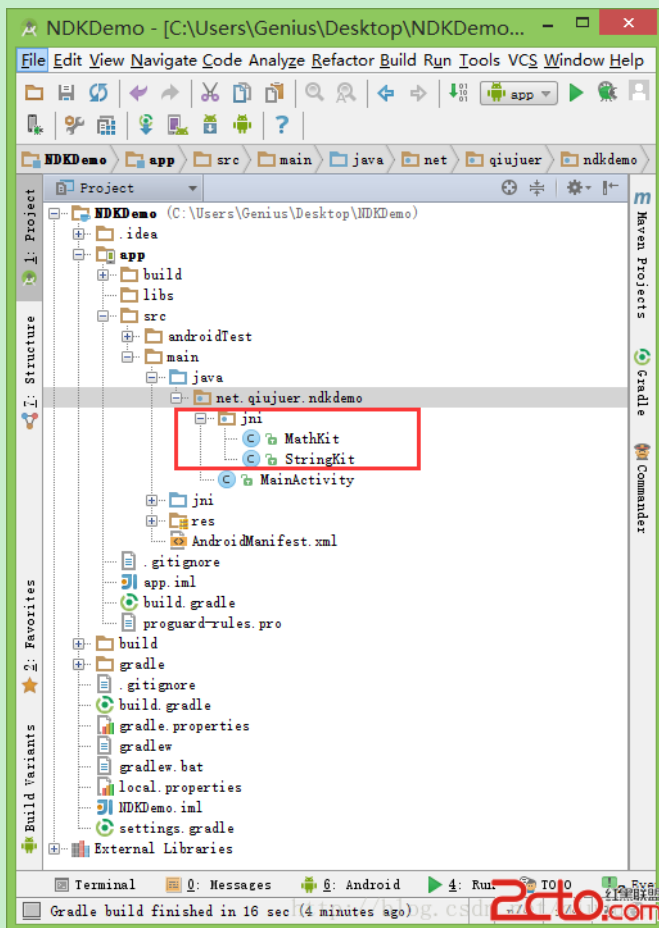
## 准备工作

### NDK 环境搭建

下载：<http://developer.android.com/tools/sdk/ndk/index.html>Windows32：android-ndk-r10d-windows-x86.exeWindows64：android-ndk-r10d-windows-x86\_64.exeMac OS X 32-bit：android-ndk-r10d-darwin-x86.binMac OS X 64-bit：android-ndk-r10d-darwin-x86\_64.binLinux 32-bit (x86)：android-ndk-r10d-linux-x86.binLinux 64-bit (x86)：android-ndk-r10d-linux-x86\_64.bin

都是可执行文件，下载后点击即可解压到当前文件夹；当然你可以使用命令行来进行跨文件夹解压的。

## 建立一个项目 NDKDemo：



建立对应的类，在 MathKit 中我们实现计算INT 值的平方数。

```
public class MathKit {
    public static native int square(int num);

    static {
        System.loadLibrary(JniDemo);
    }
}
```

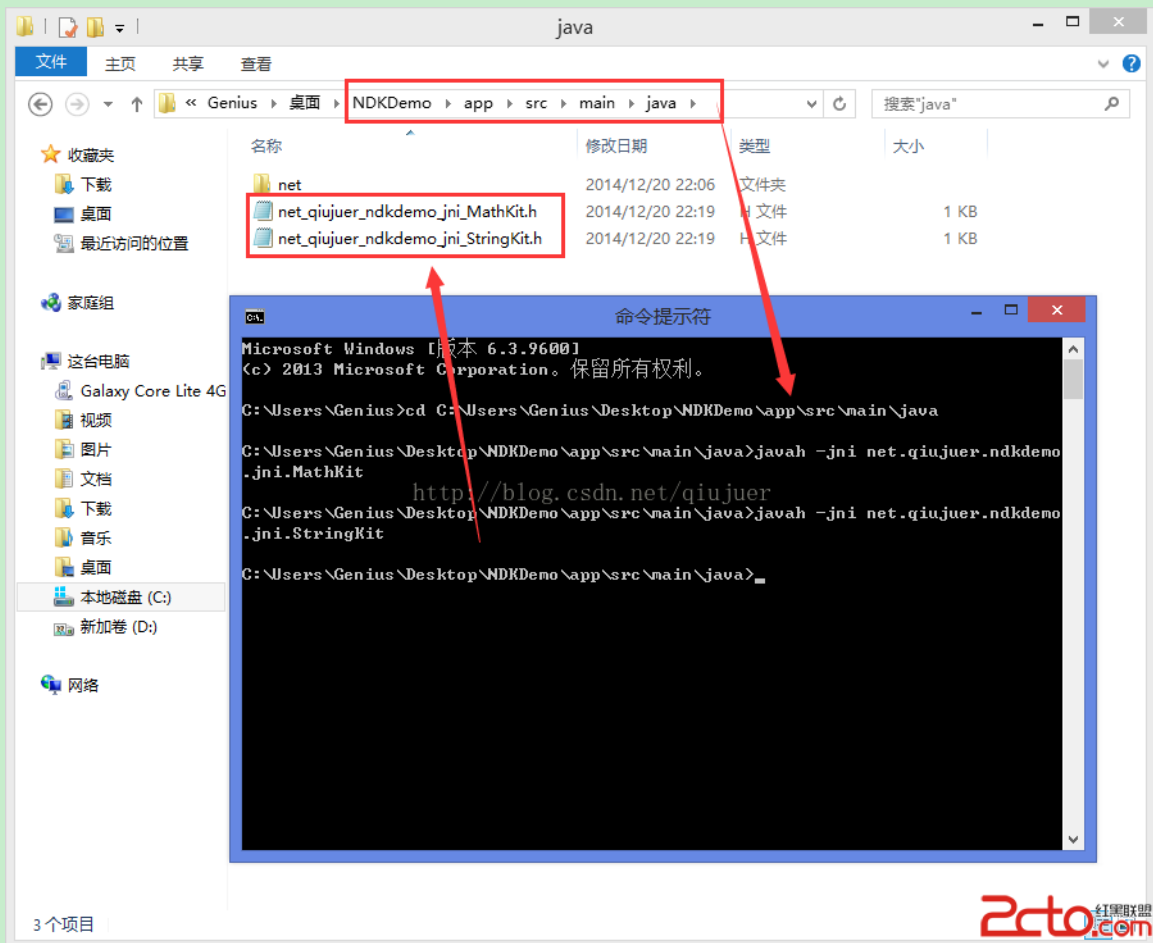
StringKit 这个就是充数用的：

```
public class StringKit {
    public static native void setNull(String str);
}
```

```
static {  
    System.loadLibrary(JniDemo);  
}  
}
```

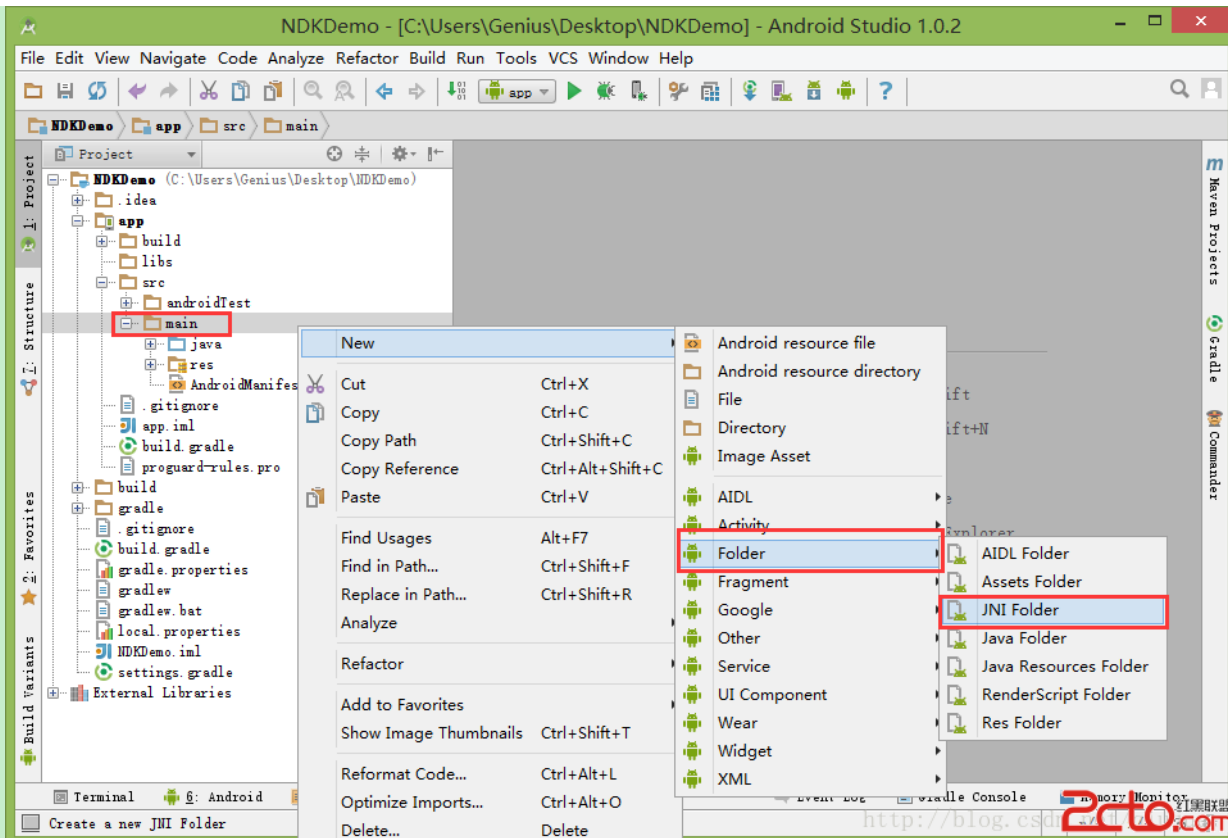
## 准备 .h 文件

进入Java目录；CMD 进入该目录进行 javah 操作：

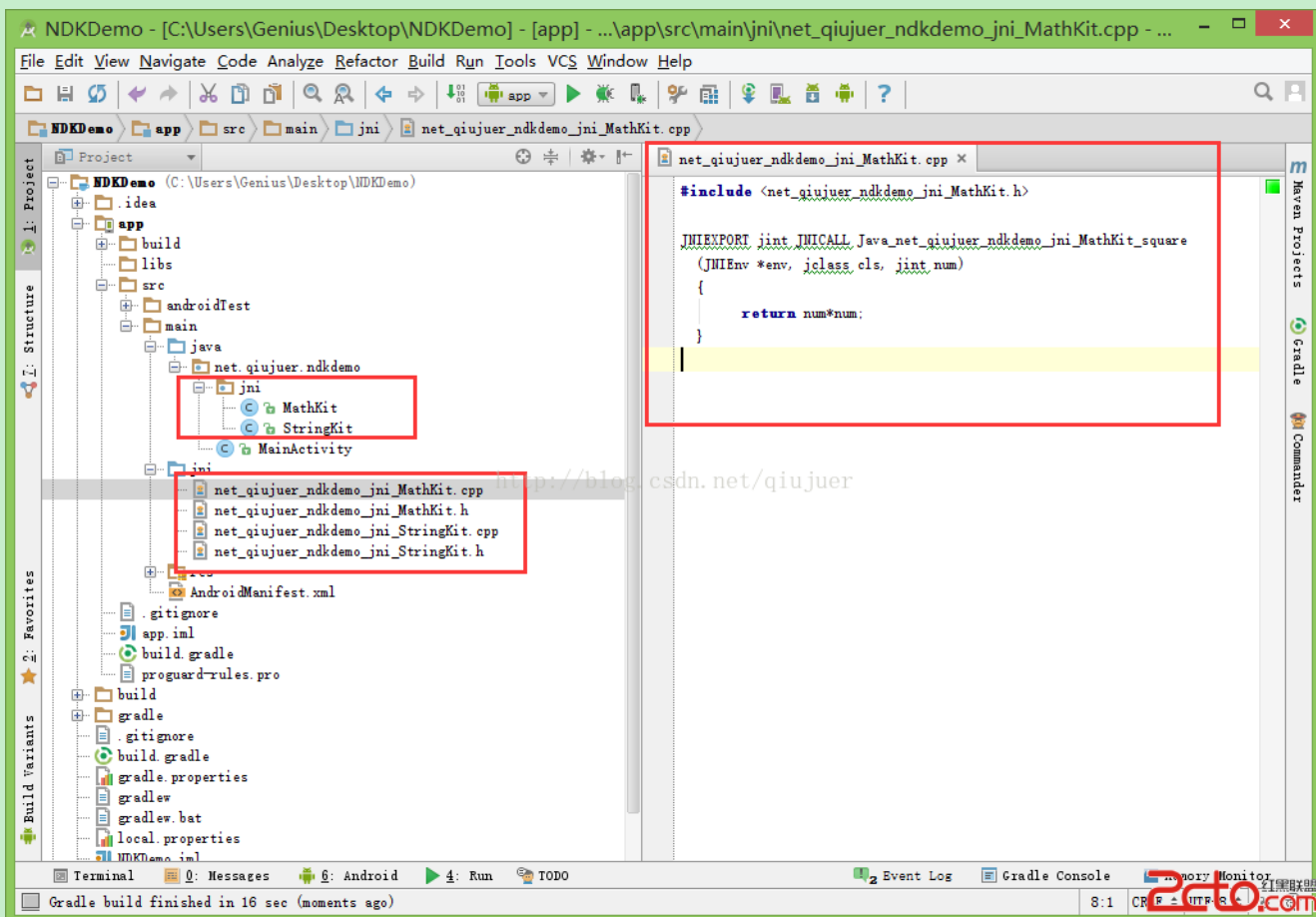


这个想必大家并不陌生吧，该文件就是 java 类对应的 头文件，当然如果你比较厉害的话可以不用这样生成，自己敲代码完成也行；反正我是不行。

如下，建立文件夹：



拷贝之前的文件进去：



分别实现：

net\_qiujuer\_ndkdemo\_jni\_MathKit.cpp

#include

```
JNIEXPORT jint JNICALL Java_net_qiujuer_ndkdemo_jni_MathKit_square
(JNIEnv *env, jclass cls, jint num)
{
    return num*num;
}
```

```

}
.h>

net_qiujuer_ndkdemo_jni_StringKit.cpp

#include

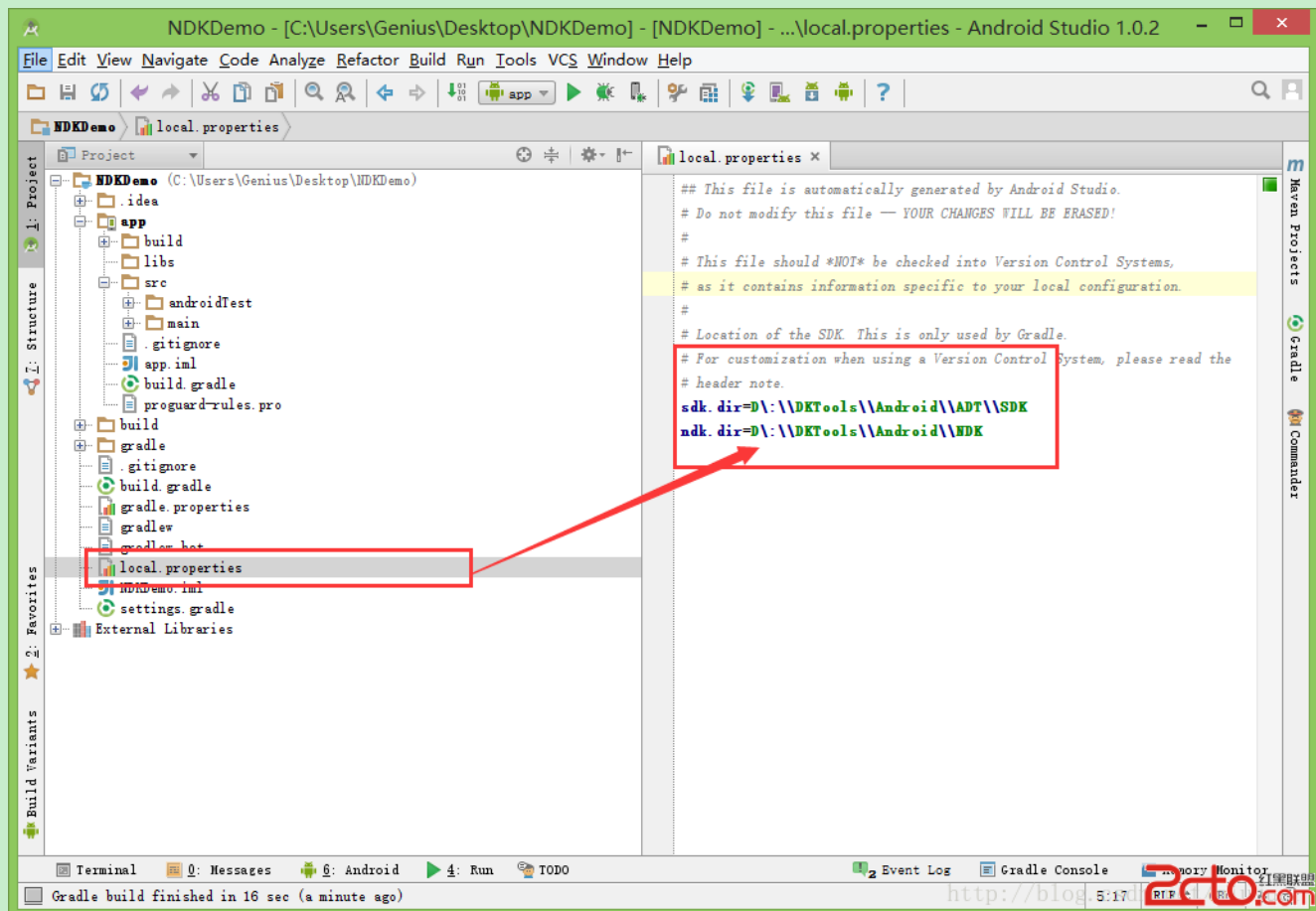
JNIEXPORT void JNICALL Java_net_qiujuer_ndkdemo_jni_StringKit_setNull
(JNIEnv *env, jclass obj, jstring str)
{

}
.h>

```

## 设置项目

设置根目录中的 : local.properties



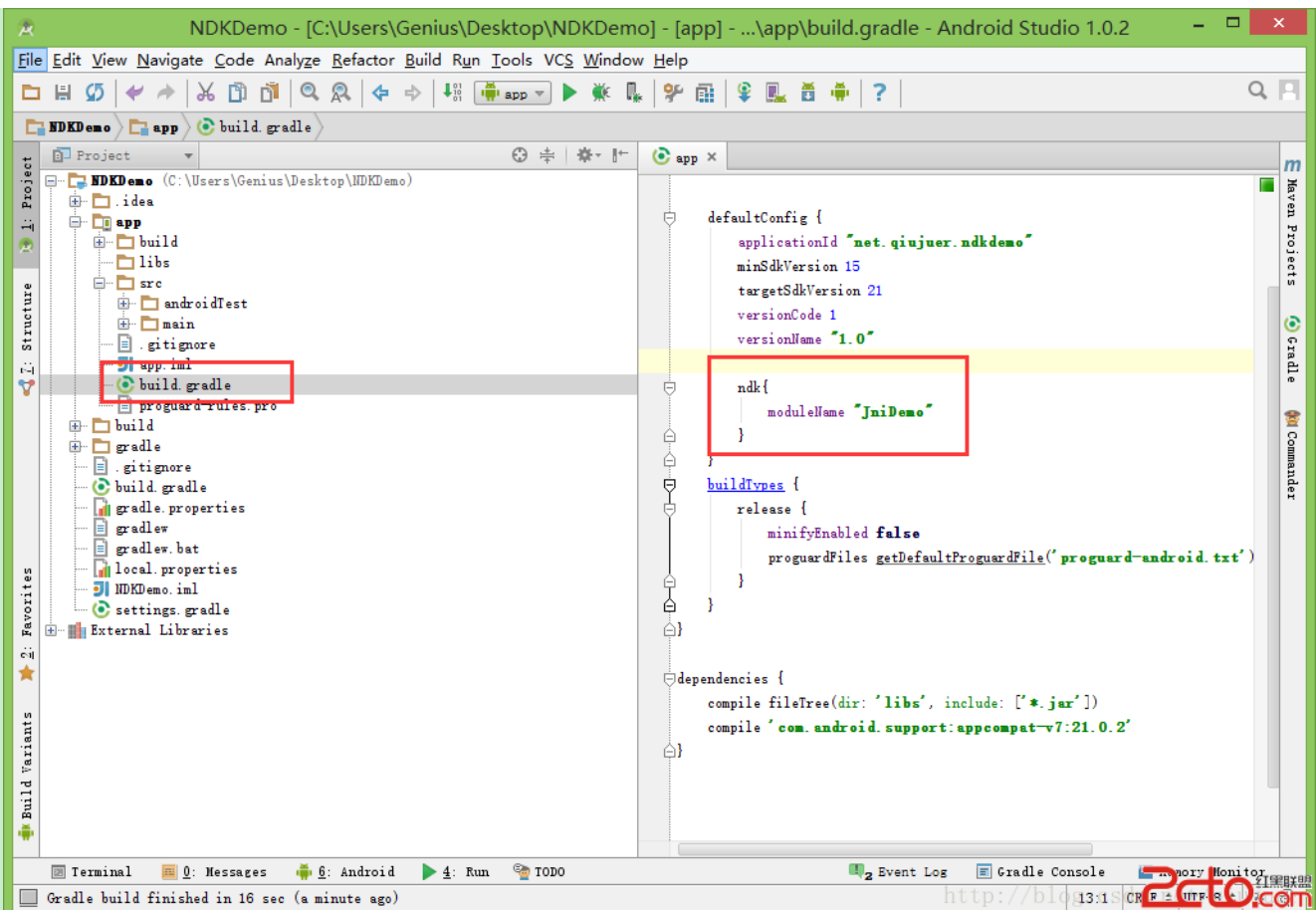
该设置不是必须，如果你设置了环境变量则可不用设置；如果没有但是又没有设置将会出现：

```

! Execution failed for task ':library:compileReleaseNdk'.
> NDK not configured.
Download the NDK from http://developer.android.com/tools/sdk/ndk/. Then add ndk.dir=path/to/ndk in your local.properties file.
(On Windows, make sure you escape backslashes, e.g. C:\ndk rather than C:\ndk)

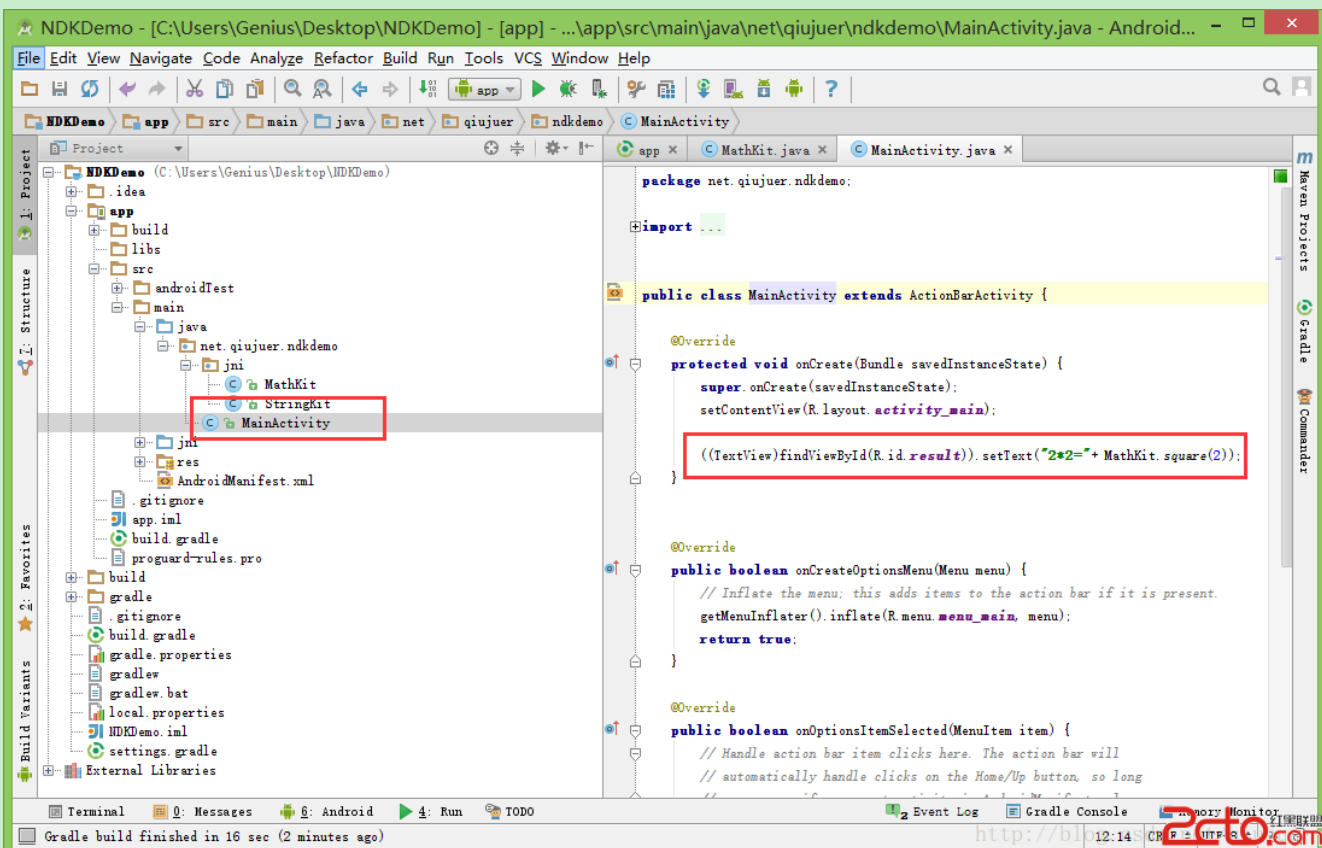
```

设置 APP 项目 build.gradle

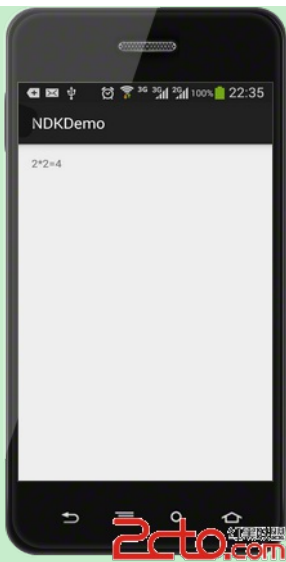


该设置是为了设置生成的Jni名称为：JniDemo。

主文件内容：



现在点击运行：



可以看见是非常简单的；这个调用可以说是我操作过最简单的了；甚至于生成 .H 文件都可以编辑 .sh 文件来进行批处理（这个后面有时间再说）。

## 更多

在平时的那些操作中我们不可避免的要建立：

Android.mk

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE := genius
LOCAL_SRC_FILES := net_qiujuer_imageblurring_jni_ImageBlur.cpp
LOCAL_LDLIBS := -lm -llog -ljnigraphics

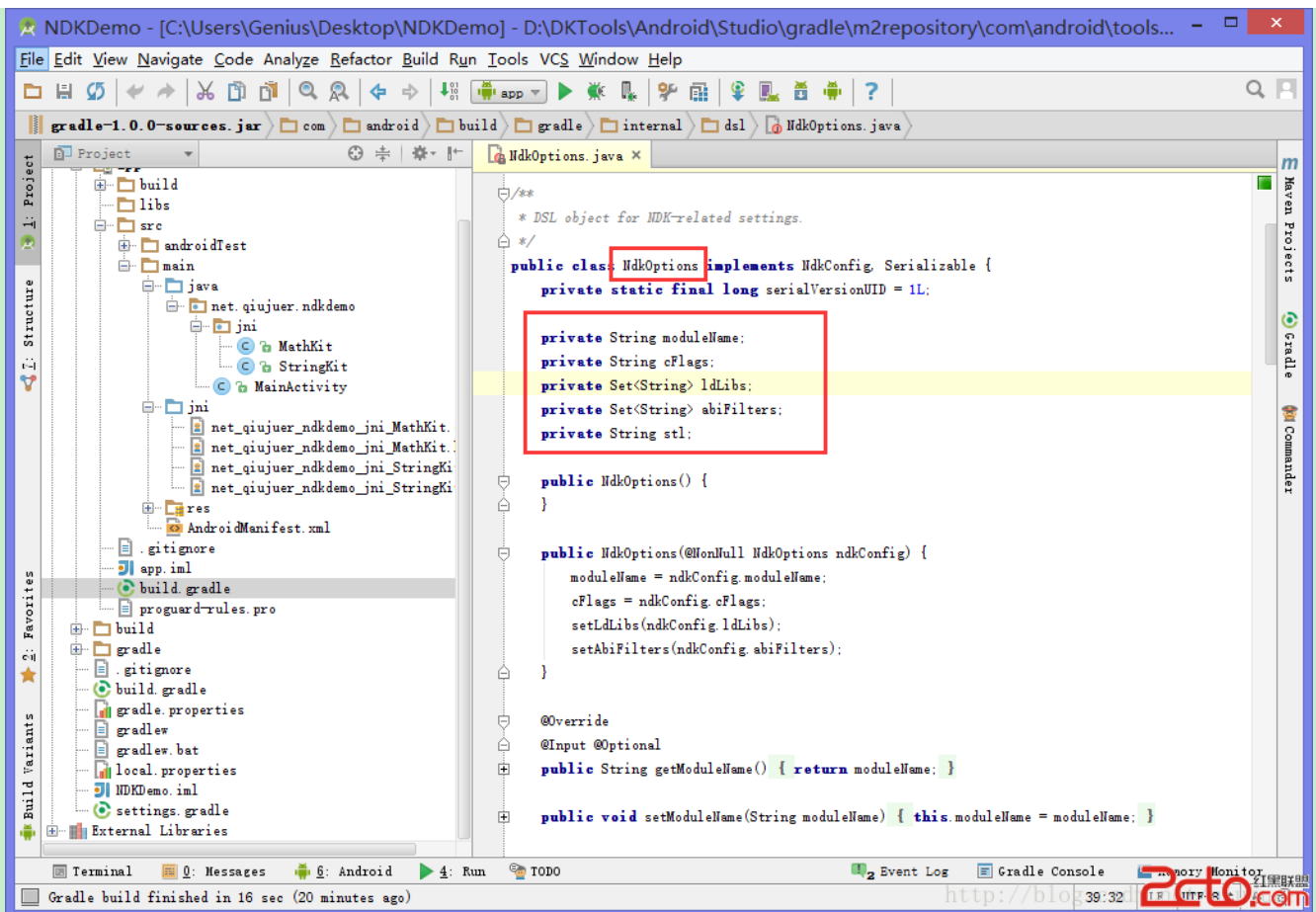
include $(BUILD_SHARED_LIBRARY)
```

Application.mk

```
APP_ABI := all
APP_PLATFORM:= android-19
APP_OPTIM := release
```

而这里这两个文件都没有建立，那么是怎么决定的呢？

我们可以在 ndk 的 modelName 上按住 Ctrl 进行点击将会出现这个界面：



这些就是全部能够设置的参数。分别对应：

```
ndk{
    moduleName JniDemo // 设置生成的文件名
    cFlags -DANDROID_NDK -D RELEASE //这个对应的是生成模式
    ldLibs m, log, jnigraphics //这个对应 LOCAL_LDLIBS 但是需要去掉前面的 "l"
    abiFilters all //这个是对应的 APP_ABI
    stl stlport_shared //这个对应的是 APP_STL
}
```

好了，就这些祝大家成功。项目打包上传。