

NDK静态注册native方法

NDK工程创建与SDK一样，创建完之后需要增加ndk支持编写步骤

1.在工程根目录下找到local.properties,增加NDK目录
ndk.dir=D:\Android\sdk\ndk-bundle

2.在工程根目录下找到gradle.properties,增加支持ndk
android.useDeprecatedNdk=true

3.在app目录下找到build.gradle,在defaultConfig中增加对ndk模块的描述
ndk{

```
//生成的so文件名称  
moduleName"ndk_lib"  
//输出三种平台的so库  
abiFilters"armeabi","armeabi-v7a","x86"  
//abiFilter.add(armeabi)
```

}

4.新建类

```
public class NdkLib{  
    //加载ndk库  
    static{  
        System.loadLibrary("ndk_lib");  
    }  
    //定义ndk库中的导出函数  
    public native String getNativeString();  
}
```

5.在getNativeString方法上使用alt+enter快捷键，自动创建jni目录。导入NDK代码

6.编写NDK代码，修改返回字符串

```
#include<jni.h>  
JNIEXPORT jstring JNICALL  
Java_com_bluelesson_hellondk_NdkLib_getNativeString  
(JNIEnv *env, jobject instance)  
{  
    return (*env)->NewStringUTF(env,"hello NDK 哈哈")  
}
```

```
//  
  
#include<jni.h>  
extern "C" JNIEXPORT jstring JNICALL //cpp文件要加extern "C"  
Java_com_example_laiwenjie_laiwenjiejni_getName  
(JNIEnv *env, jobject instance)  
{  
    //return (*env)->NewStringUTF(env,"hell NDL jah"); C方式  
    return env->NewStringUTF("hell NDL jah"); //C++方式  
}
```

7.在MainActivity类中的onCreate方法中编写使用NDK代码

```
//使用NDK  
NdkLib ndkLib=new NdkLib();  
String string=ndkLib.getNativeString();  
//获取文本框对象  
TextView tv=(TextView)findViewById(R.id.tv);  
//设置新字符串  
tv.setText(string);
```

