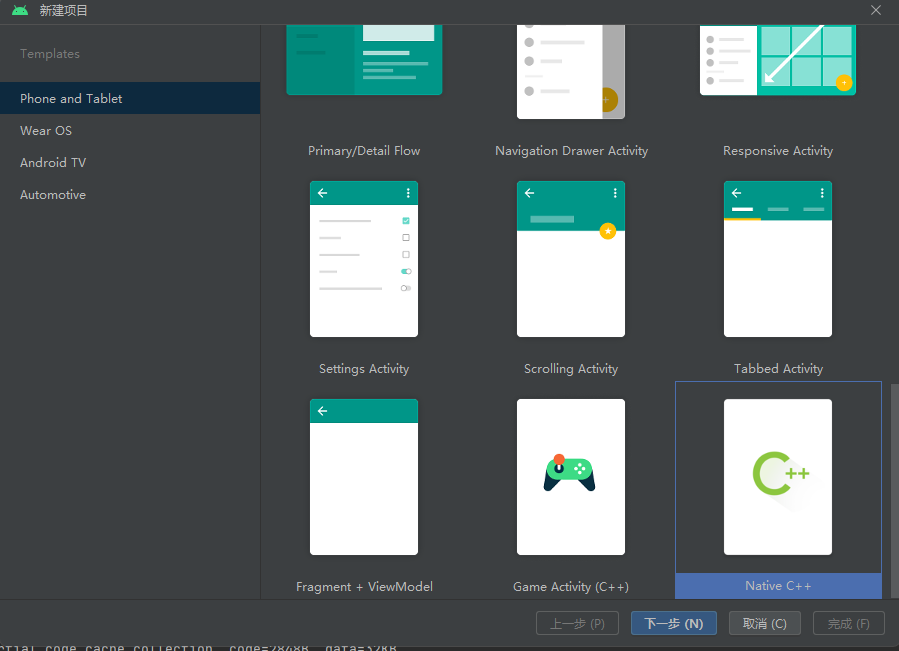
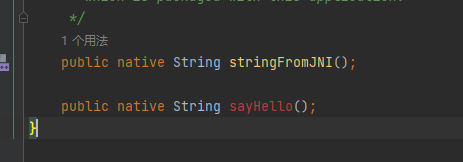
# Android Studio Native手动注册函数记录

我目前使用的环境： jdk19 windows11

第一步 选native 这个没毛病吧？



2．动态注册这个两个函数：



先附上代码 再讲解

#include <jni.h>  
#include <string>  
#include <assert.h>  
#define JNIREG\_CLASS "com/example/cjni/MainActivity"//指定要注册的类  
extern "C" JNIEXPORT jstring JNICALL  
Java\_com\_example\_cjni\_MainActivity\_stringFromJNI(  
 JNIEnv\* env,  
 jobject obj /\* this \*/) {  
 std::string hello = "Hello from C++";  
  
 return env->NewStringUTF(hello.c\_str());  
}  
  
extern "C" JNIEXPORT jstring JNICALL  
jiangsikang(  
 JNIEnv\* env,  
 jobject obj /\* this \*/) {  
 std::string hello = "Hello from C++";  
 return env->NewStringUTF(hello.c\_str());  
}  
extern "C" JNIEXPORT jstring JNICALL  
okok(  
 JNIEnv\* env,  
 jobject obj /\* this \*/) {  
 std::string hello = "Hello from C++";  
 return env->NewStringUTF(hello.c\_str());  
}  
  
static JNINativeMethod gmethod[] = {  
 {"stringFromJNI","()Ljava/lang/String;",(void\*)jiangsikang},  
 {"sayHello","()Ljava/lang/String;",(void\*)okok},  
};  
static int registerNativeMethods(JNIEnv\* env, const char\* className,  
 JNINativeMethod\* gmethod, int numMethods)  
{  
 jclass clazz;  
 clazz = env->FindClass(className);  
 if (clazz == NULL) {  
 return JNI\_FALSE;  
 }  
 if (env->RegisterNatives(clazz, gmethod, numMethods) < 0) {  
 return JNI\_FALSE;  
 }  
  
 return JNI\_TRUE;  
}  
static int registerNatives(JNIEnv\* env)  
{  
 if (!registerNativeMethods(env,JNIREG\_CLASS, gmethod,  
 sizeof(gmethod) / sizeof(gmethod[0])))  
 return JNI\_FALSE;  
  
 return JNI\_TRUE;  
}  
JNIEXPORT jint JNICALL JNI\_OnLoad(JavaVM\* vm, void\* reserved)  
{  
 JNIEnv\* env = NULL;  
 jint result = -1;  
  
 if (vm->GetEnv((void\*\*) &env, JNI\_VERSION\_1\_4) != JNI\_OK) {  
 return -1;  
 }  
 assert(env != NULL);  
  
 if (!registerNatives(env)) {//注册  
 return -1;  
 }  
  
 result = JNI\_VERSION\_1\_4;  
  
 return result;  
}

完毕

## 讲解：

动态注册主要的方法是JNI 里面的

RegisterNatives 需要三个参数

需要的参数 第一个是class 用JNI里面的findClass 方法得到class 即可

第二个参数是、

JNINativeMethod 的结构体 第一个就是 定位方法名，第二个就是方法签名，第三个就是注册的函数指针

类似：

第三个参数：

注册函数的个数