Java native interface

JNI技术使用的原因：

1.你的Java代码，需要得到一个文件的属性。但是你找遍了JDK帮助文档也找不到相关的API。

2.在本地还有一个别的系统，不过他不是Java语言实现的，这个时候你的老板要求你把两套系统整合到一起。

3.你的Java代码中需要用到某种算法，不过算法是用C实现并封装在动态链接库文件（DLL）当中的。

JNIEXPORT void JNICALL Java\_com\_jni\_demo\_JNIDemo\_sayHello (JNIEnv \* env, jobject obj)

JNIENV

NewObject:创建Java类中的对象

NewString:创建Java类中的String对象

New<Type>Array:创建类型为Type的数组对象

Get<Type>Field:获取类型为Type的字段

Set<Type>Field:设置类型为Type的字段的值

GetStatic<Type>Field:获取类型为Type的static的字段

SetStatic<Type>Field:设置类型为Type的static的字段的值

Call<Type>Method:调用返回类型为Type的方法

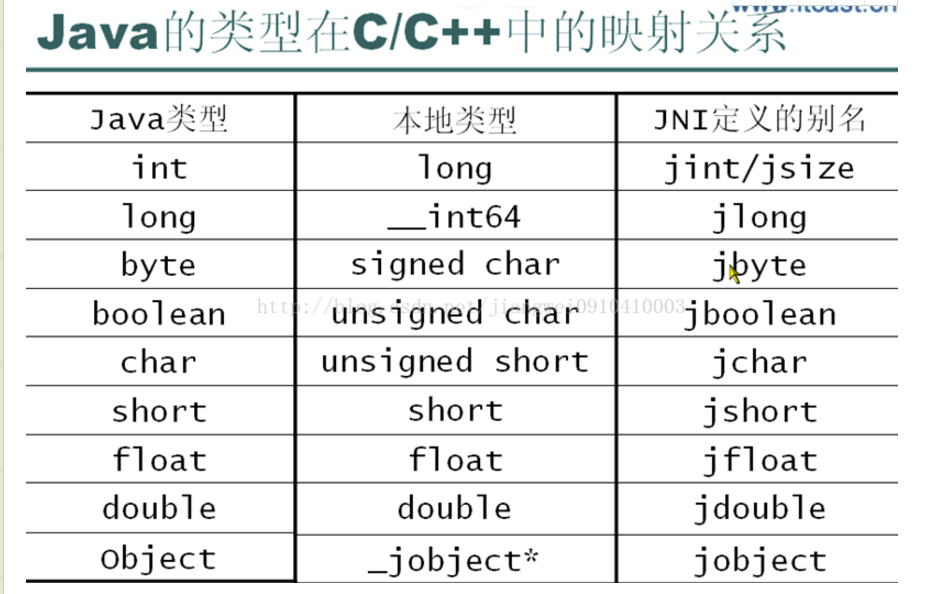
CallStatic<Type>Method:调用返回值类型为Type的static方法

等函数

参数:jobject obj的解释:

如果native方法不是static的话，这个obj就代表这个native方法的类实例

如果native方法是static的话，这个obj就代表这个native方法的类的class对象实例(static方法不需要类实例的，所以就代表这个类的class对象)



jclass GetObjectClass(jobject obj):通过对象实例来获取jclass，相当于java中的getClass方法

jclass GetSuperClass(jclass obj):通过jclass可以获取其父类的jclass对象

GetFieldID(jclass clazz,const char\* name,const char\* sign)

方法的参数说明:

clazz:这个简单就是这个方法依赖的类对象的class对象

name:这个是这个字段的名称

sign:这个是这个字段的签名(我们知道每个变量，每个方法都是有签名的)

怎么查看类中的字段和方法的签名:



GetFieldID/GetMethodID

GetStaticFieldID/GetStaticMethodID

