LZ看看百科中的例子吧  
书写步骤 ·编写带有native声明的方法的java类  
·使用[javac](http://www.so.com/s?q=javac&ie=utf-8&src=wenda_link)命令编译所编写的java类  
JNI  
·使用javah ?jni java类名生成扩展名为h的头文件  
·使用[C/C++](http://www.so.com/s?q=C%2FC%2B%2B&ie=utf-8&src=wenda_link)实现本地方法  
·将C/C++编写的文件生成动态连接库  
·ok  
1) 编写java程序:这里以[HelloWorld](http://www.so.com/s?q=HelloWorld&ie=utf-8&src=wenda_link)为例。  
代码1:  
class HelloWorld {  
public native void displayHelloWorld();//所有native关键词修饰的都是对本地的声明  
static {  
System.loadLibrary("hello"); //载入本地库  
}  
public static void main(String[] args) {  
new HelloWorld().displayHelloWorld();  
}  
}  
声明native方法:如果你想将一个方法做为一个本地方法的话，那么你就必须声明该方法为native的，并且不能实现。其中方法的参数和返回值在后面讲述。 Load动态库:System.loadLibrary("hello");加载动态库(我们可以这样理解:我们的方法 displayHelloWorld()没有实现，但是我们在下面就直接使用了，所以必须在使用之前对它进行初始化)这里一般是以static块进行加载的。同时需要注意的是System.loadLibrary();的参数“hello”是动态库的名字。  
2) 编译  
没有什么好说的了 javac HelloWorld.java  
3) 生成扩展名为h的头文件 javah ?  
jni HelloWorld 头文件的内容: /\* DO NOT EDIT THIS FILE - it is machine generated \*/  
1. include  
/\* Header for class HelloWorld \*/  
1. ifndef \_Included\_HelloWorld  
2. define \_Included\_HelloWorld  
JNI  
3. ifdef \_\_cplusplus  
extern "C" {  
1. endif  
/\*  
\* Class: HelloWorld  
\* Method: displayHelloWorld  
\* Signature: ()V  
\* /  
JNIEXPORT void JNICALL Java\_HelloWorld\_displayHelloWorld (JNIEnv \*, jobject);  
1. ifdef \_\_cplusplus  
}  
1. endif  
2. endif  
(这里我们可以这样理解:这个h文件相当于我们在java里面的接口，这里声明了一个 Java\_HelloWorld\_displayHelloWorld (JNIEnv \*, jobject);方法，然后在我们的本地方法里面实现这个方法，也就是说我们在编写C/C++程序的时候所使用的方法名必须和这里的一致)。  
4) 编写本地方法实现和由javah命令生成的头文件里面声明的方法名相同的方法。  
代码2:  
1 #include "jni.h"  
2 #include "HelloWorld.h"  
JNI  
3 //#include other headers  
4 JNIEXPORT void JNICALL Java\_HelloWorld\_displayHelloWorld(JNIEnv \*env, jobject obj)  
{  
printf("Hello world!\n");  
return;  
}  
注意代码2中的第1行，需要将jni.h(该文件可以在%JAVA\_HOME%/include文件夹下面找到)文件引入，因为在程序中的JNIEnv、 jobject等类型都是在该头文件中定义的;另外在第2行需要将HelloWorld.h头文件引入(我是这么理解的:相当于我们在编写java程序的时候，实现一个接口的话需要声明才可以，这里就是将HelloWorld.h头文件里面声明的方法加以实现。当然不一定是这样)。然后保存为 HelloWorldImpl.c就ok了。  
5) 生成动态库  
这里以在[Windows](http://www.so.com/s?q=Windows&ie=utf-8&src=wenda_link)中为例，需要生成dll文件。在保存HelloWorldImpl.c文件夹下面，使用VC的[编译器](http://www.so.com/s?q=%E7%BC%96%E8%AF%91%E5%99%A8&ie=utf-8&src=wenda_link)cl成。 cl -I%java\_home%\include -I%java\_home%\include\win32 -LD HelloWorldImp.c -Fehello.dll 注意:生成的dll文件名在选项-Fe后面配置，这里是hello，因为在HelloWorld.java文件中我们loadLibary的时候使用的名字是hello。当然这里修改之后那里也需要修改。另外需要将-I%java\_home%\include -I%java\_home%\include\win32参数加上，因为在第四步里面编写本地方法的时候引入了jni.h文件。  
6) 运行程序 java HelloWorld就ok.