



链滴

# Java 中 native 关键字

作者: [huihui](#)

原文链接: <https://ld246.com/article/1482222939991>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

native关键字说明其修饰的方法是一个原生态方法，方法对应的实现不是在当前文件，而是在用其他语言（如C和C++）实现的文件中。Java语言本身不能对操作系统底层进行访问和操作，但是可以通过JNI接口调用其他语言来实现对底层的访问。

JNI是Java本机接口（Java Native Interface），是一个本机编程接口，它是Java软件开发工具箱（Java Software Development Kit, SDK）的一部分。JNI允许Java代码使用以其他语言编写的代码和代码库。Invocation API（JNI的一部分）可以用来将Java虚拟机（JVM）嵌入到本机应用程序中，从而允许程序员从本机代码内部调用Java代码。

不过，对Java外部的调用通常不能移植到其他平台，在applet中还可能引发安全异常。实现本地代码使您的Java应用程序无法通过100%纯Java测试。但是，如果必须执行本地调用，则要考虑几个准则：

1.将您的所有本地方法都封装到一个类中，这个类调用单个的DLL。对每一种目标操作系统平台，都以用特定于适当平台的版本的DLL。这样可以将本地代码的影响减少到最小，并有助于将以后所需要移植问题考虑在内。

2.本地方法尽量简单。尽量使您的本地方法对第三方（包括Microsoft）运行时DLL的依赖减少到最小使您的本地方法尽量独立，以将加载您的DLL和应用程序所需的开销减少到最小。如果需要运行时DLL，必须随应用程序一起提供。

JNI的书写步骤如下：

- a.编写带有native声明的方法的Java类
- b.使用javac命令编译编写的Java类
- c.使用java -jni \*\*\*\*来生成后缀名为.h的头文件
- d.使用其他语言（C、C++）实现本地方法
- e.将本地方法编写的文件生成动态链接库

以下是一个在Java中调用本地C程序的简单的例子：

- a.编写HelloWorld.java类

```
class HelloWorld{  
  
public native void hello();  
  
static{  
System.loadLibrary("hello");  
}  
  
public static void main(String[] args){  
  
new HelloWorld().hello();  
}  
}
```

- b.编译

```
javac HelloWorld.java
```

c.生成.h文件

```
javah -jni HelloWorld
```

生成内容如下:

```
/* DO NOT EDIT THIS FILE - it is machine generated /
#include
/ Header for class HelloWorld */
#ifndef _Included_HelloWorld
#define _Included_HelloWorld
#ifdef __cplusplus
extern "C" {
#endif
/*
• Class: HelloWorld
• Method: hello
• Signature: ()V
*/
JNIEXPORT void JNICALL Java_HelloWorld_hello
(JNIEnv *, jobject);
#ifdef __cplusplus
}
#endif
#endif
```

第一个参数是调用JNI方法时使用的JNI Environment指针。第二个参数是指向在此Java代码中实例的Java对象HelloWorld的一个句柄。其他参数是方法本身的参数

d.c实现

```
#include
#include "HelloWorld.h"
#include
JNIEXPORT void JNICALL Java_HelloWorld_hello(JNIEnv *env,jobject obj){
printf("Hello World!\n");
return;
}
```

其中, 第一行是将jni.h文件引入(在%JAVA\_HOME%\include目录下), 里边有JNIEnv和jobject的义。

## e. 编译c实现

这里以在Windows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。

```
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loadLibrary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了!
```

例子是照着别人的网页做的，其他还有很多东西，可以看资源连接。

资源：

IBM网站的一篇文章：<http://www.ibm.com/developerworks/cn/java/jnimthds/>

china itpub一篇文章：<http://java.chinaitlab.com/JDK/36677.html>

<http://java.chinaitlab.com/JDK/36678.html>

J2SE5的Java Native Interface Specification

<http://java.sun.com/j2se/1.5.0/docs/guide/jni/spec/jniTOC.html>