# Equals方法

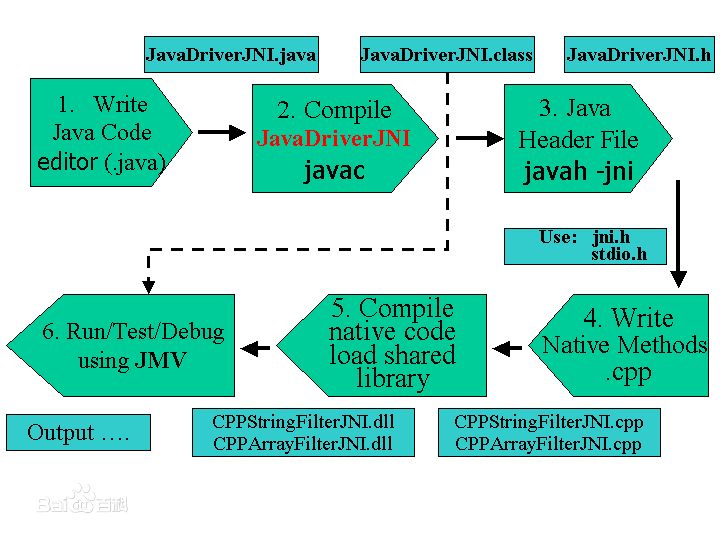
## 陷阱1：定义错误equals方法签名(signature)

# JNI

## 概述

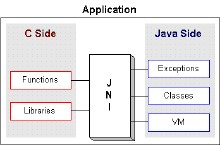
JNI是Java Native Interface的缩写，中文为JAVA本地调用。从Java1.1开始，Java Native Interface(JNI)标准成为[java](http://baike.baidu.com/view/29.htm)平台的一部分，它允许Java代码和其他语言写的代码进行交互。JNI一开始是为了本地已编译语言，尤其是C和C++而设计的，但是它并不妨碍你使用其他语言，只要调用约定受支持就可以了。

使用[java](http://baike.baidu.com/view/29.htm)与本地已编译的代码交互，通常会丧失平台可移植性。但是，有些情况下JNI这样做是可以接受的，甚至是必须的。例如，使用一些旧的库，与硬件、[操作系统](http://baike.baidu.com/view/880.htm)进行交互，或者为了提高程序的性能。JNI标准至少保证[本地代码](http://baike.baidu.com/view/3555594.htm)能工作在任何Java [虚拟机](http://baike.baidu.com/view/1132.htm)实现下。



## 设计目的

标准的[java](http://baike.baidu.com/view/29.htm)类库可能不支持你的程序所需的特性。

[](http://baike.baidu.com/picview/1272329/1272329/0/cf5a83169073a210f2de326f.html)

或许你已经有了一个用其他语言写成的库或程序，而你希望在[java](http://baike.baidu.com/view/29.htm)程序中使用它。

你可能需要用底层语言实现一个小型的时间敏感代码，比如汇编，然后在你的[java](http://baike.baidu.com/view/29.htm)程序中调用这些功能。

## 书写步骤

* 编写带有native声明的方法的[java](http://baike.baidu.com/view/29.htm)类
* 使用[javac](http://baike.baidu.com/view/1006557.htm)命令编译所编写的java类,然后使用javah + java类名生成扩展名为h的头文件
* 使用C/C++实现本地方法
* 将C/C++编写的文件生成[动态连接库](http://baike.baidu.com/view/1366783.htm)

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](http://baike.baidu.com/view/627587.htm).loadLibrary();的参数“hello”是动态库的名字。