* **JNI là gì?**
* Java Native Interface (JNI) là phần nền tảng của Java, là một interface trung gian cho phép kết nối ứng dụng Java với các ứng dụng hoặc thư viện viết bằng ngôn ngữ khác (C, C++).
* JNI cho phép gọi các hàm và truyền các biến qua lại giữa chương trình Java & C/C++.
* Cung cấp quy định chung về dữ liệu và loại exception ...
* Một số khái niệm liên quan:
* Host Environment : là những thiết lập và cài đặt trên máy chủ để chạy ứng dụng  
  JNI.
* Thư viện .dll: thư viện .dll phải đạt được tính độc lập riêng .
* JNI : cầu nối trung gian giữa JAVA & .dll.
* **Khi nào thì sử dụng?**
* Ứng dụng phải truy cập vào các tính năng hệ thống hoặc cá thiệt bị không thể truy cập thông qua nền tảng Java.
* Có một số lượng lớn mã kiểm tra, sửa đổi trong một ngôn ngữ khác và bạn biết cách nào để port tất cả chứng đến nền tảng mong muốn.
* Bạn phát hiện một số trường hợp mã Java là chậm hơn các đoạn mã tương ứng trong các ngôn ngữ khác.
* **Mục đích, lợi ích chính của việc dùng JNI**
* Sử dụng một ứng dụng, thư viện liên kết động như C, C++ vào trong chương trình Java.
* Nhúng máy ảo Java vào các ứng dụng native (C, C++).
* **Một số khó khăn, điểm hại**
* Khi sử dụng JNI thì ứng dụng Java sẽ không còn dễ dàng chạy trên nhiều môi trường nữa.
* Trong khi ngôn ngữ Java là loại an toàn thì các ngôn ngữ native (C, C++) thì không. Do đó, khi có một phương thức trong ứng dụng hay dll được sử dụng bị hỏng thì sẽ làm cho toàn bộ hệ thống bị hỏng. Do vậy, cần phải kiểm tra các thư viện dll, ứng dụng C, C++ trước khi sử dụng JNI
* **Các bước cơ bản để sử dụng JNI cơ bản:**

1. Khai báo một phương thức ***native*** trong một lớp Java.
2. Chạy ***javah*** để có được một ***header*** với một khai báo C cho phương thức.
3. Thực hiện phương thưc ***native*** trong C.
4. Chuyển mã nguồn C thành thư viện dll.
5. Load thư viện trong chương trình Java của bạn và sử dụng.

*Ghi chú*:

* Một số thư viện dll của phương thưc native phải chạy mã khởi tạo. Bạn có thể tạo phương thức JNI\_OnLoad để thực thi mã khởi tạo này. Tương tự như vậy khi máy ảo tắt thì sẽ gọi phương thức JNI\_OnUnload.
* *jint JNI\_OnLoad(JavaVM\* vm, void\* reserved);*
* *void JNI\_OnUnload(JavaVM\* vm, void\* reserved);*
* Trong đó, phương thức JNI\_OnLoad cần phải trả về phiên bản tối thiểu mà máy ảo cần. Ví dụ JNI\_VERSION\_1\_1
* **Ví dụ:** viết chương trình helloworld sử dụng eclipse

1. **Khai báo một phương thức native trong một lớp Java.**

Ta khai báo lớp HelloNative như sau:

class HelloNative

{

public static native void greeting();

}

1. **Chạy javah để có được một header với một khai báo C cho phương thức.**

Mở Run-> External Tools -> External Tools Configurations..

Chọn *New load Configuration* cho *Progam*

Ở tab Main:

* Location: C:\Program Files\Java\jdk1.7.0\_03\bin\javah.exe

Đường dẫn file javah.

* Working Directory: ${workspace\_loc:/JNI\_Demo\_Reg/bin}

Thư mục chứa file header sao khi tạo

* Arguments: -jni JNI\_Demo\_Reg.Win32RegKey

Buil Project để tạo file header.

Nội dung file *HelloNative.h* sao khi tạo

/\* DO NOT EDIT THIS FILE - it is machine generated \*/

#include <jni.h>

/\* Header for class HelloNative \*/

#ifndef \_Included\_HelloNative

#define \_Included\_HelloNative

#ifdef \_\_cplusplus

extern "C" {

#endif

/\*

\* Class: HelloNative

\* Method: greeting

\* Signature: ()V

\*/

JNIEXPORT void JNICALL Java\_HelloNative\_greeting

(JNIEnv \*, jclass);

#ifdef \_\_cplusplus

}

#endif

#endif

1. **Thực hiện phương thưc native trong C.**

Tạo phương thức *HelloNative.c*  có nội dung như sau:

/\*

@version 1.10 1997-07-01

@author Cay Horstmann

\*/

#include "HelloNative.h"

#include <stdio.h>

JNIEXPORT void JNICALL Java\_HelloNative\_greeting(JNIEnv\* env, jclass cl)

{

printf("Hello Native World!\n");

}

1. **Chuyển mã nguồn C thành thư viện dll.**

Buil mã nguồn C thành thư viện liên kết động. Có thể dùng GCC, Microsoft Visual C++.. Chi tiết mời xem demo.

1. **Load thư viện trong chương trình Java của bạn và sử dụng.**

Tạo phương thức HelloNativeTest.java để test có nội dung như sau:

class HelloNativeTest

{

public static void main(String[] args)

{

HelloNative.greeting();

}

Static

{

System.loadLibrary("HelloNative"); //tên của file dll

}

}

Trong lúc thực hiện có một vài ghi chú. Xin xem demo để biết rõ hơn.

Một số ví dụ của sun: <http://java.sun.com/docs/books/jni/>