Field và Method

# Truy xuất Field

JNI cung cấp các hàm để native code truy xuất (get/set) các field của object và các static field của class.

## Truy xuất Field của đối tượng

Để truy xuất vào một Field JNI cung cấp các hàm sau:

(\*env)->GetAAAField(env, jclass, fieldID);

|  |  |
| --- | --- |
| **Get<Type>Field** | **<NativeType>** |
| GetObjectField | jobject |
| GetBooleanField | jboolean |
| GetByteField | jbyte |
| GetCharField | jchar |
| GetShortField | jshort |
| GetIntField | jint |
| GetLongField | jlong |
| GetFloatField | jfloat |
| GetDoubleField | jdouble |

**Parameters**

env: JNIEnv interface pointer.

jclass: trỏ đến class có field được truy xuất.

fieldID: một biến có kiểu jfieldID đại diện cho field được truy xuất.

**Return Values**

Trả về giá trị của field.

jclass là một biến có kiểu jclass có đượng bằng hàm:

jclass cls = (\*env)->GetObjectClass(env, obj);

Với env, obj là tham số đầu vào của hàm JNI.

Để có được fieldID ta gọi hàm:

jfieldID GetFieldID(JNIEnv \*env, jclass clazz,  
const char \*name, const char \*sig);

**Parameters**

env: JNIEnv interface pointer.

jclass: trỏ đến class có field được truy xuất.

name: chuỗi tên field trong java.

sig:  field descriptor – được nói rõ trong mục ngay phía sau.

**Return Values**

Trả về fieldID.

Để **set** giá trị field ta dùng các hàm sau:

**Prototype** void *Set<Type>Field*(JNIEnv \*env, jobject obj,   
jfieldID fieldID,*<NativeType>* value);

|  |  |
| --- | --- |
| **Set<Type>Field** | **<NativeType>** |
| SetObjectField | jobject |
| SetBooleanField | jboolean |
| SetByteField | jbyte |
| SetCharField | jchar |
| SetShortField | jshort |
| SetIntField | jint |
| SetLongField | jlong |
| SetFloatField | jfloat |
| SetDoubleField | jdouble |

Ý nghĩa các trường tương tự hàm get.

## Field Descriptors

Là một chuỗi dùng để biểu diễn kiểu dữ liệu của field.

Với các kiểu cơ bản mô tả sẽ đơn giản hơn như: int có chuỗi mô tả là “I”, float có “F”, double có “D”, boolean có “Z”.

Với các kiểu dữ liệu khác thì mô tả theo package name của nó. Chẳng hạn như kiểu String có thuộc package: java.lang.String thì chuỗi mô tả tương ứng là “Ljava/lang/String;”.

Có một cách khác để biết chuỗi mô tả của các filed trong lớp java là dùng tool javap đi kèm trong JDK với tham số -s -p TenClass

## Truy xuất static field

Static field truy xuất giống như field của đối tượng chỉ là tên các hàm có thay đổi chút ít như thay vì gọi GetFieldID thì ta gọi GetStaticFieldID tương tự đối với các hàm Get/Set dữ liệu.

# Gọi Method

Trong java cũng có vài loại method như method của đối tượng, static method, và các hàm dựng của đối tượng.

## Gọi method của object

Đầu tiên ta cần có MethodId của method cần gọi bằng cách gọi hàm:

jmethodID GetMethodID(JNIEnv \*env, jclass clazz,   
const char \*name, const char \*sig);

Các tham số giống như phần truy xuất Field. Tuy nhiên phần sig sẽ là method descriptor.

Gọi method bằng hàm sau:

**Prototype** <*NativeType>* *Call<Type>Method*(JNIEnv \*env,  
jobject obj, jmethodID methodID, ...);

|  |  |
| --- | --- |
| **Call<Type>Method** | **<NativeType>** |
| CallVoidMethod | void |
| CallObjectMethod | jobject |
| CallBooleanMethod | jboolean |
| CallByteMethod | jbyte |
| CallCharMethod | jchar |
| CallShortMethod | jshort |
| CallIntMethod | jint |
| CallLongMethod | jlong |
| CallFloatMethod | jfloat |
| CallDoubleMethod | jdouble |

Hàm trả về kết quả gọi hàm. Các tham số đã được nói ở phần trên. Tham số của method có thể truyền vào phần sau hàm.

## Method descriptor

Là một chuỗi tương tự như prototype của method nhưng có mô tả khác. Gồm 2 phần “(?)?” Với chấm hỏi thứ nhất là phần các tham số truyền vào method, Chấm hỏi thứ hay là giá trị trả về. Còn nội dụng của ? thì giống với Field descriptor. Lưu ý là void có ký hiệu là “V”. Một vài ví dụ:

“()V”

“(I)I”

“(Ljava/lang/String;)Ljava/lang/String;”

## Gọi static method

Hoàn toàn giống với gọi hàm của object. Chỉ khác nhau ở tên hàm trong JNI như thay vì gọi GetMethodID, CallVoidMethod… thì sẽ gọi GetStaticMethodID, CallStaticVoidMethod…

## Gọi method của một lớp cha của object

Trong JNI có thể gọi hàm của lớp cha (khi dùng đa hình) bằng hàm:

|  |  |
| --- | --- |
| **CallNonvirtual<Type>Method** | **<NativeType>** |
| CallNonvirtualVoidMethod | void |
| CallNonvirtualObjectMethod | jobject |
| CallNonvirtualBooleanMethod | jboolean |
| CallNonvirtualByteMethod | jbyte |
| CallNonvirtualCharMethod | jchar |
| CallNonvirtualShortMethod | jshort |
| CallNonvirtualIntMethod | jint |
| CallNonvirtualLongMethod | jlong |
| CallNonvirtualFloatMethod | jfloat |
| CallNonvirtualDoubleMethod | jdouble |

Việc gọi hàm này tương đương với lời gọi super.f() trong java.

## Tạo dựng đối tượng Java trong JNI

JNI cho phép ta lấy được hàm dựng của một đối tượng qua đó tạo dựng được đối tượng Java.

Cũng giống như gọi method bình thường ta cần có methodId, jclass…

Để có được methodId có thể gọi hàm GetMethodID với name là “<init>” mà chuỗi mô tả method là tùy theo tham số đầu vào của hàm dựng (giá trị trả về luôn là “V” vì hàm dựng không trả về).

Để tạo đối tượng với hàm dựng có được ta gọi hàm:

jobject NewObject(JNIEnv \*env, jclass clazz,   
jmethodID methodID, ...);