

Native

1. 什么是Native Method

2. 为什么要使用Native Method

3. 本地方法栈

1. 什么是Native Method

简单地讲，一个Native Method就是一个java调用非java代码的接口。一个Native Method是这样一个java的方法：该方法的实现由非java语言实现，比如C。这个特征并非java所特有，很多其它的编程语言都有这一机制，比如在C++中，你可以用extern "C"告知C++编译器去调用一个C的函数。

在定义一个native method时，并不提供实现体（有点类似定义一个java interface），因为其实体是由非java语言在外面实现的。

标识符native可以与所有其它的java标识符连用，但是abstract除外。这是合理的，因为native暗示这些方法是有实现体的，只不过这些实现体是非java的，但是abstract却显然的指明这些方法无实现体。native与其它java标识符连用时，其意义同非Native Method并无差别，比如native static表明这个方法可以在不产生类的实例时直接调用，这非常方便，比如当你想用一個native method去调用一个C的类库时。

2. 为什么要使用Native Method

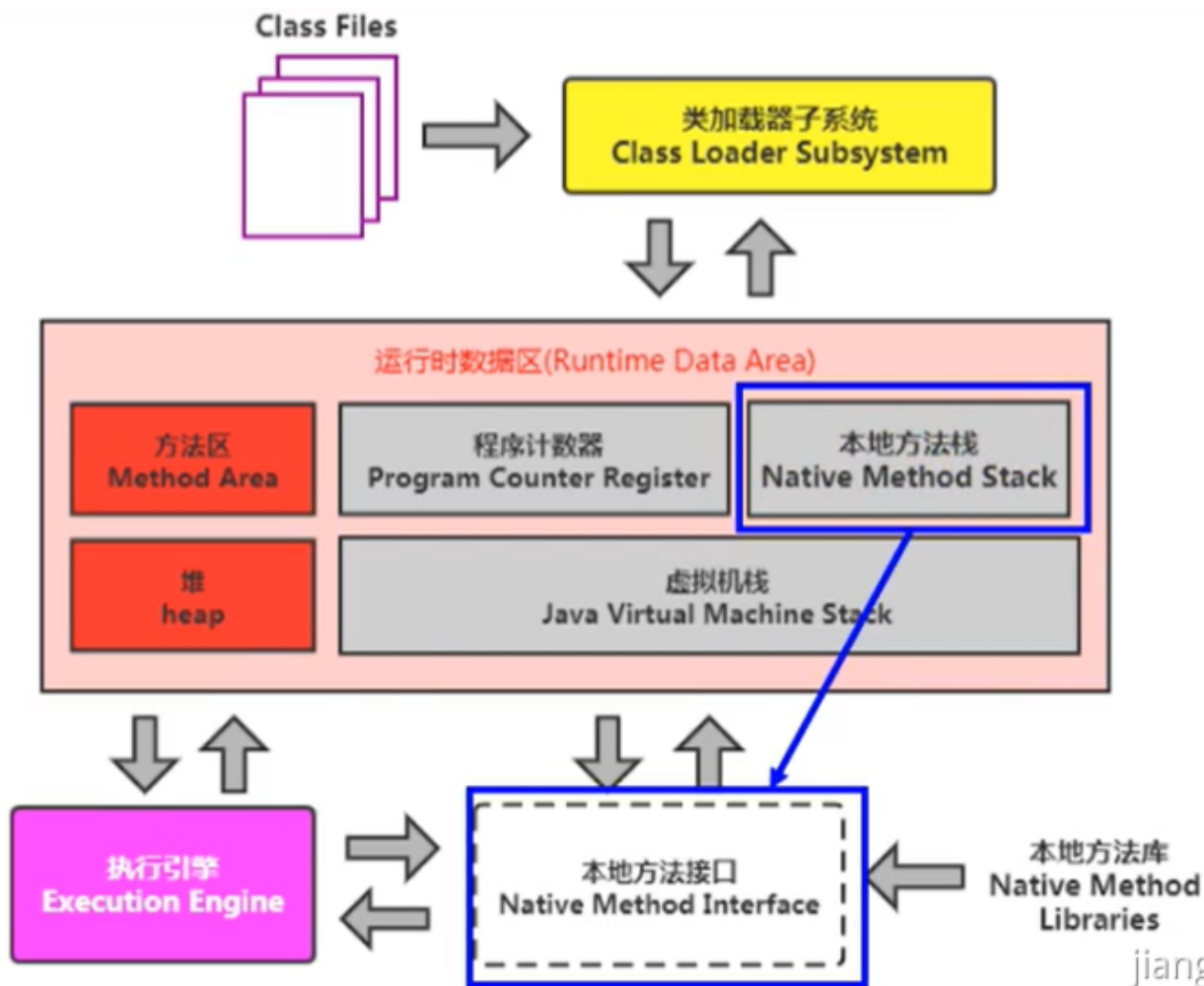
java使用起来非常方便，然而有些层次的任务用java实现起来不容易，或者我们对程序的效率很在意时，问题就来了。

- 与java环境外交互
- 与操作系统交互
- Sun's Java

最后需要提示的是，使用本地方法是有开销的，它丧失了java的很多好处。如果别无选择，我们可以选择使用本地方法。

3. 本地方法栈

java虚拟机栈用于管理java方法的调用，而本地方法栈用于管理本地方法的调用



- 本地方法栈是一个后入先出（Last In First Out）栈,也是线程私有的，生命周期随着线程启动而产生，线程结束而消亡。本地方法是使用C语言实现的。
- 允许被实现成固定或者是可动态扩展的内存大小。（在内存溢出方面是相同的）
- 它的具体做法是Native Method stack中登记native方法,在Execution Engine执行时加载本地方法库

==> 阳哥翻译：

java刚诞生的时候是c/c++横向的时候，为了生存可耻的屈服于他们，给他们交保护费，于是就在内存中专门开辟了一个本地方法栈用于调用c/c++程序。

Native Interface本地接口

本地接口的作用是融合不同的编程语言为 Java 所用，它的初衷是融合 C/C++程序，Java 诞生的时候是 C/C++横行的时候，要想立足，必须有调用 C/C++程序，于是就在内存中专门开辟了一块区域处理标记为native的代码，它的具体做法是 Native Method Stack中登记 native方法，在Execution Engine 执行时加载native libraies。

目前该方法使用的越来越少了，除非是与硬件有关的应用，比如通过Java程序驱动打印机或者Java系统管理生产设备，**在企业级应用中已经比较少见**。因为现在的异构领域间的通信很发达，比如可以使用 Socket通信，也可以使用Web Service等等，不多做介绍。

Native Method Stack

它的具体做法是Native Method Stack中登记native方法，在Execution Engine 执行时加载本地方法库。

jiang.liu