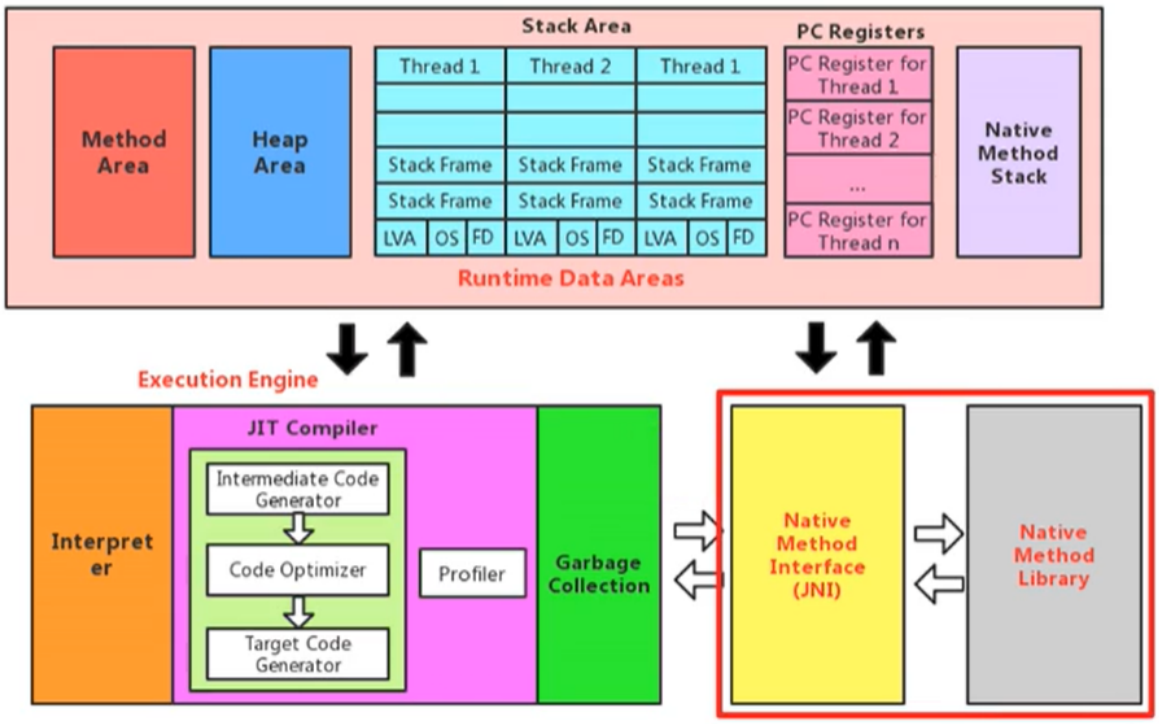
第6章 本地方法接口



- 什么是本地方法？
  - 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，很多其他的编译语言都有这一机制，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。
  - “A native method is a Java method whose implementation is provided by nono-java code.”
  - 在定义一个native method时，并不提供实现体（有些像定义一个Java的interface），因为其实现体是由非Java语言在外面实现的。
  - 本地方法的作用是融合不同编程语言为Java所用，它的初衷是融合C/C++程序。
- 代码演示

```
public class IHaveNatives {
    public native void Native1(int x);

    public native static long Native2();

    private native synchronized float Native3(Object o);

    native void Native4(int[] ary) throws Exception;
}
```

标识符native可以与所有其他的java标识符连用，但是abstract除外。

- 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**
    - **优势Java应用需要与Java外面的环境交互，这是本地方法存在的主要原因**。你可以想想Java需要与一些底层系统，如操作系统或某些硬件交换信息时的情况。本地方法正式这样一种交流机制：它为我们提供了一个非常简洁的接口，而且我们无需去了解Java应用之外的繁琐的细节。
    - **与操作系统交互：**
      - JVM支持着Java语言本身和运行时库，它是Java程序赖以生存的平台，它由一个解释器（解释字节码）和一些连接到本地代码的库组成。然而不管怎样，它毕竟不是一个完整的的系统，它经常依赖于一些底层系统的支持。这些底层系统常常是强大的操作系统。**通过使用这些本地方法，我们得以用Java实现jre的与底层的交互，甚至JVM的一些部分就是C写的**。还有，如果我们要使用一些Java语言本身没有提供封装的操作系统的特性时，我们也需要使用本地方法。
    - Sun's Java
      - **Sun的解释器 使用C实现的，这使得它能像一些普通的C一样与外部交互**。jre大部分是用Java实现的，它也通过一些本地方法与外界交互。例如：类java.lang.Thread的setPriority()方法是用Java实现的，但是它实现的是该类的本地方法setPriority0()。这个本地方法是用C实现的，并被植入JVM内部，在Windows 95的平台上，这个本地方法最终将调用Win32 setPriority()API。这是一个本地方法的具体实现由JVM直接提供，更多的情况是本地方法由外部的动态链接库（external dynamic link library）提供，然后被JVM调用。
- 现状
  - **目前该方法是用的越来越少了，除非是与硬件有关的应用**，比如通过Java程序确定打印机或者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键的通信很发达，比如可以使用Socket通信，也可以使用Web Service等等，不多做介绍。