

如何理解“..\jre\bin\server\jvm.dll是jvm的核心？”

1 Class loader 是什么

Java Class loaders是JRE(Java运行环境, Java Runtime Environment)一部分, Java Class loaders将Java Classes动态地加载到JVM(Java Virtual Machine)中。由于Java Class loader的存在, JVM不需要具体地去了解文件和文件系统就可以运行Java程序。

参考: [初识Class Loader](#)

2 Class loader在哪里加载?

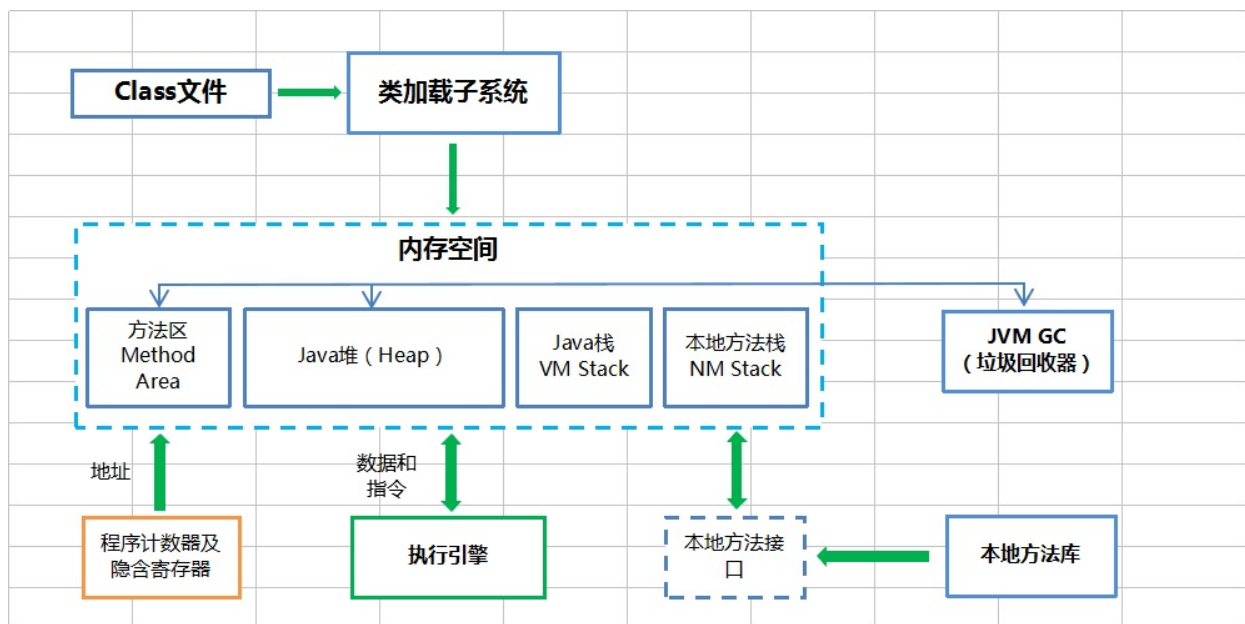
1. 编写*.java文件;
2. 编译成*.class文件 (比如使用: javac) ;
3. 运行编译好的java项目。当开始运行程序, 我们将会创建一个JVM实例;

JVM负责确保Java应用程序拥有在环境中良好运行和执行所需的资源**

4. 加载class loader系统;

因此只有完成了加载Class loader系统后, 才能进一步运行java程序

5.剩余步骤在下图中, 不过对理解本次问题不重。



参考: [了解JRE和JVM](#)

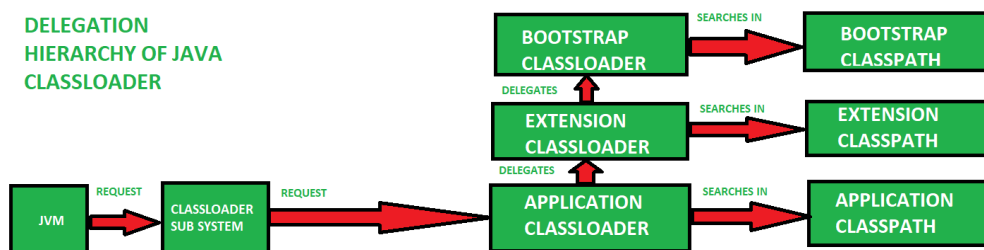
3 Java Bootstrap ClassLoader是什么?

Bootstrap ClassLoader是特定于平台的机器指令（它是由C/C++代码编写的），可启动Java Class loader，让程序正常开始执行。

所以，我们可以简单理解成：

Bootstrap classloader —— (加载) ——> Java类的class loader —— (实现) ——> 程序正常开始执行

如果没有第一步的Bootstrap Classloader加载，程序就无法正常执行



参考: [Java Bootstrap Classloader是如何加载的?](#)

4 如何在Java程序中加载Bootstrap Classloader?

在前面我们已经了解到：为了让java程序正常运行，我们首先要加载class loader系统。为了能够成功加载class loader系统，我们要用到bootstrap classloader。

然而，bootstrap classloader是Non-Java（非Java）程序写的，我们该如何运行呢？

因此，我们就要用到jvm.dll。jvm.dll是JVM的Windows实现（或者说是“主要入口点”）。C或C++应用程序可以使用此DLL来运行嵌入式Java运行时，这将允许该应用程序直接与JVM进行接口。

dll文件的知识补充：（Dynamically linked library又称为native library）

简而言之，native library是非Java代码的Java接口。它是Java与“外部世界”的链接。更具体地说，native library是Java方法，其实现由非Java代码（最有可能是C）提供。此功能不是Java专用的。大多数语言都提供了某种机制来调用用另一种语言编写的例程。

参考：

[什么是 jvm.dll，Java.exe和javaw.exe之间的区别？](#)

[什么是dll（native library）](#)

5 总结“为何jvm.dll是jvm的核心？”

我的简单理解是：因为jvm.dll为我们提供了一个接口，这个接口允许我们在执行java程序时去调用外部的Non-java代码文件。在“jvm.dll是jvm的核心？”问题中，因为jvm.dll所提供的接口，才能让我们的java程序和外界的“Non-java”文件进行交互，使BootStrap classloader加载成功。因此，后续的java程序才能正常开始执行。