

JVM、Java编译器和Java解释器

java解释器就是把在java虚拟机上运行的目标代码（字节码）解释成为具体平台的机器码的程序。即jdk或jre目录下bin目录中的java.exe文件，而javac.exe是编译器。

运行java程序的过程是先用javac编译，然后用java解释。而一经编译成功后，就可以直接用java.exe随处解释运行了。

JVM：JVM有自己完善的硬件架构，如处理器、堆栈（Stack）、寄存器等，还具有相应的指令系统（字节码就是一种指令格式）。JVM屏蔽了与具体操作系统平台相关的信息，使得Java程序只需要生成在Java虚拟机上运行的目标代码（字节码），就可以在多种平台上不加修改地运行。JVM是Java平台无关的基础。JVM负责运行字节码：JVM把每一条要执行的字节码交给解释器，翻译成对应的机器码，然后由解释器执行。JVM解释执行字节码文件就是JVM操作Java解释器进行解释执行字节码文件的过程。

Java编译器：将Java源文件（.java文件）编译成字节码文件（.class文件，是特殊的二进制文件，二进制字节码文件），这种字节码就是JVM的“机器语言”。javac.exe可以简单看成是Java编译器。

Java解释器：是JVM的一部分。Java解释器用来解释执行Java编译器编译后的程序。java.exe可以简单看成是Java解释器。

注意：通常情况下，一个平台上的二进制可执行文件不能在其他平台上工作，因为此可执行文件包含了对目标处理器的机器语言。而Class文件这种特殊的二进制文件，是可以运行在任何支持Java虚拟机的硬件平台和操作系统上的！

维基百科定义：

JVM：一种能够运行Java字节码（Java bytecode）的虚拟机。

字节码：字节码是已经经过编译，但与特定机器码无关，需要解释器转译后才能成为机器码的中间代码。

Java字节码：是Java虚拟机执行的一种指令格式。

解释器：是一种电脑程序，能够把高级编程语言一行一行直接翻译运行。解释器不会一次把整个程序翻译出来，只像一位“中间人”，每次运行程序时都要先转成另一种语言再作运行，因此解释器的程序运行速度比较缓慢。它每翻译一程序叙述就立刻运行，然后再翻译下一行，再运行，如此不停地进行下去。它会先将源码翻译成另一种语言，以供多次运行而无需再经编译。其制成品无需依赖编译器而运行，程序运行速度比较快。

即时编译(Just-in-time compilation: JIT)：又叫实时编译、及时编译。是指一种在运行时期把字节码编译成原生机器码的技术，一句一句翻译源代码，但是会将翻译过的代码缓存起来以降低性能损耗。这项技术是被用来改善虚拟机的性能的。

JIT编译器是JRE的一部分。原本的Java程序都是要经过解释执行的，其执行速度肯定比可执行的可执行二进制字节码程序慢。为了提高执行速度，引入了JIT。在运行时，JIT会把翻译过来的机器码保存起来，以备下次使用。而如果JIT对每条字节码都进行编译，则会负担过重，所以，JIT只会对经常执行的字节码进行编译，如循环，高频度使用的方法等。它会以整个方法为单位，一次性将整个方法的字节码编译为本地机器码，然后直接运行编译后的机器码。