**Java**已经成为一个庞大而复杂的技术平台，对于开发人员而言，要想更好的掌握Java技术，深入理解底层的技术处理细节必不可少。现在介绍下java的四个核心概念：

　　1.Java虚拟机

　　Java虚拟机的主要任务是装载class文件并且执行其中的字节码。Java虚拟机包含一个类装载器，它可以从程序和 API中装载class文件。**Java**API中只有程序执行时需要的那些类才会被装载。字节码由执行引擎来执行。不同的Java虚拟机中，执行引擎可能实现得非常不同。在由软件实现的虚拟机中，最简单的执行引擎就是一次性解释字节码。

　　另一种执行引擎更快，但是也更消耗内存，叫做"即时编译器(just-in-time compiler)"。在这种情况下，第一次被执行的字节码会被编译成本地机器代码。编译出的本地机器代码会被缓存，当方法以后被调用的时候可以重用。

　　第三种执行引擎是自适应优化器。在这种方法里，虚拟机开始的时候解释字节码，但是会监视运行中程序的活动，并且记录下使用最频繁的代码段。程序运行的时候，虚拟机只把那些活动最频繁的代码编译成本地代码，其他的代码由于使用得不是很频繁，继续保留为字节码-由虚拟机继续解释它们。

　　一个自适应的优化器可以使得Java虚拟机在80%~90%的时间里执行被优化过的本地代码，而只需要编译10%~20%的对性能有影响的代码。

　　2.类装载器的体系结构

　　一个Java应用程序可以使用两种类装载器："启动(bootstrap)"类装载器和用户定义的类装载器。启动类装载器(这是系统中唯一的)是Java虚拟机实现的一部分。启动类装载器通常使用某种默认方式从本地磁盘中装载类，包括**java**API类(启动类装载器也被称为原始类装载器、系统类装载器或者默认类装载器)。

　　Java应用程序能够在运行时安装用户定义的类装载器，这种类装载器能够使用自定义的方式来装载类。例如，从网络下载class文件。尽管启动类装载器是虚拟机实现的本质部分，而用户定义的类装载器不是，但用户定义的类装载器能够用Java来编写,能够被编译成class文件，能够被虚拟机装载，还能够像其它对象一样实例化。

　　3.Java class文件

　　Java class文件主要在平台无关性和网络移动性方面使Java更适合网络。它在平台无关性方面的任务是：为Java程序提供独立于底层主机平台的二进制形式的服务。这种途径途径打破了C或者C++等语言所遵循的传统，使用这些传统语言写的程序通常首先被编译，然后被连接成单独的、专门支持特定硬件平台和**操作系统**的二进制文件。

　　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。而Java class文件时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二进制文件。

　　4.Java API

　　Java API通过支持平台无关性和安全性，使得Java适应于网络应用。Java API是运行库的集合，它提供了一套访问主机系统资源的标准方法。运行Java程序时，虚拟机装载程序的class文件所使用的Java API class文件。所有被装载的class文件(包括从应用程序中和从Java API中提取的)和所有已经装载的动态库(包含本地方法)共同组成了在Java虚拟机上运行的整个程序。