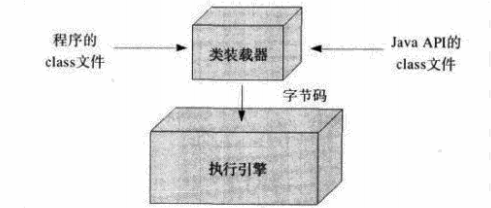
Java虚拟机的主要任务是装载class文件并且执行其中的字节码。

Java虚拟机的基本结构图：



Java虚拟机包含一个类装载器（class loader），可以从程序和API中装载class文件，其中API只有程序执行时需要的那些类才会加载，字节码由执行引擎执行。

执行引擎的种类：

1. 最简单的执行引擎，一次性解释字节码
2. 即时编译器（just-in-time compiler）:第一次被执行的字节码会被编译成本地机器代码缓存下来，以后调用时可以重用。它的速度快，但更耗内存。
3. 自适应优化器：虚拟机开始的时候解释字节码，监听运行中程序的活动，并记录下使用最频繁的代码段。程序运行时，虚拟机只把那些活动最频繁的代码编译成本地机器码。其他的代码继续保留为字节码----由虚拟机继续解释他们。
4. 执行引擎内嵌在芯片里面的，虚拟机由硬件芯片构成，他用本地方法执行Java字节码。

Java中有两种方法：Java方法和本地方法。Java方法是由Java语言编写，编译成字节码，存储在class文件中。本地方法由其它语言（c、c++或者汇编）编写，编译成和处理器相关的机器代码。本地方法保存在动态链接库中，格式是各个平台专有的。而Java方法与平台无关。因此如果使用本地方法，那么程序就变得和平台相关了。

## 类装载其体系结构

启动（bootstrap）类装载器和用户定义的类装载器，bootstrap是Java虚拟机实现的一部分。