**Java 程序执行过程**

1. Java文件编译成 字节码(. class) 文件 开发工具在保存代码时会自动编译
2. JRE加载器从硬盘读取 class 文件. 载入到 运行数据区(Runtime Data Areas)
3. 执行引擎解释或者编译类文件,转化成特定CPU的机器码
4. CPU执行机器码 完成整个过程

**什么是 JRE 的 类加载器**

层级结构:

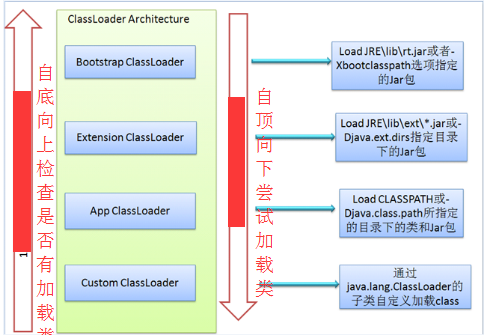
Bootstrap Class Loader -- 运行虚拟机时创建,加载基本API和Object类(C/C++编写)

--> Extension Class Loader -- 加载拓展类 安全性能相关的类

--> System Class Loader -- 应用程序中的类, classPath中配置的类

-->User-Defined Class Loader -- 通过拓展 Class Loader 类定义的自定义加载器, 自定义的类

委派模式



加载器加载了类之后,继续的流程:

Loading->Verifying->Preparing->Resolving->Initialing

Loading: 类加载, 将class 文件加载到JVM 内存(运行数据区域)

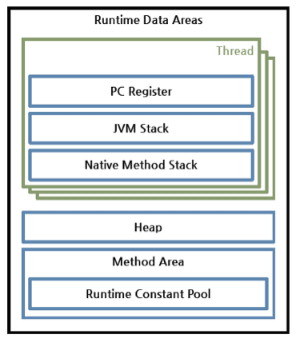
Verifying: 检查载入的类文件是否符合java规范和虚拟机规范

Preparing: 分配内存给该类,确定这个类的属性,方法等所需的数据结构

Resolving: 将给累常量池中的符号引用变为直接引用.

Initialing: 初始化类的局部变量,为静态域赋值,同时执行静态初始化块

**运行数据区域**:



运行数据区域可以划分为6大块：

Java栈、

程序计数寄存器（PC寄存器）、

本地方法栈（Native Method Stack）、

Java堆、

方法区域、

运行常量池（Runtime Constant Pool）