

## 类加载机制



本质上是把类搬进和搬出的过程

## 加载

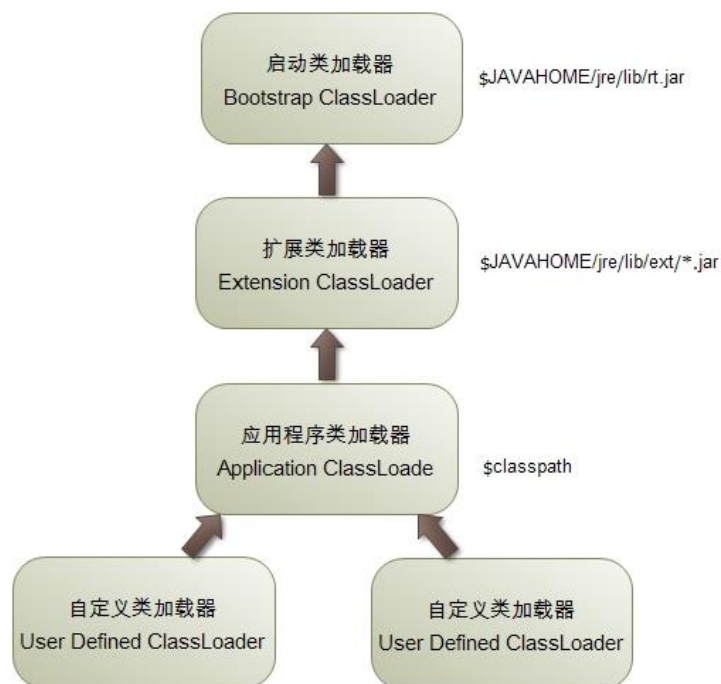
时机：

1. 创建一个对象、使用或设置类的静态变量和调用类的静态方法
2. 初始化一个类时，如果父类还未初始化，父类先初始化。
3. 当虚拟机启动时，包含 main 方法的类先初始化

内容：

1. 通过类的全限定名来获取类的字节流
2. 将类的字节流的静态结构转换为方法区的存储结构
3. 在内存中生成类对象，作为访问该类方法区数据的入口

类加载器：



双亲委派模型：

如果一个类加载器收到一个类加载任务，它先不会加载，而是把它丢给它的父类先加载。如果父类加载失败，它再自己加载

优点：

1. 类与类的加载顺序实际上变成了一个优先级队列，最基础的类会被首先加载（Object），而不会被加载几次

缺点：

1. 基础类很可能需要加载下一级的类，这样就变成了上一层加载器把加载任务委派给下一层。

## 验证：

工作：

文件格式验证

字节流是否符合 Class 文件规范

元数据验证

元数据是否符合 JAVA 语言规范

字节码验证

检查类的方法体

符号引用验证

符合引用能否找到类等

## 准备：

对类变量进行初始化，`static int a = 5`,a 实际上在这个阶段是等于 0

## 解析：

从符合引用到直接引用的装换

类或接口的解析

字段解析

接口方法解析

## 初始化：

执行类变量赋值操作和静态语句块