



链滴

java虚拟机-01-类的加载

作者: [tianma630](#)

原文链接: <https://ld246.com/article/1375368527530>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

一、java虚拟机结束生命周期的条件
1、程序正常结束
2、执行system.exit()
3、程序出现异常或错误而非正常结束
4、操作系统出现错误

二、类的加载、连接与初始化
1、加载：查找并加载类的二进制数据
a、定义：类的加载指的是将类的.class文件中的二进制数据读入到内存中，将其放在运行时数据区的方法区内，然后在堆区创建一个java.lang.Class对象，用来封装类在方法区内的数据结构。
b、方式：
I、从本地系统中直接加载
II、通过网络加载.class文件
III、从zip、jar等归档文件中加载.class文件
IV、从专有的数据库中提取.class文件
V：将java源文件动态编译成.class文件
2、连接
a、验证：确保被加载的类的准确性
I、类文件的结果检查
II、语义查
III、字节码检查
IV、二进制兼容的验证
b、准备：为类的静态变量分配内存，将其初始化为默认值
c、解析：吧类中符号引用转换为直接引用
3、初始化：为类的静态变量赋予正确的初始值

三、类的使用方式
1、主动使用：所有的java虚拟机实现必须在每类或接口被java程序“首次主动使用”是才初始化他们
a、创建类的实例
b、访问某个类或接口的静态变量，或者对静态变量赋值
c、调用类的静态方法
d、反射
e、初始化一个类的子类
f、java虚拟机启动时被标明为启动类的类
2、被动使用

四、类加载器的类型
1、java虚拟机自带的加载器
a、跟类加载器 (Bootstrap)
b、扩展类加载器 (Extension)
c、系统类加载器 (System)
2、用户自定义的类加载器
java.lang.ClassLoader的子类

五、类的初始化步骤
1、假如这个类还没有被加载和连接，那就先行加载和连接。
2、假如类存在直接的父类，并且这个父类还没有被初始化，那就先初始化直父父类。
3、假如类中存在初始化语句，那就依次执行这些初始化语句。

六、类的初始化的前提条件
1、初始化一个类时，要求它的所有父都已经被初始化，但是，这条规则并不适用于接口。
2、在初始化一个类时，并不会先初始化所有实现的接口。
3、在初始化一个接口时，并不会先初始化它的父接口。
4、只有当序首次使用特定接口的静态变量时，才会导致该接口的初始化。
5、调用ClassLoader的loadClass方法加载一个类，并不是对类的主动使用，不会导致类的初始化。

七、类加载的父委托机制
1、定义：在此委托机制中，除了java虚拟机在带的根加载器以外，其余的类加载器都有且只有一个父加载器。当java程序请求加载器加载一个类时，该加载器会首先委托父加载器去加载该类，如果父加载器不能加载该类，才会由该加载器进行加载
2、加载器顺序：用户自定义加载器 > 系统类加载器 > 扩展类加载器 > 根类加载器

八、命名空间
每个类加载器都有自己的命名空间，命名空间由该加载器及所有父加载器所加载的类组成。在同一个命名空间中，不会出现类的完整名字相同的两个类。

九、运行时包
由同一类加载器加载的属于相同包的类组成了运行时包。决定两个类是不是属于同一个运行时包，不仅要看他们的包名是否相同，还要看他们的类加载器是否相同。只有属于同一运行时包的类才能相互访问。

十、类的卸载
当代表一个类的Class对象不再被引用，即不可触及时Class对象就会结束生命周期，该类在方法区内的数据也会被卸载，从而结束该类的生命周期。由此可见一个类的生命周期取决于代表该类的Class对象何时结束生命周期。有java虚拟机自带的类加载器所加载的类，在虚拟机的生命周期内，始终不会被卸载。