Java类加载器解析：

运行java代码就是开启一个jvm虚拟机的进程，程序运行结束，则进程终止

注：将java字节码文件通过类加载器加载到内存中，程序才能运行

类加载器加载类的过程中完成的操作有：

1.加载：查找并加载类的二进制数据

2.连接：

验证：确保加载的类的正确性

准备：为类的静态成员变量分配内存，并将其初始化为默认值（静态成员变量属于类而不属于某个对象，所以在类被加载的时候初始化，这个时候还不存在对象）

解析：把类中的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

加载类T的时候，在准备阶段，为类的静态成员变量a赋予默认值0，在解析阶段，赋予正确的值（用户指定的值）3；

**class** T{

**private** **static** **int** *a* = 3;

}

java程序对类的使用方式有两种：

1.主动使用

2.被动使用

所有的java虚拟机实现必须在每个类或接口被java程序“首次主动使用”时才初始化他们

属于主动使用的六种情况：

1.创建类的实例，如：new ClassA();

2.访问某个类或接口的静态变量，或者对该静态变量赋值 如ClassA.a或ClassA.a = 4;

3.调用类的静态方法 ClassA.method();

4.反射

Class.forName(“com.shengsiyuan.ClassA”);

5.初始化一个类的子类

6.java虚拟机启动时被标明为启动类的类（即main方法所在的类）

如一个Test.java文件包含三个类：Test、ClassA、ClassB

Test类中包含了main方法，则Test类属于主动使用

**除了以上六种情况外，其他使用java类的方式都被看作是对类的被动使用，都不会导致类的初始化**

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