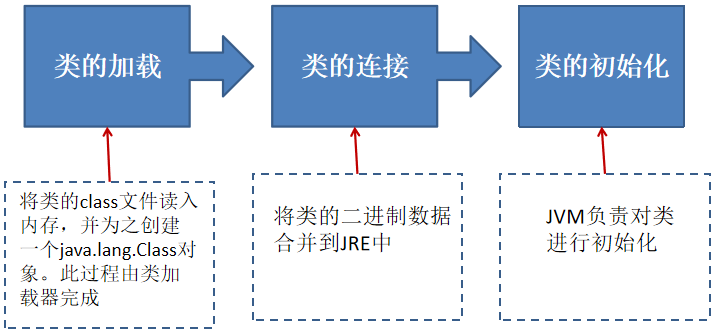
* **类的加载过程**

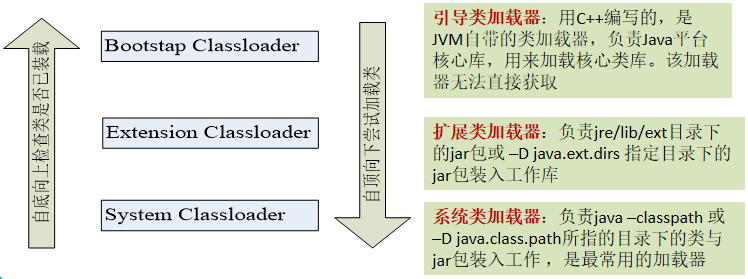
当程序主动使用某个类时，如果该类还未被加载到内存中，则系统会通过如下三个步骤来对该类进行初始化。





* **ClassLoader**

类加载器是用来**把类(class)装载进内存**的。JVM规范定义了两种类型的类加载器：启动类加载器(bootstrap)和用户自定义加载器(user-defined class loader)。JVM在运行时会产生3个类加载器组成的初始化加载器层次结构，如下图所示：



//1.获取一个系统类加载器

ClassLoader classloader = ClassLoader.getSystemClassLoader();

System.out.println(classloader);

//2.获取系统类加载器的父类加载器，即扩展类加载器

classloader = classloader.getParent();

System.out.println(classloader);

//3.获取扩展类加载器的父类加载器，即引导类加载器

classloader = classloader.getParent();

System.out.println(classloader);

//4.测试当前类由哪个类加载器进行加载

classloader= Class.forName("exer2.ClassloaderDemo").getClassLoader();

System.out.println(classloader);

//5.测试JDK提供的Object类由哪个类加载器加载

classloader = Class.forName("java.lang.Object").getClassLoader();

System.out.println(classloader);

//\*6.关于类加载器的一个主要方法：getResourceAsStream(String str):获取类路径下的指定文件的输入流

InputStream in = null;

in=this.getClass().getClassLoader().getResourceAsStream("exer2\\test.properties");

System.out.println(in);