编写的java文件最终被编译为class文件，在运行期间将class文件加载到方法区，对文件流进行校验解析初始化，最终成为可以使用的java对象。Class文件加载到内存到从内存卸载经历了这些阶段：加载、连接、初始化、使用、卸载。其中连接有分为验证、准备、解析三个阶段，如下图。



我们重点研究下加载、连接、初始化这三个过程中发生了什么。

一、使用类加载器加载class文件

先让我们自定义一个类加载器，并初始化这个class对象，来初步体验下类加载到使用的过程，首先实现一个简单的class类

public class MgDemoSample {  
  
 public void startSample()  
 {  
 System.*out*.println("MgDemoSample");  
 }  
}

使用javac MgDemoSample.java 编译出 class文件，将class文件放到E:/cltmp/目录下，然后创建一个URLClassLoader加载MgDemoSample，并实例化对象，通过反射执行startSample这个函数

URL url = new URL("file:/E:/cltmp/");  
URLClassLoader loader = new URLClassLoader(new URL[]{url});  
Class cl = loader.loadClass("MgDemoSample");  
*log*.info("....{}",cl.getName());  
Object instance = cl.newInstance();  
Method mainMethod = cl.getMethod("startSample");

这样我们就把类的，加载、链接、初始化、使用，走了一遍。

接下来让我们一起研究下，加载、连接、初始化，这三个过程。

二、加载

在加载过程中主要做了什么呢

1、获取类的二进制字节流

2、将字节流转换为静态存储结构存储到方法区

3、生成class对象

注意：class的连接这个过程并不是在加载结束后才会开始，这两个阶段是并行的，例如需要验证阶段对文件格式验证通过以后，字节流才能存储到方法区中。

可以通过url或者直接加载二进制流

二、连接

1、验证

文件格式验证、元数据验证、字节码验证、符号引用验证

2、准备

3、解析

四、初始化