

```
MyTest23.java ×
4  在运行期，一个Java类是由该类的完全限定名（binary name，二进制名）和用于加载该类的定义类加载器（defining loader）所共同决定的。
5  如果同样名字（即相同的完全限定名）的类是由两个不同的加载器所加载，那么这些类就是不同的，即便.class文件的字节码完全一样，并且从相同的位置加载亦如此。
6  */
7
8
9  /*
10  在Oracle的Hotspot实现中，系统属性sun.boot.class.path如果修改错了，则运行会出错，提示如下错误信息：
11
12  Error occurred during initialization of VM
13  java/lang/NoClassDefFoundError: java/lang/Object
14  */
15
16  public class MyTest23 {
17
18      public static void main(String[] args) {
19          System.out.println(System.getProperty("sun.boot.class.path"));
20          System.out.println(System.getProperty("java.ext.dirs"));
21          System.out.println(System.getProperty("java.class.path"));
22
23          /*
24           内建于JVM中的启动类加载器会加载java.lang.ClassLoader以及其他的Java平台类，
25           当JVM启动时，一块特殊的机器码会运行，它会加载扩展类加载器与系统类加载器，
26           这块特殊的机器码叫做启动类加载器（Bootstrap）。
27
28           启动类加载器并不是Java类，而其他的加载器则都是Java类，
29           启动类加载器是特定于平台的机器指令，它负责开启整个加载过程。
30
31           所有类加载器（除了启动类加载器）都被实现为Java类。不过，总归要有一个组件来加载第一个Java类加载器，从而让整个加载过程能够顺利
32           进行下去，加载第一个纯Java类加载器就是启动类加载器的职责。
33
34           启动类加载器还会负责加载供JRE正常运行所需要的基本组件，这包括java.util与java.lang包中的类等等。
35          */
36
37          System.out.println(ClassLoader.class.getClassLoader());
38      }
39  }
40
41
```