

计算机二级辅导:java类装载(3)计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_645449.htm

奇怪的隔离性 我们不难发现，图2中的类装载机AA和AB，AB和BB，AA和B等等位于不同分支下，他们之间没有父子关系，我不知道如何定义这种关系，姑且称他们位于不同分支下。两个位于不同分支的类装载机具有隔离性，这种隔离性使得在分别使用它们装载同一个类，也会在内存中出现两个Class类的实例。因为被具有隔离性的类装载机装载的类不会共享内存空间，使得使用一个类装载机不可能完成的任务变得可以轻而易举，例如类的静态变量可能同时拥有多个值（虽然好像作用不大），因为就算是被装载类的同一静态变量，它们也将被保存不同的内存空间，又例如程序需要使用某些包，但又不希望被程序另外一些包所使用，很简单，编写自定义的类装载机。类装载器的这种隔离性在许多大型的软件应用和服务程序得到了很好的应用。下面是同一个类静态变量为不同值的例子。

```
package test. public class A { public static void main( String[] args ) { try { //定义两个类装载机 MyClassLoader aa= new MyClassLoader(). MyClassLoader bb = new MyClassLoader(). //用类装载机aa装载testb.B类 Class clazz=aa.loadClass("testb. B"). Constructor constructor= clazz.getConstructor(new Class[] {Integer.class}). Object object = constructor.newInstance(new Object[] {new Integer(1)}). Method method = clazz.getDeclaredMethod("printB",new Class[0]). //用类装载机bb装载testb.B类 Class clazz2=bb.loadClass("testb. B").
```

```
Constructor constructor2 = clazz2.getConstructor(new
Class[] {Integer.class}). Object object2 =
constructor2.newInstance(new Object[] {new Integer(2)}). Method
method2 = clazz2.getDeclaredMethod("printB", new Class[0]). //显
示test.B中的静态变量的值 method.invoke( object, new
Object[0]). method2.invoke( object2, new Object[0]). } catch (
Exception e ) { e.printStackTrace(). } } } //Class B 必须位
于MyClassLoader的查找范围内, //而不应该在MyClassLoader
的父类装载器的查找范围内。 package testb. public class B {
static int b . public B(Integer testb) { b = testb.intValue(). } public
void printB() { System.out.print("my static field b is ", b). } } 100Test
下载频道开通, 各类考试题目直接下载。 详细请访问
www.100test.com
```