

Java中的Class Loader与Package机制 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/353/2021_2022_Java_E4_B8_AD_E7_9A_84_c104_353757.htm 为了深入了解Java

的ClassLoader机制，我们先来做以下实验：package

```
java.lang.public class Test { public static void main(String[] args) {  
char[] c = "1234567890".toCharArray(). String s = new String(0, 10,  
c.)}}
```

String类有一个Package权限的构造函数String (int offset , int length , char[] array) ，按照默认的访问权限，由于Test属于java.lang包，因此理论上应该可以访问String的这个构造函数。

编译通过！执行时结果如下：Exception in thread "main"

```
java.lang.SecurityException:Prohibited package name:java.lang at
```

```
java.lang.ClassLoader.defineClass(Unknown Source) at
```

```
java.security.SecureClassLoader.defineClass(Unknown Source) at
```

```
java.net.URLClassLoader.defineClass(Unknown Source) at
```

```
java.net.URLClassLoader.access$100(Unknown Source) at
```

```
java.net.URLClassLoader$1.run(Unknown Source) at
```

```
java.security.AccessController.doPrivileged(Native Method) at
```

```
java.net.URLClassLoader.findClass(Unknown Source) at
```

```
java.lang.ClassLoader.loadClass(Unknown Source) at
```

```
sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
```

```
at java.lang.ClassLoader.loadClass(Unknown Source) at
```

```
java.lang.ClassLoader.loadClassInternal(UnknownSource) 奇怪吧
```

？要弄清为什么会有SecurityException，就必须搞清楚

ClassLoader的机制。Java的ClassLoader就是用来动态装

载class的，ClassLoader对一个class只会装载一次，JVM使用

的ClassLoader一共有4种：启动类装载器，标准扩展类装载器，类路径装载器和网络类装载器。这4种ClassLoader的优先级依次从高到低，使用所谓的“双亲委派模型”。确切地说，如果一个网络类装载器被请求装载一个java.lang.Integer，它会首先把请求发送给上一级的类路径装载器，如果返回已装载，则网络类装载器将不会装载这个java.lang.Integer，如果上一级的类路径装载器返回未装载，它才会装载java.lang.Integer。类似的，类路径装载器收到请求后（无论是直接请求装载还是下一级的ClassLoader上传的请求），它也会先把请求发送到上一级的标准扩展类装载器，这样一层一层上传，于是启动类装载器优先级最高，如果它按照自己的方式找到了java.lang.Integer，则下面的ClassLoader都不能再装载java.lang.Integer，尽管你自己写了一个java.lang.Integer，试图取代核心库的java.lang.Integer是不可能的，因为自己写的这个类根本无法被下层的ClassLoader装载。再说说Package权限。Java语言规定，在同一个包中的class，如果没有修饰符，默认为Package权限，包内的class都可以访问。但是这还不够准确。确切的说，只有由同一个ClassLoader装载的class才具有以上的Package权限。比如启动类装载器装载了java.lang.String，类路径装载器装载了我们自己写的java.lang.Test，它们不能互相访问对方具有Package权限的方法。这样就阻止了恶意代码访问核心类的Package权限方法。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