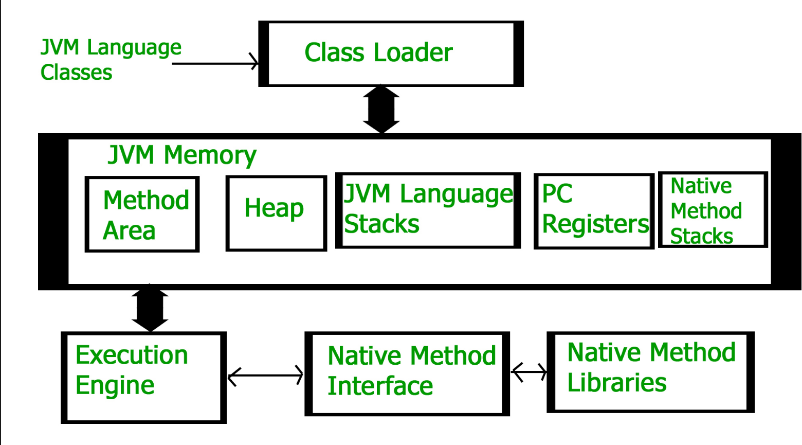
**Class loaders**

Class file : C:/target folder

Run my code: jvm (java virtual machine)



When u run ur code, jvm will first load ur class files into the method area.

Various classloaders:

1. Bootstrap: rt.jar (C:\Program Files\Java\jdk1.8.0\_191\jre\lib\rt.jar)
2. Extension: (C:\Program Files\Java\jdk1.8.0\_191\jre\lib\ext)
3. Application/System: Any library in ur classpath

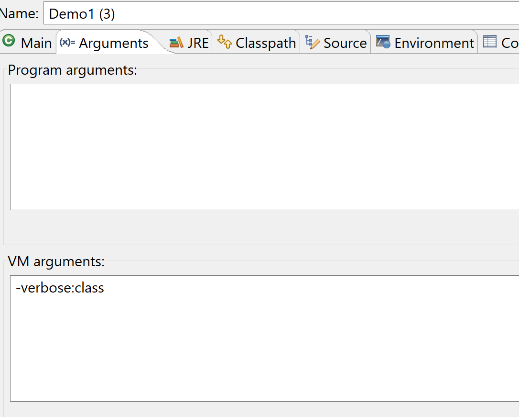
To create ur own classloader u can extend java.lang.ClassLoader class

static class sun.misc.Launcher$ExtClassLoader: Written in java

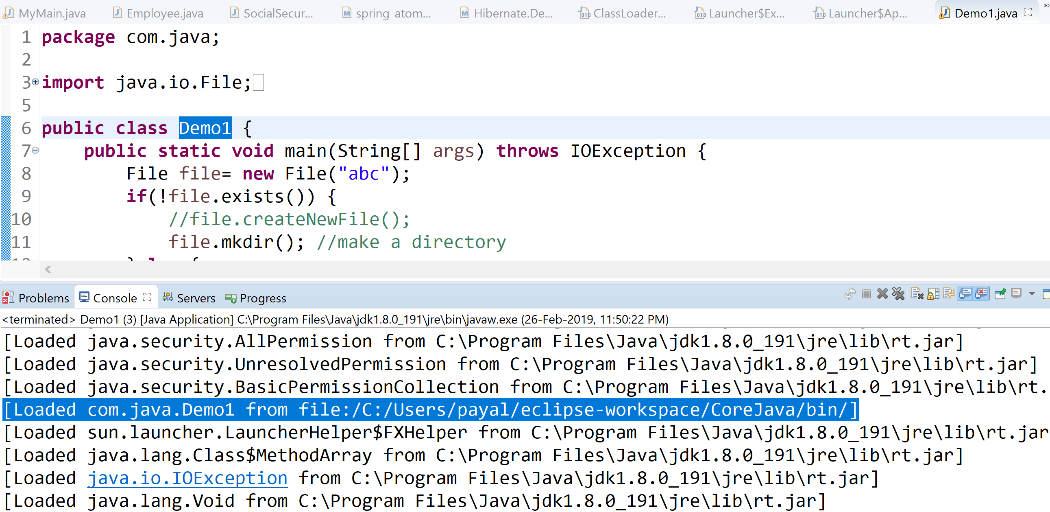
static class sun.misc.Launcher$AppClassLoader

Bootstrap is written in native language. Not in java language

To see classes which got loaded into the jvm:

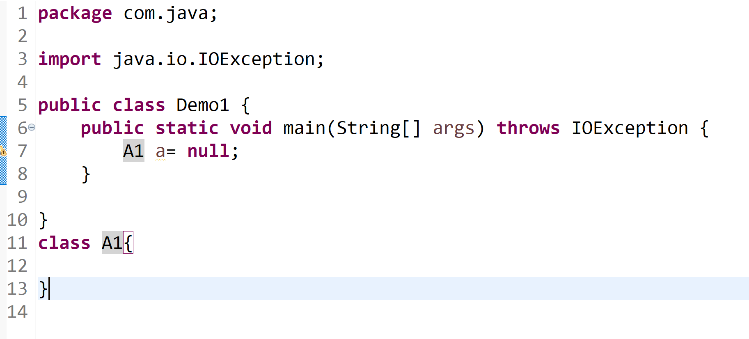


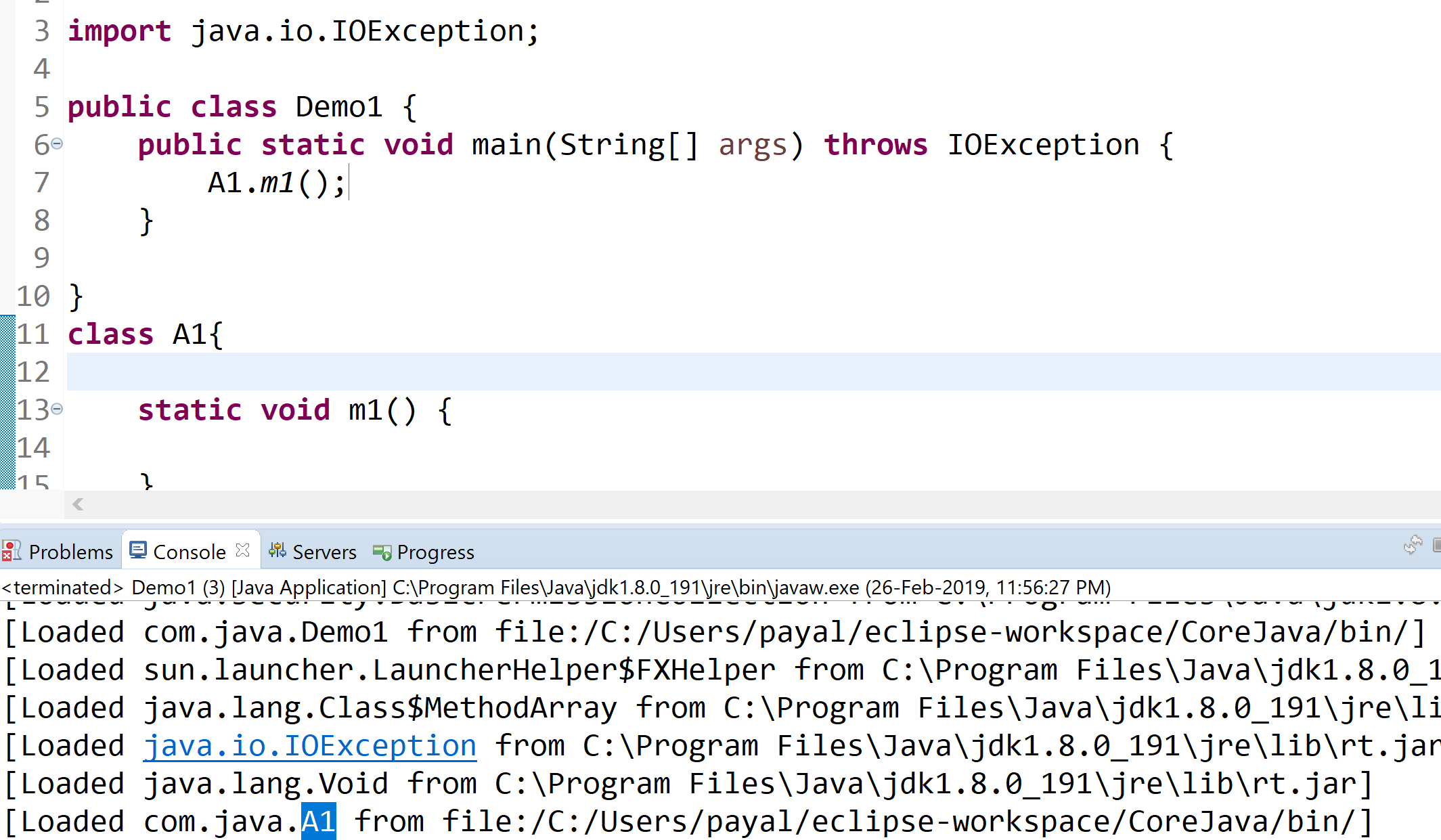
So by-default first it loads all classes from rt.jar into ur jvm memory area.



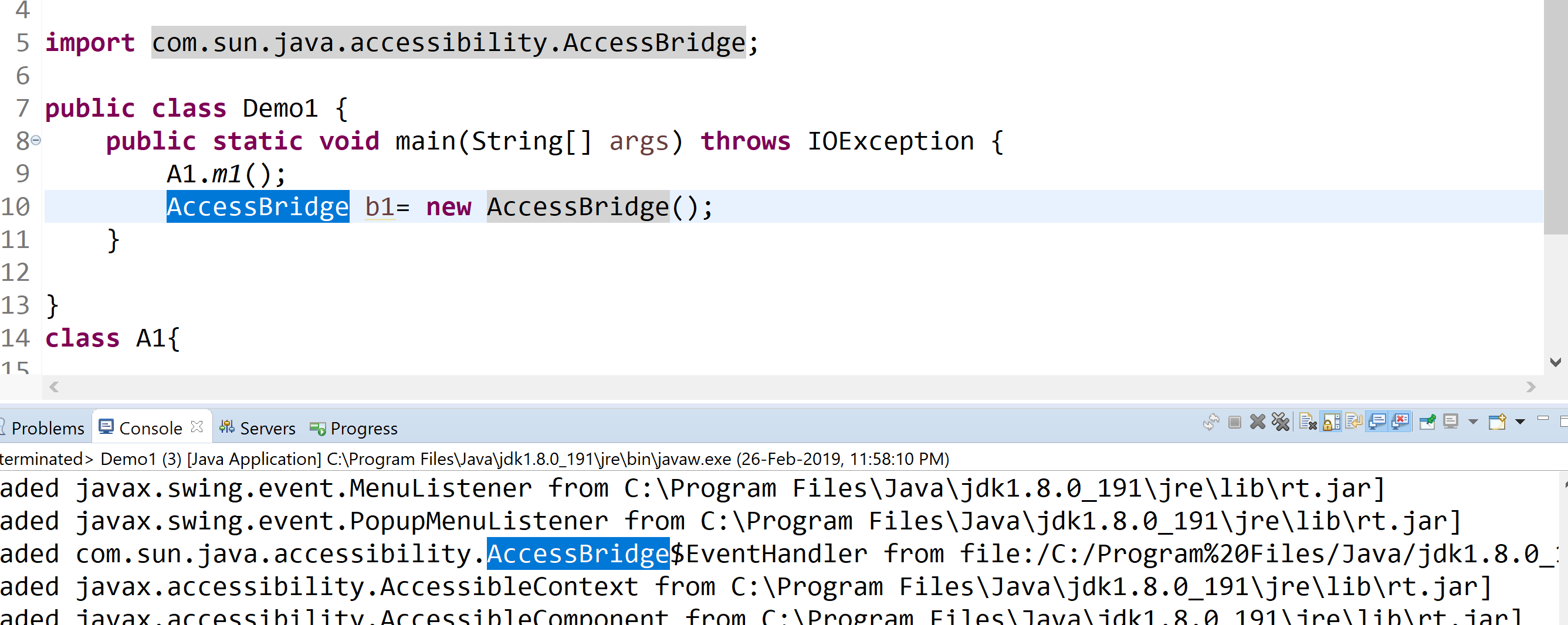
It does not load classes frm ext folder on it’s own.

Class A1 will not be loaded into the jvm:





Demo1 and A1 will be loaded this time.



Even classes from ext would be loaded if we use them in our project

Loading follows a **delegating pattern**

When a class has to be loaded, first bootstrap will try to load it. It not from rt.jar, the responsibility for loading the class will be delegated to extension class loader. If not from ext folder, then appclassloader will try to load it. (priority)