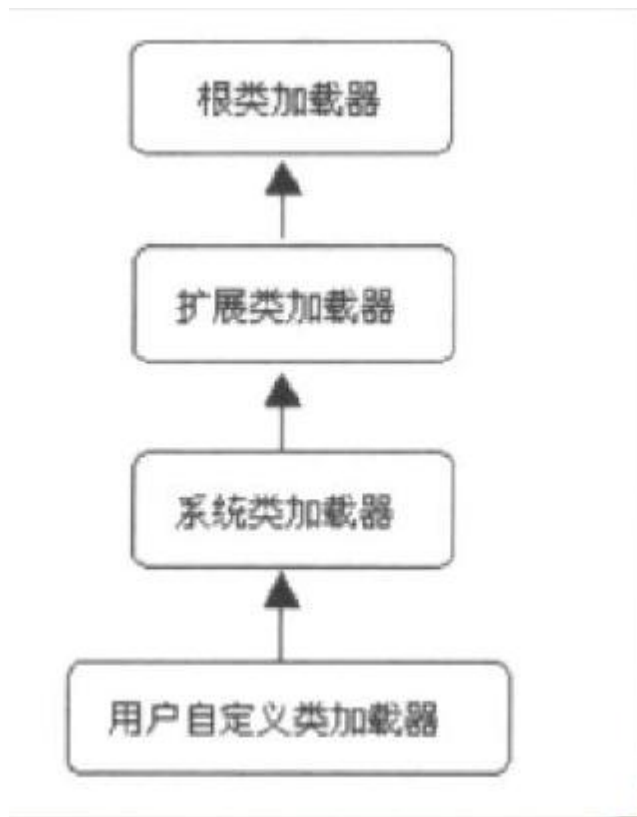


Java SE Lesson 22

1. `ClassLoader`（类加载器）
2. JVM 提供了三种类加载器
 - 1) 根类加载器（使用 C++ 编写，程序员无法在 Java 代码中获得该类）
 - 2) 扩展加载器，使用 Java 代码实现
 - 3) 系统加载器（应用加载器），使用 Java 代码实现
3. 用户自定义的类加载器都是 `java.lang.ClassLoader` 的子类
4. 程序中对子类的“主动使用”会导致父类被初始化；但对父类的“主动”使用并不会导致子类初始化（不可能说生成一个 `Object` 类的对象就导致系统中所有的子类都会被初始化）
5. 类加载器的父亲委托机制（Parent Delegation）
6. 父子加载器并非继承关系，也就是说子加载器不一定是继承了父加载器。
7. JVM 自带的类加载器之间的关系：



8. 定义类加载器：如果某个类加载器能够加载一个类，那么该类加载器就称作：**定义类加载器**；定义类加载器及其所有子加载器都称作：**初始类加载器**。
9. 当生成一个自定义的类加载器实例时，**如果没有指定它的父加载器**，那么系统类加载器就将成为该类加载器的父加载器。

Creates a new class loader using the `ClassLoader` returned by the method `getSystemClassLoader()` as the parent class loader.

10.

