目标:

1. 热部署原理解析
2. 案例分析
3. 项目演示
4. 测试
5. 发布程序
6. 课程总结
7. **java热部署与热加载的联系和区别**

Java热部署与热加载联系

不重启服务器编译/部署项目

基于Java的类加载器实现

部署方式区别

热部署在服务器运行时重新部署项目

热加载在运行时重新加载class

实现原理区别

热部署直接重新加载整个应用

热加载在运行时重新加载class

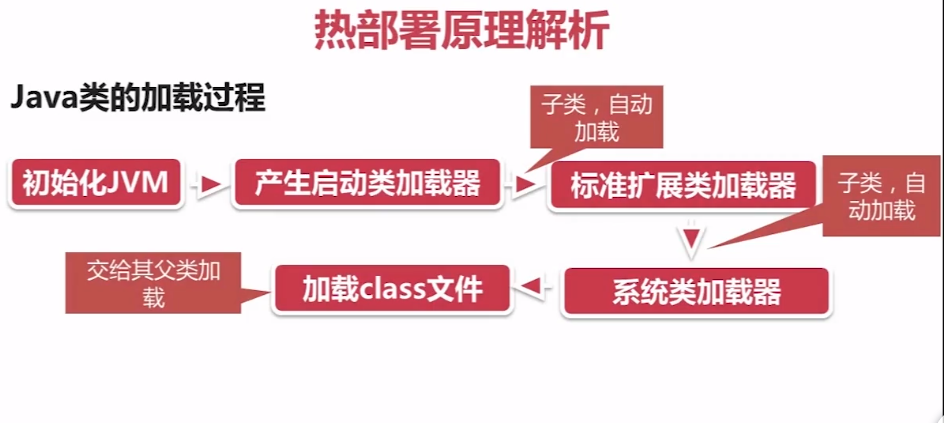
使用场景区别

热部署更多的是在生产环境使用

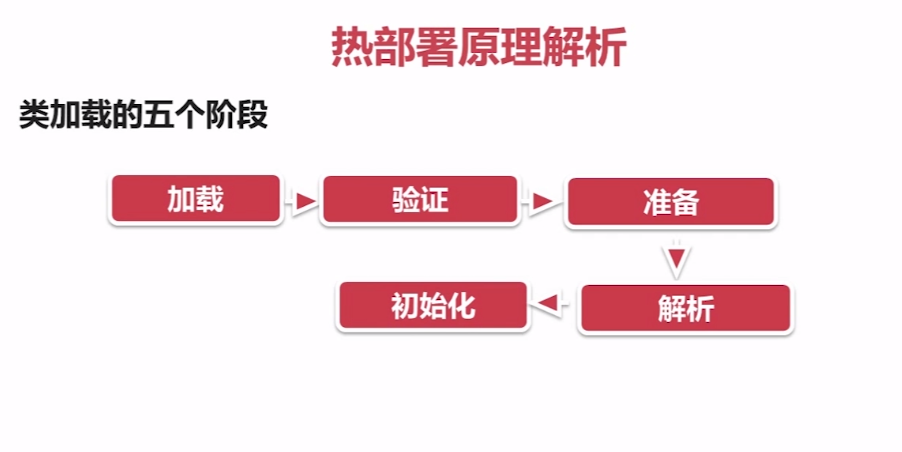
热加载则更多的是在开发环境使用

1. **热部署的原理解析**

**2.1、java类的加载过程**



**2.2、类加载的五个阶段**



**2.3、Java类加载器特点**

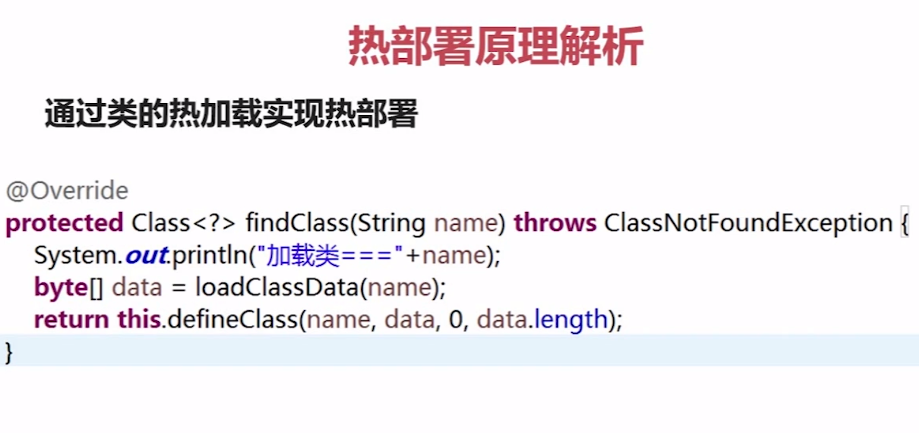
1 由AppClassLoader(系统类加载器)开始加载指定类

2 类加载器将加载任务交给其父，如果其父找不到，再交给自己去加载

3 BootstrapLoader(启动类加载器)是最顶级的类加载器

2.4、Java类的热部署

1 类的热加载



2 配置tomcat

1 直接把项目web文件夹放在webapps里

2 在tomcat\conf\server.xml中的<host></host>内部添加<context/>标签

3 在%tomcat\_home%\conf\Catalina\localhost中添加一个XML

1. **Java类热加载案例分析**

写一个Java类热加载的实际案例

要求:

1. 类层次结构清晰、修改某一个Java类文件

不需要重启服务或者重新编译运行程序

2、可适当的运用一些设计模式使代码结构更加清晰明了，比如工厂模式等