所谓热部署，就是在应用正在运行的时候升级软件，却不需要重新启动应用。

对于[Java](http://baike.baidu.com/subview/29/12654100.htm" \t "http://baike.baidu.com/view/_blank)应用程序来说，热部署就是在运行时更新Java类文件

《深入探索Java热部署》

热部署是在不重启 Java 虚拟机的前提下，能自动侦测到 class 文件的变化，更新运行时 class 的行为。

友好的方法是创建自己的 classloader 来加载需要监听的 class，这样就能控制类加载的时机，从而实现热部署

自定义加载器的实现

自定义加载器仍然需要执行类加载的功能。这里却存在一个问题，同一个类加载器无法同时加载两个相同名称的类，由于不论类的结构如何发生变化，生成的类名不会变，而 classloader 只能在虚拟机停止前销毁已经加载的类，这样 classloader 就无法加载更新后的类了。

利用 ASM 修改 class 文件

改变创建对象的行为

Java 虚拟机常见的创建对象的方法有两种，一种是静态创建，直接 new 一个对象，一种是动态创建，通过反射的方法，创建对象。

Java EE，Java平台企业版（Java Platform Enterprise Edition）

它更多的表达着一种[软件架构](https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E6%9E%B6%E6%9E%84" \o "软件架构)和设计思想。

类加载器

基本概念：类加载器（class loader）用来加载 Java 类到 Java 虚拟机中。

java.lang.ClassLoader类的介绍

java.lang.ClassLoader类的基本职责就是根据一个指定的类的名称，找到或者生成其对应的字节代码，然后从这些字节代码中定义出一个 Java 类，即 java.lang.Class类的一个实例。

类加载器的树状组织结构

引导类加载器（bootstrap class loader）：它用来加载 Java 的核心库，是用原生代码来实现的，并不继承自 java.lang.ClassLoader。

扩展类加载器（extensions class loader）：它用来加载 Java 的扩展库。Java 虚拟机的实现会提供一个扩展库目录。该类加载器在此目录里面查找并加载 Java 类。

系统类加载器（system class loader）：它根据 Java 应用的类路径（CLASSPATH）来加载 Java 类。一般来说，Java 应用的类都是由它来完成加载的。可以通过 ClassLoader.getSystemClassLoader()来获取它。

类加载器的代理模式

类加载器在尝试自己去查找某个类的字节代码并定义它时，会先代理给其父类加载器，由父类加载器先去尝试加载这个类，依次类推。

java 虚拟机是如何判定两个 Java 类是相同的。Java 虚拟机不仅要看类的全名是否相同，还要看加载此类的类加载器是否一样。只有两者都相同的情况，才认为两个类是相同的。即便是同样的字节代码，被不同的类加载器加载之后所得到的类，也是不同的。