Atitit class load 流程 说明

热替换

ClassLoader中重要方法： loadClass：ClassLoader.loadClass(…) 是ClassLoader的入口点。当一个类没有指明用什么加载器加载的时候，JVM默认采用AppClassLoader加载器加载没有加载过的class，调用的方法的入口就是loadClass(…)。如果一个class被自定义的ClassLoader加载，那么JVM也会调用这个自定义的ClassLoader.loadClass(…)方法来加载class内部引用的一些别的class文件。重载这个方法，能实现自定义加载class的方式，会抛弃双亲委托机制，但是即使不采用双亲委托机制，比如java.lang包中的相关类还是不能自定义一个同名的类来代替，主要因为JVM解析、验证class的时候，会进行相关判断。

系统默认的AppClassLoader加载器，他们内部会缓存加载过的class，重新加载的话，就直接取缓存。所与对于热加载的话，只能重新创建一个ClassLoader，然后再去加载已经被加载过的class文件。

Class卸载

在Java中class也是可以unload。JVM中class和Meta信息存放在PermGen space区域。如果加载的class文件很多，那么可能导致PermGen space区域空间溢出。引起：java.lang.OutOfMemoryErrorPermGen space. 对于有些Class我们可能只需要使用一次，就不再需要了，也可能我们修改了class文件，我们需要重新加载 newclass，那么oldclass就不再需要了。那么JVM怎么样才能卸载Class呢。

JVM中的Class只有满足以下三个条件，才能被GC回收，也就是该Class被卸载（unload）：

该类所有的实例都已经被GC。

加载该类的ClassLoader实例已经被GC。

该类的java.lang.Class对象没有在任何地方被引用。