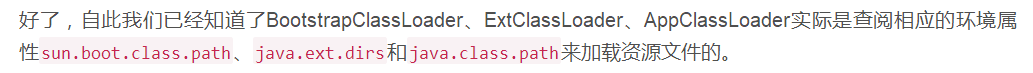
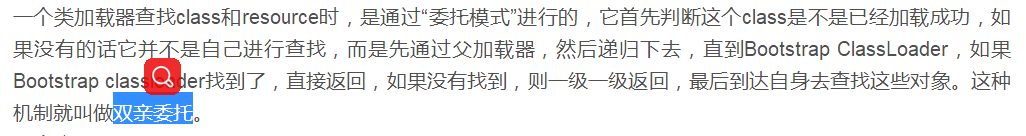
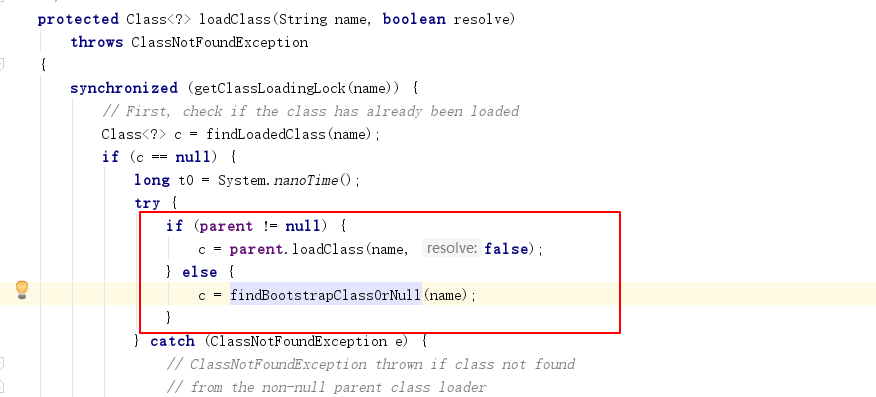
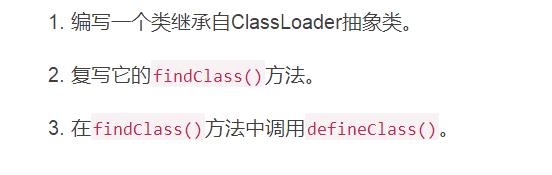
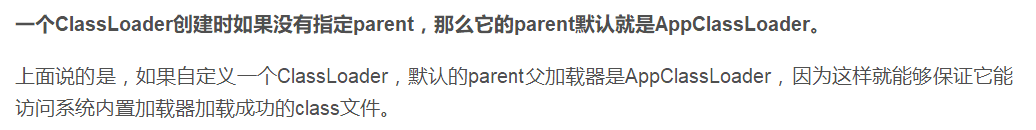
<https://blog.csdn.net/briblue/article/details/54973413>

* Java类加载器：
* 每一个加载器都有一个父加载器，通过*getParent*获得
* Bootstrap ClassLoader是由C/C++编写的，并不是Java的一个类
* JVM启动时通过Bootstrap类加载器加载rt.jar等核心jar包中的class文件，如：int.class,String.class
* 双亲委托：当父classloader为空的时候，会调用Bootstrapclassloader查找类是否已经被加载
* 自定义classloader：
* 解释下图↓：因为自定义加载器会继承Classloader类，Classloader类的默认无参构造方法就是将AppClassloader设置成父加载器
* 子线程的ContextClassLoader默认是AppClassLoader：*Thread.currentThread().getContextClassLoader()*
* 子线程中的AppClassLoader加载不了父线程当中已经加载的class