

实现JAVA的动态类载入机制 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E5_AE_9E_E7_8E_B0JAVA_c104_145414.htm 作为充分利用Java的动态类载入机制的最好例子，带有Java扩展的Web浏览器根据请求从网络或本地文件系统中动态加载Java applet（遵循一定规则的Java小应用程序类），然后在本地系统中执行它，大大增强了主页的功能。---- 其实，Java本身就是一种极具动态性的语言。类似Windows的动态链接库（DLL），Java应用程序总是被编译成若干个单独的class文件，程序执行时根据需要由Java虚拟机动态载入相应的类。这种机制使编写动态的分布式应用程序成为可能：我们可以在客户端编写自己的类载入器，而真正执行的程序却存放在本地、局域网或世界另一端的主机上。下面将介绍如何在自己的应用程序中实现Java的动态类载入机制。与动态类载入有关的系统类---- 为支持动态类载入机制，在系统类组java.lang中提供了两个类：
：Class类和ClassLoader类。---- 1、类java.lang.Class。在Java虚拟机中，每一个类或接口都是由Class类来操纵的，它不能被显式的实例化，必须用其他方法来获取Class类的对象。动态类载入机制的关键一步在于如何获得指定类的Class类型的对象。相关方法主要有：---- public static Class.forName(String className) ---- 这是一个静态方法，它获取指定名字的类的Class类型对象，类名可以是象“sun.applet.Applet”这样的字符串，

但不能带有路径或网络地址等信息。这是从本地系统中动态载入类的最方便的办法。 ---- public Object newInstance() ---- 这是最重要的一个方法，它建立由Class类型对象描述的指定类的实例。 ---- 下面是一个用forName()和newInstance()方法实现动态类载入的代码，share类包含一个接口，详细内容将在第三部分中解释。 try{//根据类名建立Class类型的对象。 Class cc =Class.forName("类名"); //建立被载入类类的实例并强制类型转换，值赋给share类型的变量。 share oo= ((share) cc) .newInstance().//调用该类的方法进行工作。 }catch (Exception ex){//如果发生例外，则进行相应处理。 }。---- 2、类java.lang.ClassLoader。这是一个抽象类，如果打算运用它，必须继承它并重写它的loadClass()方法。其主要方法有： ---- protected ClassLoader(). ---- 这是一个建构元，可以用它建立一个ClassLoader类的实例。注意继承这个类的类必须重写这个方法，而不能使用缺省的建构元。 ---- protected abstract Class loadClass(String name, boolean resolve) ---- 载入指定的类数据，建立Class类型的对象并根据需要解析它。这是一个抽象方法，大家必须在自己的子类中重写这个方法，重写的规则可以参考第三部分的例子。 ---- protected final Class defineClass(byte data[], int offset, int length) ---- 将字节数组中的数据定义为Class类型的对象，字节数组的格式由虚拟机规定。 ---- protected final Class findSystemClass(String name) ---- 根据指定的类名载入类，它会自动在当前目录和环境变量“CLASSPATH”指定的路径中寻找，

如果找不到，则会抛出ClassNotFoundException例外。

---- protected final void resolveClass(Class c) ---- 通过载入与指定的类相关的所有类来解析这个类，这必须在类被使用之前完成。扩充ClassLoader类以实现动态类载入

---- 理解动态类载入机制的最好办法是通过例子，下面这个完整的例子由四个类组成，分别解释如下：

---- 1、MyClassLoader类是ClassLoader类的子类，它重写了loadClass方法，实现了将网络上用URL地址指定的类动态载入，取得它的Class类型对象的功能。读者可根据自己载入类的具体方式改写下面的代码。

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com