

在Java中动态执行类的静态方法 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_9C_A8J
ava_E4_B8_AD_c104_145114.htm](https://www.100test.com/kao_ti2020/145/2021_2022__E5_9C_A8Java_E4_B8_AD_c104_145114.htm) 在Java中，通

过Class.forName()方法可以动态决定加载哪一个类，这个功能对于相同接口的不同实现来讲非常有用。比如对于设计好的数据库连接池接口，我们可以有多种的接口实现类来完成相同的功能，同时用户可以简单的通过修改配置文件来指定实际使用哪一个实现类，在源代码里面通过读取配置文件信息，并用Class.forName(configClassName).newInstance()就可以构造一个特定的实现类实例，而不用每次修改源代码。这样对于程序来讲只用关心接口的定义，用户只用进行配置文件的设置就完成了同一功能的不同实现的切换。但是如果实现类需要通过静态方法来进行初始化的时候，动态加载的过程就会复杂一些。同样的以数据库连接池为例，通常连接池的构造函数都会被定义为私有的，而通过自定义的newInstance()静态方法来得到唯一实例。这种情况下简单的通

过Class.forName().newInstance()就无法正确构造实例。幸好Java所提供的反射机制(Reflection)为我们提供了完整了探悉类内部结构的方法。通过反射机制，我们能够完成基本上所有的运行时决定的动作(虽然这一实现要比其他动态语言，比如PHP，的eval()的使用要复杂的多)。下面通过实际的例子说明如何在运行时动态访问类的静态方法

```
。 -----*/import  
java.lang.reflect.*.public class myTestClass{ private static Object  
pLock = new Object(). private static myTestClass p_instance = null.
```

```
private String s_configName = "". private boolean
b_isFromResource = true. public static Object getInstance(String
sConfigName, Boolean blsFromResource){ synchronized(pLock){
if(null == p_instance){ p_instance =new
myTestClass(sConfigName,blsFromResource). } } return
p_instance. } private myTestClass(String sConfigName,Boolean
blsFromResource){ s_configName = sConfigName.
b_isFromResource = blsFromResource.booleanValue(). } public
void echoInfo(){ System.out.println("current arguments :
configName=[" s_configName "],isFromResource=["
b_isFromResource "]). } public static void main(String[] args)
throws Exception{ // 设置方法的传入参数的类型. Class[]
parameterTypes = new Class[]{ java.lang.String.class,
java.lang.Boolean.class }. Method mGetInstance = null. String
className = "myTestClass". Class curTestClass =
Class.forName(className). try{ mGetInstance =
curTestClass.getMethod("getInstance",parameterTypes). }
catch(NoSuchMethodException e){ e.printStackTrace().
mGetInstance = null. } if(mGetInstance != null){ myTestClass pObj
= (myTestClass) mGetInstance.invoke(null,new Object[]{
"src/myconfig.properties", Boolean.FALSE } ). pObj.echoInfo(). }
else{ throw new Exception("myTest Init Failed from class"
className System.getProperty("line.seperator","\n") "method
getInstance(String, Boolean) exists."). } } 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
```