类加载与反射

SOFTCITS@2016

SoftCITS
SOFT CULTURE IT SALON
软件园IT文化沙龙

类加载

类加载指的是将类的class文件读入内存，并为之创建一个java.lang.class对象。也就是说，当程序中使用任何类时，系统都为之创建一个java.lang.class对象。

所以系统中所有的类实际上也是实列，都是java.lang.class实列。

当JVM启动时，会形成由三个类加载器组成的初始类加载器层次结构：

Bootstrap ClassLoader：根类加载器，加载核心类。
Extension ClassLoader：扩展类加载器。加载%JAVA_HOME%/jre/lib/ext
System ClassLoader：系统类加载器。加载%CLASSPATH%

JVM中除根类加载器，所有类加载器都是 ClassLoader子类的实列。

ClassLoader的子类URLClassLoader是扩展类加载器和系统类加载器的父类。

执行的时候在类加载器中寻找类的结构的顺序是：根类加载器-》 扩展类加载器-》系统类加载器-》我们自定义的一些类加载器

示例代码-1.1

反射

通过反射获取Class对象

Java程序中获得Class对象通常有如下三种方式：

1. 使用Class类的forName()静态方法。该方法需要传入字符串参数，该字符串参数的值是某个类的全限定类名（必须添加完整包名）。

2. 调用某个类的class属性来获取该类对应的Class对象。例如Person.class将会返回Person类对应的Class对象。

3. 调用某个对象的getClass()方法，该方法是java.lang.Object类中的一个方法，所有Java对象都可以调用该方法，该方法将会返回该对象所属类对应的Class对象。

使用反射查看类信息:示例代码 ClassTest

使用反射生成与操作对象

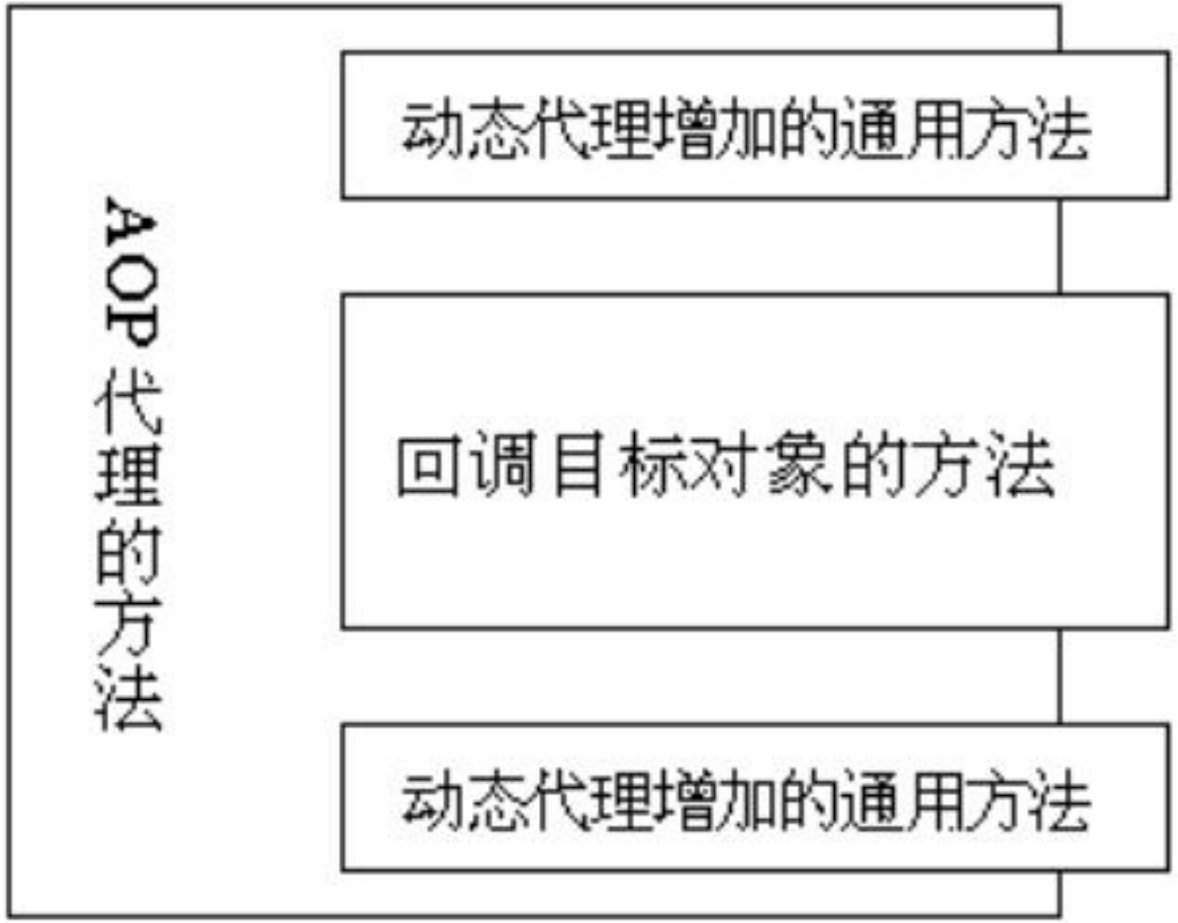
Class对象可获得：

1. 方法 Method对象
2. 构造器 Constructor对象
3. 成员变量 Field对象

以上三个类都位于java.lang.reflect包下

使用反射生成与操作对象: 示例代码 ReflectionTest

AOP动态代理（选学）



AOP动态代理（选学）

Proxy类： 创建动态代理类和动态代理实例：

static Object newProxyInstance(ClassLoader loader,Class<?>[] interfaces, InvocationHandler h)：直接创建一个动态代理对象，该代理对象的实现类实现了interfaces指定的系列接口，执行代理对象的每个方法时都会被替换执行InvocationHandler对象的invoke方法。

InvocationHandler类：通过invoke方法来替换被代理对象的方法

public Object invoke(Object proxy, Method method, Object[] args) throws Throwable：传入动态代理对象，正在执行的方法，目标方法的传入参数

示例代码 -1.3

谢谢观看

SeeYou！

SoftCITS 软件园IT文化沙龙

SOFT CULTURE IT SALON