

## Java动态代理

- 之前为大家讲解过代理机制的操作，属于静态代理，特征是代理类和目标对象的类都是在编译期间确定下来，不利于程序的扩展。同时，每一个代理类只能为一个接口服务，这样一来程序开发中必然产生过多的代理。<sup>1</sup>
- 最好可以通过一个代理类完成全部的代理功能

## Java动态代理

- 动态代理是指客户通过代理类来调用其它对象的方法，并且是在程序运行时根据需要动态创建目标类的代理对象。
- 动态代理使用场合：
  - 调试
  - 远程方法调用
- 代理设计模式的原理：

使用一个代理将对象包装起来,然后用该代理对象取代原始对象.任何对原始对象的调用都要通过代理.代理对象决定是否以及何时将方法调用转到原始对象上

反射是动态代理的一个关键

5.动态代理---反射的应用。体会反射的动态性

**代理设计模式的原理:**

使用一个代理将对象包装起来, 然后用该代理对象取代原始对象. 任何对原始对象的调用都要通过代理. 代理对象决定是否以及何时将方法调用转到原始对象上

**静态代理:**要求被代理类和代理类同时实现相应的一套接口; 通过被代理类的对象调用重写接口的方法时, 实际上执行的是被代理类的同样的方法的调用。

**动态代理:**在程序运行时, 根据被代理类及其实现的接口, 动态的创建一个代理类。当调用代理类的实现的抽象方法时, 就发起对被代理类同样方法的调用。

涉及到的技术点: ①提供一个实现了InvocationHandler接口实现类, 并重写其invoke()方法

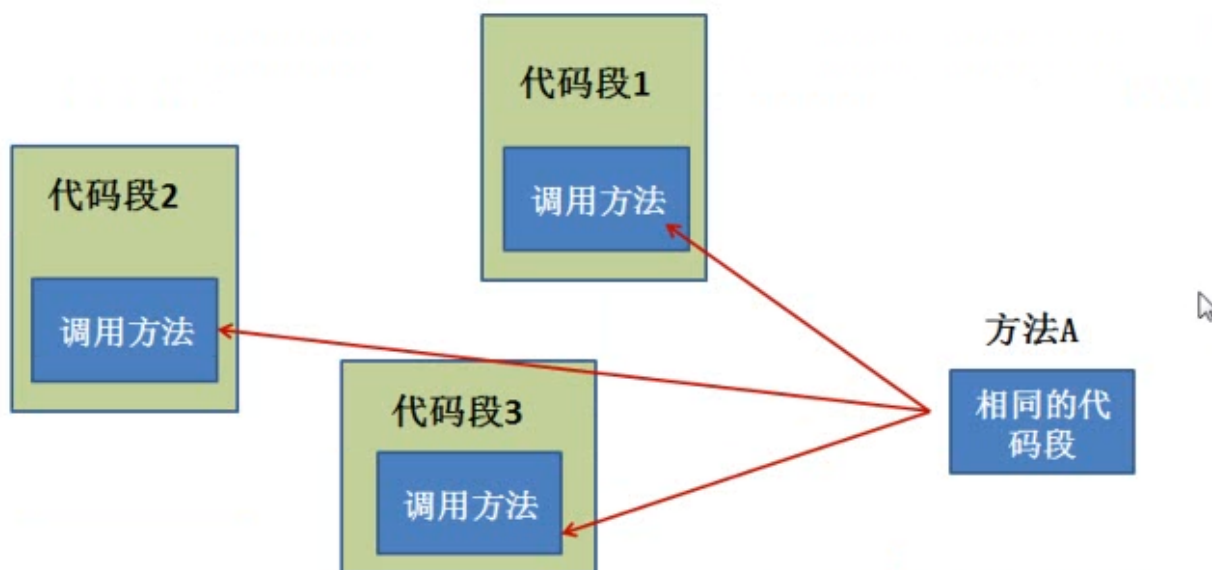
②Proxy.newProxyInstance(obj.getClass().getClassLoader(),obj.getClass().getInterfaces(),h);

//注: obj: 被代理类对象 ; h:实现了InvocationHandler接口的实现类的对象

例子见: day19-com.atguigu.java1.TestProxy.java例子

www.atguigu.com

## 动态代理与AOP (Aspect Orient Programming)



改进后的说明: 代码段1、代码段2、代码段3和深色代码段分离开了, 但代码段1、2、3又和一个特定的方法A耦合了! 最理想的效果是: 代码块1、2、3既可以执行方法A, 又无须在程序中以硬编码的方式直接调用深色代码的方法

## 动态代理与AOP (Aspect Orient Programming)

