java动态代理：一个简单的例子

首先创建一个接口，PersonDao

public interface PersonDao {

    public void say();

}

然后写一个实现类PersonDaoImpl

public class PersonDaoImpl implements PersonDao{

    @Override

    public void say() {

        System.out.println("time to eat");

    }

}

通过实现 InvocationHandler 接口创建自己的调用处理器：

public class PersonHandler implements InvocationHandler {

    private Object obj;

    public PersonHandler(Object obj){

        this.obj=obj;

    }

    @Override

    public Object invoke(Object proxy, Method method, Object[] args)  throws Throwable {

        System.out.println("before");

        Object result = method.invoke(obj, args);

        System.out.println("after");

        return result;

    }

}

然后再写个测试类

public class PersonTest {

    @Test

    public void test(){

        PersonDao pDao = new PersonDaoImpl();

        PersonHandler handler = new PersonHandler(pDao);

//为指定类装载器、一组接口及调用处理器生成动态代理对象

        PersonDao proxy = (PersonDao)Proxy.newProxyInstance(pDao.getClass().getClassLoader(), pDao.getClass().getInterfaces(), handler);

        proxy.say();

    }

}

最后控制台输出  
