AllLogAdvice.javapackage nuc.edu.cn.advice;
import java.text.SimpleDateFormat;
import java.util.Arrays;
import java.util.Date;
import java.util.List;
import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.AfterReturning;
import org.aspectj.lang.annotation.AfterThrowing;
import org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
import org.aspectj.lang.annotation.Pointcut;
import org.springframework.stereotype.Component;
@Aspect
@Component
public class AllLogAdvice {
@Pointcut("execution(\* nuc.edu.cn.service.IProductService.\*(..))")
private void allMethod() {
}
@Before("allMethod()")
public void myBeforeAdvice(JoinPoint joinPoint) {
List<Object> args = Arrays.asList(joinPoint.getArgs());
String loginInfoText = "前置通知:" + new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()) + " "
+ args.get(0).toString() + " 浏览商品： " + args.get(1).toString();
System.out.println(loginInfoText);
}
@After("allMethod()")
public void myAfterAdvice(JoinPoint joinPoint) {
List<Object> args = Arrays.asList(joinPoint.getArgs());
String loginInfoText = "后置通知:" + new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()) + " "
+ args.get(0).toString() + " 浏览商品： " + args.get(1).toString();
System.out.println(loginInfoText);
}
@AfterReturning("allMethod()")
public void myAfterReturnAdvice(JoinPoint joinPoint) {
List<Object> args = Arrays.asList(joinPoint.getArgs());
String loginInfoText = "返回通知:" + new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date()) + " "
+ args.get(0).toString() + " 浏览商品： " + args.get(1).toString();
System.out.println(loginInfoText);
}
@AfterThrowing(pointcut = "allMethod()", throwing = "e")
public void myThrowingAdvice(JoinPoint joinPoint, Exception e) {
String targetClassName = joinPoint.getTarget().getClass().getName();
String targetMethodName = joinPoint.getSignature().getName();
String loginInfoText = "异常通知 ִ调用" + targetClassName + " 类 " + targetMethodName + " 方法发生异常 ";
System.out.println(loginInfoText);
}
@Around("allMethod()")
public void myAroundAdvice(ProceedingJoinPoint joinPoint) throws Throwable {
long beginTime = System.currentTimeMillis();
joinPoint.proceed();
long endTime = System.currentTimeMillis();
String targetMethodName = joinPoint.getSignature().getName();
String loginInfoText = "环绕通知: " + targetMethodName + "方法被调用时间" + beginTime + "毫秒, " + " 执行完成时间 " + endTime
+ "毫秒. ";
System.out.println(loginInfoText);
}
}

Client.javapackage nuc.edu.cn.client;
import nuc.edu.cn.service.IUserService;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
public class Client {
public static void main(String[] args) {
ApplicationContext ac=new ClassPathXmlApplicationContext("applicationContext.xml");
IUserService userService= (IUserService) ac.getBean("userService");
userService.saveUser("张三");
}
}

TestAOP.javapackage nuc.edu.cn.client;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import nuc.edu.cn.service.IProductService;
public class TestAOP {
public static void main(String[] args) {
// TODO Auto-generated method stub
ApplicationContext ctx = new ClassPathXmlApplicationContext("applicationContext.xml");
IProductService productService = (IProductService) ctx.getBean("productService");
productService.browse("张三", "联想电脑");
}
}

UserDaoImpl.javapackage nuc.edu.cn.dao.impl;
import nuc.edu.cn.dao.IUserDao;
import org.springframework.stereotype.Repository;
@Repository
public class UserDaoImpl implements IUserDao {
public void saveUser(String uname) {
System.out.println("持久层被调用，接收参数："+uname);
}
}

IUserDao.javapackage nuc.edu.cn.dao;
public interface IUserDao {
public void saveUser(String uname);
}

ProductServiceImpl.javapackage nuc.edu.cn.service.impl;
import org.springframework.stereotype.Component;
import nuc.edu.cn.service.IProductService;
@Component("productService")
public class ProductServiceImpl implements IProductService {
public void browse(String loginName, String productName) {
System.out.println("ִbrowse方法被调用");
throw new RuntimeException("这是为测试抛出异常");
// int i = 100000000;
// while (i > 0) {
// i--;
// }
}
}

UserServiceImpl.javapackage nuc.edu.cn.service.impl;
import nuc.edu.cn.dao.IUserDao;
import nuc.edu.cn.service.IUserService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
@Service("userService")
public class UserServiceImpl implements IUserService {
@Autowired
private IUserDao userDao;
// public void setUserDao(IUserDao userDao) {
// this.userDao = userDao;
// }
public void saveUser(String uname) {
System.out.println("业务层被调用，接收参数："+uname);
userDao.saveUser(uname);
}
}

IProductService.javapackage nuc.edu.cn.service;
public interface IProductService {
public void browse(String loginName, String productName);
}

IUserService.javapackage nuc.edu.cn.service;
public interface IUserService {
public void saveUser(String uname);
}

test.java