

PRACTICAL 8(C) – SPRING AOP (AROUND ADVICE)

AIM

To demonstrate Spring AOP – Around Advice.

THEORY

Around Advice is the most powerful advice type in Spring AOP. It surrounds the execution of the target method and can control whether the method executes or not. It uses ProceedingJoinPoint.

SOURCE CODE – Target Class (BusinessLogic.java)

```
package aopdemo;

public class BusinessLogic {
    public void process() {
        System.out.println("Business logic executed");
    }
}
```

SOURCE CODE – Aspect Class (AroundAspect.java)

```
package aopdemo;

import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect;

@Aspect
public class AroundAspect {

    @Around("execution(* aopdemo.BusinessLogic.process(..))")
    public Object aroundAdvice(ProceedingJoinPoint pjp) throws Throwable {
        System.out.println("Before method execution (Around Advice)");
        Object result = pjp.proceed();
        System.out.println("After method execution (Around Advice)");
        return result;
    }
}
```

SOURCE CODE – applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.springframework.org/schema/aop
           http://www.springframework.org/schema/aop/spring-aop.xsd">

    <aop:aspectj-autoproxy />
```

```
<bean id="business" class="aopdemo.BusinessLogic"/>
<bean class="aopdemo.AroundAspect"/>
</beans>
```

SOURCE CODE – MainApp.java

```
package aopdemo;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {
    public static void main(String[] args) {
        ApplicationContext context =
            new ClassPathXmlApplicationContext("applicationContext.xml");

        BusinessLogic obj = (BusinessLogic) context.getBean("business");
        obj.process();
    }
}
```

NETBEANS 7 – EXECUTION STEPS

1. Use the same Spring AOP project and libraries from Practical 8(A)
2. Create AroundAspect.java in aopdemo package
3. Update applicationContext.xml
4. Run MainApp.java using Shift + F6

OUTPUT

Before method execution (Around Advice) Business logic executed After method execution (Around Advice)

RESULT

Thus, Spring AOP Around Advice was successfully implemented.