

Program No:	
Roll No :	1525
Title of Program :	
Objective :	Bank Autoproxy

SOURCE CODE:

appctx.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="acBean" class="edu.met.p1.BankAccount">
    <property name="acno" value="101"></property>
    <property name="acname" value="bhairav"></property>
    <property name="acbal" value="2000"></property>

</bean>

</beans>
```

BankMain.java

```
package edu.met.p1;
import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationConte
xt;

public class BankMain {

    static ApplicationContext ctx;
    public static void main(String[] args)
    {
```

```

        // TODO Auto-generated method stub
        ctx=new ClassPathXmlApplicationContext("appctx.xml");
        BankAccount b1 = (BankAccount)ctx.getBean("acBean");
        System.out.println(b1.toString());
    }
}

```

BankAspect.java

```

package edu.met.p1;

import org.aspectj.lang.annotation.*;
import org.aspectj.lang.JoinPoint;

@Aspect
public class BankAspect
{
    // Define a pointcut for all setter methods in BankAccount
    @Pointcut("execution(* edu.met.p1.BankAccount.set*(..))")
    public void getPC()
    {
        // This method is empty because it's just a pointcut
    }

    // Advice that runs before the matched methods
    @Before("getPC()")
    public void beforeMethod(JoinPoint joinPoint)
    {
        // Log the method being called
        System.out.println("Before calling: " );
    }
}

```

BankAccount.java

```

package edu.met.p1;

```

```
public class BankAccount {  
    int acno;  
    String acname;  
    double acbal;  
  
    public int getAcno() {  
        return acno;  
    }  
  
    public void setAcno(int acno) {  
        this.acno = acno;  
    }  
  
    public String getAcname() {  
        return acname;  
    }  
  
    public void setAcname(String acname) {  
        this.acname = acname;  
    }  
  
    public double getAcbal() {  
        return acbal;  
    }  
  
    public void setAcbal(double acbal) {  
        this.acbal = acbal;  
    }  
}
```

```

public BankAccount() {
    // TODO Auto-generated constructor stub
}

public void deposit(double amt)
{
    System.out.println("Amount Deposited");
    this.acbal+=amt;
}
public void withdraw(double amt)
{
    if(this.acbal-amt>=0)
    {
        System.out.println("Amount withdrawn");
        this.acbal-=amt;

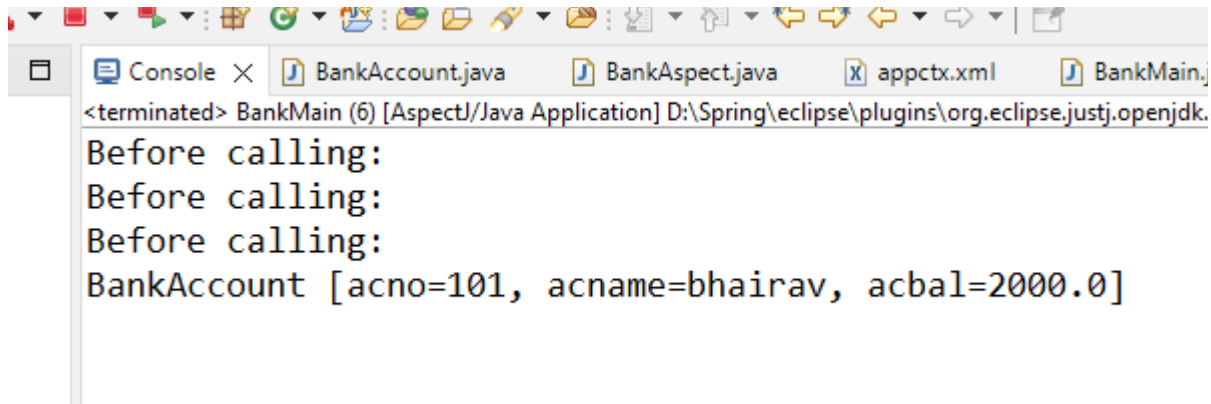
    }
    else
        throw new RuntimeException();
}

@Override
public String toString() {
    return "BankAccount [acno=" + acno + ", acname=" +
acname + ", acbal=" + acbal + "];"
}

}

```

OUTPUT:



```
<terminated> BankMain (6) [AspectJ/Java Application] D:\Spring\eclipse\plugins\org.eclipse.justj.openjdk.  
Before calling:  
Before calling:  
Before calling:  
BankAccount [acno=101, acname=bhairav, acbal=2000.0]
```

FOR WITHDRAWL

BankMain.java

```
package edu.met.p1;  
import org.springframework.context.ApplicationContext;  
import  
org.springframework.context.support.ClassPathXmlApplicationConte  
xt;  
  
public class BankMain {  
  
    static ApplicationContext ctx;  
    public static void main(String[] args)  
    {  
        // TODO Auto-generated method stub  
        ctx=new ClassPathXmlApplicationContext("appctx.xml");  
        BankAccount b1 = (BankAccount)ctx.getBean("acBean");  
        System.out.println(b1.toString());  
        b1.withdraw(1000);  
        System.out.println("After: "+b1);  
    }  
  
}
```

BankAspect.java

```
package edu.met.p1;
```

```
import org.aspectj.lang.annotation.*;
```

```
import org.aspectj.lang.JoinPoint;
```

```
@Aspect
```

```
public class BankAspect
```

```
{
```

```
    // Define a pointcut for all setter methods in BankAccount
```

```
    @Pointcut("execution(* edu.met.p1.BankAccount.set*(..))")
```

```
    public void setterMethods()
```

```
    {
```

```
        // This method is empty because it's just a pointcut
```

```
    }
```

```
    // Advice that runs before the matched methods
```

```
    @Before("setterMethods()")
```

```
    public void beforeSetterMethod(JoinPoint joinPoint)
```

```
    {
```

```
        // Log the method being called
```

```
        System.out.println("Before calling: " +  
joinPoint.getSignature().getName());
```

```
    }
```

```
    @After("setterMethods()")
```

```
    public void afterSetterMethod(JoinPoint joinPoint)
```

```
    {
```

```
        // Log the method being called
```

```
        System.out.println("After calling: " +  
joinPoint.getSignature().getName());
```

```
    }
```

```
    // Pointcut for the withdraw method
```

```
    @Pointcut("execution(*
```

```
edu.met.p1.BankAccount.withdraw(..))")
```

```
    public void withdrawMethod()
```

```
    {
```

```
        // This method is empty because it's just a pointcut
```

```
    }
```

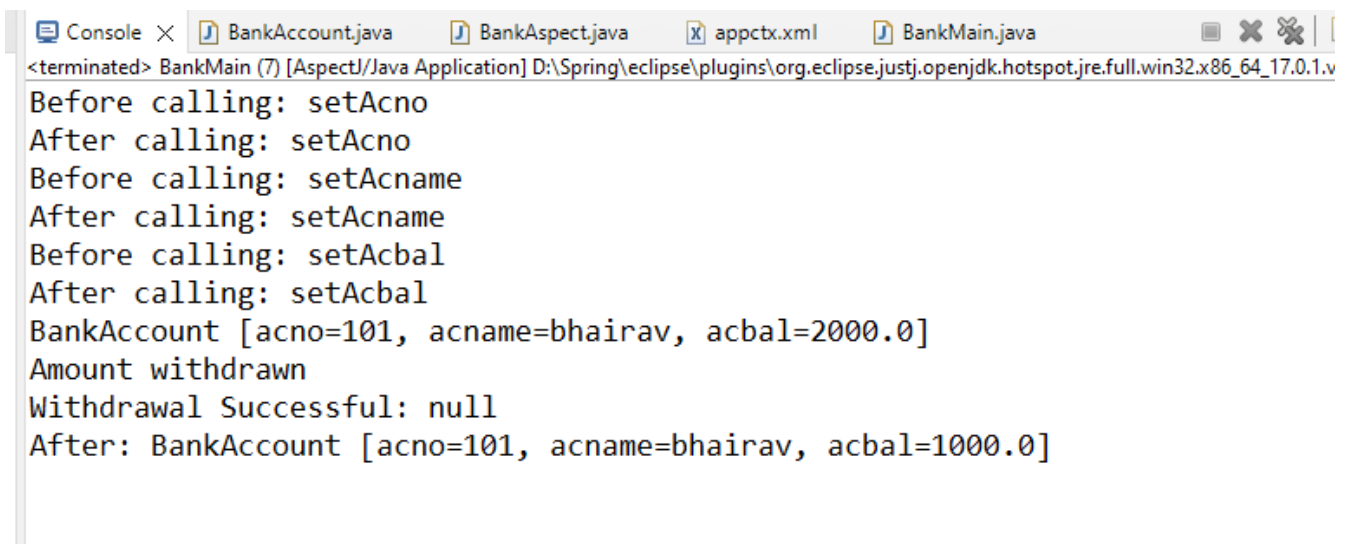
```

    // Advice for successful withdrawal
    @AfterReturning(pointcut = "withdrawMethod()", returning =
"result")
    public void afterSuccessfulWithdrawal(JoinPoint joinPoint,
Object result)
    {
        System.out.println("Withdrawal Successful: " + result);
    }

    // Advice for unsuccessful withdrawal
    @AfterThrowing(pointcut = "withdrawMethod()", throwing =
"ex")
    public void afterFailedWithdrawal(JoinPoint joinPoint,
Throwable ex)
    {
        System.out.println("Withdrawal Unsuccessful: " +
ex.getMessage());
    }
}

```

LATEST:



```

Console × BankAccount.java BankAspect.java appctx.xml BankMain.java
<terminated> BankMain (7) [AspectJ/Java Application] D:\Spring\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v
Before calling: setAcno
After calling: setAcno
Before calling: setAcname
After calling: setAcname
Before calling: setAcbal
After calling: setAcbal
BankAccount [acno=101, acname=bhairav, acbal=2000.0]
Amount withdrawn
Withdrawal Successful: null
After: BankAccount [acno=101, acname=bhairav, acbal=1000.0]

```

