AOP using Decorator Pattern

# Introduction

You can also achieve AOP in core java using Decorator Design Pattern. The example is given below.

# AbstractAccount.java

**package** com.ddlab.rnd.aop1;  
  
**public abstract class** AbstractAccount {  
  
 **public abstract void** withdraw( String actNo , **int** amount);  
}

# RetailAccount.java

**package** com.ddlab.rnd.aop1;  
  
**public class** RetailAccount **extends** AbstractAccount {  
  
 @Override  
 **public void** withdraw(String actNo, **int** amount) {  
  
 System.***out***.println(**"Amount "** + amount + **" is debited from the account no "** + actNo);  
 }  
}

# BeforeDecorator.java

**package** com.ddlab.rnd.aop1;  
  
**public class** BeforeDecorator **extends** AbstractAccount {  
  
 **private** RetailAccount **retailAccount**;  
  
 **public** BeforeDecorator(RetailAccount retailAccount) {  
 **this**.**retailAccount** = retailAccount;  
 }  
  
 @Override  
 **public void** withdraw(String actNo, **int** amount) {  
 System.***out***.println(**"Doing validation before debiting from acount..."**);  
 **retailAccount**.withdraw(actNo,amount);  
 }  
}

# AroundDecorator.java

**package** com.ddlab.rnd.aop1;  
  
**public class** AroundDecorator **extends** AbstractAccount {  
  
 **private** RetailAccount **retailAccount**;  
  
 **public** AroundDecorator(RetailAccount retailAccount) {  
 **this**.**retailAccount** = retailAccount;  
 }  
  
 @Override  
 **public void** withdraw(String actNo, **int** amount) {  
  
 System.***out***.println(**"Doing before withdrawing money ........."**);  
 **retailAccount**.withdraw(actNo, amount);  
 System.***out***.println(**"Doing after withdrawing money ........."**);  
  
 }  
}

# TestAOPDecorator.java

**package** com.ddlab.rnd.aop1;  
  
**public class** TestAOPDecorator {  
  
 **public static void** main(String[] args) {  
  
 RetailAccount account = **new** RetailAccount();  
*// BeforeDecorator beforeDecorator = new BeforeDecorator(account);* AroundDecorator aroundDecorator = **new** AroundDecorator(account);  
 aroundDecorator.withdraw(**"1122334455"**,500);  
 }  
}