**设计模式 实验1-策略模式**

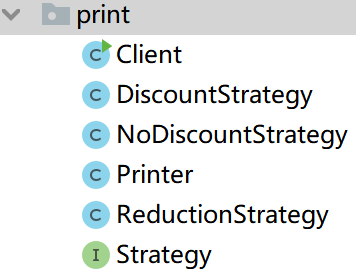
一.简答题（共1题）

某公司专门销售各种打印机，销售打印机时都有一定的折扣让利给顾客，但折扣计算的方法有很多种，如：不打折；每台减扣固定的金额；按售价的5%打折等等，且折扣计算方法可能发生变化。现在要开发该公司的销售系统，请你设计一个方案，实现打印机销售时的折扣计算，你的方案最好要能够使得在销售打印机（即使是同一种打印机）时可以灵活的选用折扣计算方法，并且可以很容易地增加或修改折扣计算方法而不至于对整个系统的维护造成困难。请用strategy模式设计解决方案，给出Java源程序，绘制相应的类图，编写实验报告。实验报告中需要给出源程序截图、类图和运行结果截图，以及相应的文字说明。源程序一起打包。

1. 类图



1. 源程序



①声明策略接口Strategy,声明打折方法getPrice

*/\*\*  
 \* 抽象策略类  
 \*/***public interface** Strategy {  
 **double** getPrice(**double** price); *//策略方法*}

②各种Strategy的实现类,实现不同的打折方式

*/\*\*  
 \* 具体策略类-不打折  
 \*/***public class** NoDiscountStrategy **implements** Strategy {  
 @Override  
 **public double** getPrice(**double** price) {  
 System.***out***.println(**"无折扣优惠"**);  
 **return** price;  
 }  
}

*/\*\*  
 \* 具体策略类-每台减扣固定的金额  
 \*/***public class** ReductionStrategy **implements** Strategy {  
 **private double reduction**=100d;  
 @Override  
 **public double** getPrice(**double** price) {  
 System.***out***.println(**"每台减扣固定的金额"**);  
 **return** price-**reduction**;  
 }  
}

*/\*\*  
 \* 具体策略类-打折5%  
 \*/***public class** DiscountStrategy **implements** Strategy {  
 **private double rate**=0.05d;  
 @Override  
 **public double** getPrice(**double** price) {  
 System.***out***.println(**"5%折扣优惠"**);  
 **return** price \* (1-**rate**);  
 }  
}

③编写环境类Printer

*/\*\*  
 \* 环境类  
 \*/***public class** Printer {  
 **private** Strategy **strategy**;**public void** setStrategy(Strategy strategy) {  
 **this**.**strategy** = strategy;  
 }  
 **public double** getPrice(**double** price){  
 **return strategy**.getPrice(price);  
 }  
}

④使用Printer来查看当它改变策略Strategy时的行为变化

*/\*\*  
 \* 客户端  
 \*/***public class** Client {  
 **public static void** main(String[] args) {  
 Printer c=**new** Printer();*//持有抽象策略的引用来调用* Strategy s;  
 **double** price=1000d;  
 s=**new** NoDiscountStrategy();  
 c.setStrategy(s);  
 System.***out***.println(c.getPrice(price));  
 System.***out***.println(**"-----------------"**);  
 s=**new** ReductionStrategy();  
 c.setStrategy(s);  
 System.***out***.println(c.getPrice(price));  
 System.***out***.println(**"-----------------"**);  
 s=**new** DiscountStrategy();  
 c.setStrategy(s);  
 System.***out***.println(c.getPrice(price));  
 }  
}

1. 运行结果截图

