Name of the Course : Java 9 Core principles

Level : Moderate

Tool Stack : List interface, ArrayList, Abstarction and Encapsulation, Polymorphism, and access specifiers

Problem Statement :

Lemon Tree Hotel is one of the fastest growing Hotel. It needs to develop software application for billing in the Hotel. And for different bank Atm card have to give discount on total bill. As start up, they need to automate the Hotel Billing system.

Description : You are provided with a public class FoodProduct with private attributes :

**int** foodId;

String foodName;

**double** costPerUnit;

**int** quantity;

Appropriate public getters and setters are already written.

You are provided with a public class Order with private attribute :

**double** discountPercentage;

List<FoodProduct> foodList

Appropriate public getters and setters are already written.

Write a **public** **Boolean** findDiscount(String bankName)in Order class as, **public** **Boolean** findDiscount(String bankName)

this method will return discount percentage according to respected bank.

Write a **public** **boolean** addToCart(FoodProduct foodProductObject) in Order class used to add multiple food product details in list.

Write a **public** **double** calculateTotalBill() in Order class as,

**public** **double** calculateTotalBill()

Use this to find final bill amount

bill=bill-((bill\*discountPercentage)/100)

Bill amount is calculated as follows :

If the Bank name selected as “SBI” then 50% discount on total bill.

If the Bank name selected as “HDFC” then 15% discount on total bill.

If the Bank name selected as “ICICI” then 25% discount on total bill.

If the Bank name selected as “CUB” then 30% discount on total bill.

If the Bank name selected as “OTHERS” then 0% discount on total bill.

You are provided with a public class UserInterface which has the main method.

Check the correctness of the methods written in these classes.

Note : All class, methods needs to be declared as public

**Code:**

**package** main.java.yaksha;

**public** **class** FoodProduct {

**private** **int** foodId;

**private** String foodName;

**private** **double** costPerUnit;

**private** **int** quantity;

**public** **int** getFoodId() {

**return** foodId;

}

**public** **void** setFoodId(**int** foodId) {

**this**.foodId = foodId;

}

**public** String getFoodName() {

**return** foodName;

}

**public** **void** setFoodName(String foodName) {

**this**.foodName = foodName;

}

**public** **double** getCostPerUnit() {

**return** costPerUnit;

}

**public** **void** setCostPerUnit(**double** costPerUnit) {

**this**.costPerUnit = costPerUnit;

}

**public** **int** getQuantity() {

**return** quantity;

}

**public** **void** setQuantity(**int** quantity) {

**this**.quantity = quantity;

}

}

**import** java.util.ArrayList;

**import** java.util.List;

**import** main.java.yaksha.FoodProduct;

**import** main.java.yaksha.Order;

**public** **class** Order{

**private** **double** discountPercentage;

**private** List<FoodProduct> foodList=**new** ArrayList<FoodProduct>();

**public** **double** getDiscountPercentage() {

**return** discountPercentage;

}

**public** **void** setDiscountPercentage(**double** discountPercentage) {

**this**.discountPercentage = discountPercentage;

}

**public** List<FoodProduct> getFoodList() {

**return** foodList;

}

**public** **void** setFoodList(List<FoodProduct> foodList) {

**this**.foodList = foodList;

}

**public** **boolean** findDiscount(String bankName)

{

**if**(bankName.equals("HDFC")) {

discountPercentage=15.0;

}

**else** **if**(bankName.equals("ICICI")) {

discountPercentage=25.0;

}

**else** **if**(bankName.equals("CUB")) {

discountPercentage=30.0;

}

**else** **if**(bankName.equals("SBI")) {

discountPercentage=50.0;

}

**else** **if**(bankName.equals("OTHERS")) {

discountPercentage=0.0;

}

setDiscountPercentage(discountPercentage);

**return** **true**;

}

**public** **boolean** addToCart(FoodProduct foodProductObject)

{

List<FoodProduct> f=getFoodList();

f.add(foodProductObject);

setFoodList(f);

**return** **true**;

}

**public** **double** calculateTotalBill()

{

**double** bill = 0;

List<FoodProduct> f=getFoodList();

**for**(**int** i=0;i<f.size();i++)

{

// System.out.println(f.get(i).getCostPerUnit());

// System.out.println(f.get(i).getQuantity());

bill+=f.get(i).getQuantity()\*f.get(i).getCostPerUnit()\*1.0;

}

// System.out.println(bill);

// System.out.println(dis);

bill=bill-((bill\*getDiscountPercentage())/100);

**return** bill;

}

}

package main.java.yaksha;

import java.util.Scanner;

import main.java.yaksha.FoodProduct;

import main.java.yaksha.Order;

public class UserInterface{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int itemno;

String bank;

System.out.println("Enter the number of items");

itemno=sc.nextInt();

System.out.println("Enter the item details");

Order o=new Order();

for(int i=0;i<itemno;i++)

{

FoodProduct fd=new FoodProduct();

System.out.println("Enter the item id");

fd.setFoodId(sc.nextInt());

System.out.println("Enter the item name");

fd.setFoodName(sc.next());

System.out.println("Enter the cost per unit");

fd.setCostPerUnit(sc.nextDouble());

System.out.println("Enter the quantity");

fd.setQuantity(sc.nextInt());

o.addToCart(fd);

}

System.out.println("Enter the bank name to avail offer");

bank=sc.next();

o.findDiscount(bank);

System.out.println("Calculated Bill Amount:"+o.calculateTotalBill());

}

}

Junit Testing

**package** test.java.yaksha;

**import** main.java.yaksha.FoodProduct;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.List;

**import** org.junit.jupiter.api.Test;

**import** main.java.yaksha.Order;

**import** main.java.yaksha.TestUtils;

**class** UserInterfaceTest {

@Test

**void** testFindDiscount() **throws** IOException {

Order o=**new** Order();

assertEquals(true,o.findDiscount("SBI") );

}

@Test

**void** testAddToCart() **throws** IOException {

Order o=**new** Order();

FoodProduct foodProductObject=**new** FoodProduct() ;

assertEquals(true,o.addToCart(foodProductObject) );

}

@Test

**public** **void** testCalculateTotalBill() **throws** IOException {

FoodProduct f=**new** FoodProduct();

Order o=**new** Order();

f.setFoodId(1);

f.setFoodName("Rice");

f.setCostPerUnit(100);

f.setQuantity(2);

o.setDiscountPercentage(50.0);

o.addToCart(f);

assertEquals(100, o.calculateTotalBill());

}

}

Test Data1

Enter the number of items

2

Enter the item details

Enter the item id

1

Enter the item name

Rice

Enter the cost per unit

100

Enter the quantity

1

Enter the item id

2

Enter the item name

DalFry

Enter the cost per unit

100

Enter the quantity

1

Enter the bank name to avail offer

SBI

Calculated Bill Amount:100.0

Learning outcome: Participant could able to learn how to use List interface, ArrayList class method as well as encapsulation and access specifires. Participant also learn how to add multiple object in the List.