|  |
| --- |
| import java.util.\*; |
|  | import java.lang.\*; |
|  |  |
|  | abstract class DesertItem { |
|  |  |
|  | abstract public int getCost(int desertItem); |
|  | } |
|  | class Candy extends DesertItem { |
|  | public int addCandies(int candies){ |
|  | return candies; |
|  | } |
|  | @Override |
|  | public int getCost(int candies) { |
|  | int dollar =60; |
|  | int cost=candies\*dollar; |
|  | return cost; |
|  | } |
|  | } |
|  |  |
|  | class Cookie extends DesertItem { |
|  | public int addCookies(int cookies){ |
|  | return cookies; |
|  | } |
|  | @Override |
|  | public int getCost(int cookies) { |
|  | int euro=70; |
|  | int cost=cookies\*euro; |
|  | return cost; |
|  | } |
|  | } |
|  | class IceCream extends DesertItem { |
|  | public int addIceCreams(int icecreams){ |
|  | return icecreams; |
|  | } |
|  | @Override |
|  | public int getCost(int icecreams) { |
|  | int rupees=65; |
|  | int cost=icecreams\*rupees; |
|  | return cost; |
|  | } |
|  | } |
|  |  |
|  | public class Assignment2Q7 { |
|  | static int Cookies=0; |
|  | static int Candies=0; |
|  | static int Icecreams=0; |
|  | Cookie cookie=new Cookie(); |
|  | Candy candy=new Candy(); |
|  | IceCream icecream=new IceCream(); |
|  |  |
|  | public static void main(String[] args) { |
|  | Assignment2Q7 ob=new Assignment2Q7(); |
|  | ob.selectRoles(); |
|  | } |
|  | Scanner sc=new Scanner(System.in); |
|  | private void selectRoles(){ |
|  | String role=""; |
|  | System.out.println("Items:\nCandies: "+Candies+ "\nCookies: "+Cookies+ "\nIcecreams: "+Icecreams); |
|  | System.out.println("Enter your Role: "); |
|  | role=sc.next(); |
|  | role=role.toLowerCase(); |
|  | roles(role); |
|  | } |
|  |  |
|  | private void roles(String role){ |
|  | if(role.equals("owner")) { |
|  | addItems(); |
|  | } |
|  | else if(role.equals("customer")) { |
|  | placeOrder(); |
|  | } |
|  | else { |
|  | System.out.println("Please provide a valid Input(ownew/customer)"); |
|  | selectRoles(); |
|  | } |
|  | } |
|  |  |
|  | private void addItems() { |
|  | System.out.println("Select the Deserts you want to add\n1)- Candy\n2)- Cookies\n3)- Icecream\n4)- Main Menue\n5)-Quit"); |
|  | int item; |
|  | item=sc.nextInt(); |
|  | addItemsOperation(item); |
|  | } |
|  |  |
|  | private void addItemsOperation(int choice) { |
|  | switch(choice) { |
|  | case 1: |
|  | System.out.println("Enter Number of Candies to be addes: "); |
|  | int num=sc.nextInt(); |
|  | Candies =num; |
|  | candy.addCandies(Candies); |
|  | addItems(); |
|  | break; |
|  | case 2: |
|  | System.out.println("Enter Number of Cookies to be addes: "); |
|  | int num1=sc.nextInt(); |
|  | Cookies =num1; |
|  | cookie.addCookies(Cookies); |
|  | addItems(); |
|  | break; |
|  | case 3: |
|  | System.out.println("Enter Number of IceCreams to be addes: "); |
|  | int num2=sc.nextInt(); |
|  | Icecreams =num2; |
|  | icecream.addIceCreams(num2); |
|  | addItems(); |
|  | break; |
|  | case 4: |
|  | selectRoles(); |
|  | case 5: |
|  | System.exit(0); |
|  | default: |
|  | System.out.println("Please provide a valid input!!"); |
|  | addItems(); |
|  | break; |
|  | } |
|  | } |
|  | private void placeOrder() { |
|  | if(Candies!=0 || Cookies!=0 || Icecreams!=0 || Candies+Cookies+Icecreams!=0 ) { |
|  | System.out.println("What deserts do you want to buy: \n1)- Candy \n2)- Cookies \n3)- Icecream \n4)Back to Main Menu"); |
|  | int ind=sc.nextInt(); |
|  | placeOrderOperation(ind); |
|  | } |
|  | else { |
|  | System.out.println("stock is Empty.\nPlease tell the owner to add items."); |
|  | selectRoles(); |
|  | } |
|  | } |
|  | private void placeOrderOperation(int choice) { |
|  | switch(choice) { |
|  | case 1: |
|  | System.out.println("Candies are sold in Dollars. \n 1 candy costs 1 dollar"); |
|  | System.out.println("Enter number of candies to be bought: "); |
|  | int candies=sc.nextInt(); |
|  | if(candies<Candies) |
|  | { |
|  | Candies=Candies-candies; |
|  | System.out.println("Cost in Rupees: "+candy.getCost(candies)); |
|  | } |
|  | else |
|  | { |
|  | System.out.println("Please enter a number according to the stock"); |
|  | } |
|  | placeOrder(); |
|  | break; |
|  |  |
|  | case 2: |
|  | System.out.println("Cookies are sold in Euros. \n 1 cookie costs 1 euro"); |
|  | System.out.println("Enter number of cookies to be bought: "); |
|  | int cookies=sc.nextInt(); |
|  | if(cookies<Cookies) |
|  | { |
|  | Cookies=Cookies-cookies; |
|  | System.out.println("Cost in Rupees: "+cookie.getCost(cookies)); |
|  | } |
|  | else |
|  | { |
|  | System.out.println("Please enter a number according to the stock"); |
|  | } |
|  | placeOrder(); |
|  | break; |
|  |  |
|  | case 3: |
|  | System.out.println("Icecreams are sold in Rupees. \n 1 icecream costs 65 rupees"); |
|  | System.out.println("Enter number of Icecreams to be bought: "); |
|  | int icecreams=sc.nextInt(); |
|  | if(icecreams<Icecreams) |
|  | { |
|  | Icecreams=Icecreams-icecreams; |
|  | System.out.println("Cost in Rupees: "+icecream.getCost(icecreams)); |
|  | } |
|  | else |
|  | { |
|  | System.out.println("Please enter a number according to the stock"); |
|  | } |
|  | placeOrder(); |
|  | break; |
|  |  |
|  | case 4: |
|  | selectRoles(); |
|  |  |
|  | default: |
|  | System.out.println("Please enter a valid choice!"); |
|  | placeOrder(); |
|  | break; |
|  | } |
|  | } |
|  | } |