Card.java

import java.text.SimpleDateFormat;

import java.util.Date;

import java.text.ParseException;

import java.util.Calendar;

import java.util.Locale;

import static java.util.Calendar.\*;

import java.util.Date;

import java.lang.String.\*;

import static java.lang.Math.abs;

//import all the necessary modules

public class Card

{

//initialise all necessary variables

private String id;

private String name;

//Calendar to get the current date

private Calendar creationCalendar = Calendar.getInstance();

private Date createddate = null;

protected int years;

private SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");

private address full;

private int balance;

protected double discount;

private double coupon;

protected double couponRate;

protected String CardType;

protected StringBuilder fullDetail = new StringBuilder();

//no parameter constructor just in case

public Card ()

{

}

//constructor for the class card

public Card (String id, String name, String date, address address, int balance)

{

this.id = id;

this.balance = balance;

this.name = name;

full = address;

Calendar todaycalendar = Calendar.getInstance();

Date today = todaycalendar.getTime();

try {

createddate = dateFormat.parse(date);

} catch (ParseException e) {

System.out.println("error can't parse");

}

creationCalendar.setTime(createddate);

years = Math.floor((((todaycalendar.getTimeInMillis() - creationCalendar.getTimeInMillis())/ (24 \* 60 \* 60 \* 1000)) / 356));

}

public String toString()

{

fullDetail.append("Your card id is " + id).append(System.getProperty("line.separator"));

fullDetail.append("Your name is " + name).append(System.getProperty("line.separator"));

fullDetail.append("You have spent " + balance).append(System.getProperty("line.separator"));

fullDetail.append("This card was created on " + dateFormat.format(createddate)).append(System.getProperty("line.separator"));

fullDetail.append("You have been with us for " + years + " years").append(System.getProperty("line.separator"));

fullDetail.append("Your discount for each purchase is " + discount\*100 + "%").append(System.getProperty("line.separator"));

fullDetail.append("Your full address is " + full).append(System.getProperty("line.separator"));

fullDetail.append("Your card type is " + CardType).append(System.getProperty("line.separator"));

fullDetail.append("Your coupon is " + Math.round(coupon) + "$").append(System.getProperty("line.separator"));

return fullDetail.toString();

//return (id + "\r\n" + name + " " + years + " " + balance + " " + type );

}

public double getDiscount()

{

return discount;

}

public double getBalance()

{

return balance;

}

public String getID()

{

return id;

}

public void calCoupon()

{

coupon = balance\*couponRate;

}

public double getCouponRate()

{

return couponRate;

}

public int getYear()

{

return years;

}

public void printCoupon()

{

System.out.println(this);

}

}

BasicCard.java

//adding Basic card subclass

public class BasicCard extends Card

{

private static final int annualfee = 5;

//add BasicCard constructor

public BasicCard (String id, String name, String date, address address, int balance)

{

//calling the parent constructor

super(id, name, date, address, balance);

discount = 0.03;

CardType = "Basic";

if (balance <2000)

{

couponRate = 0.02;

}

else if ( balance >=2000)

{

couponRate = 0.03;

}

fullDetail.append("Your annual fee is " + annualfee + "$").append(System.getProperty("line.separator"));

fullDetail.append("Your coupon rate is " + couponRate + "%").append(System.getProperty("line.separator"));

}

}

SilverCard.java

//adding Silver card subclass

public class SilverCard extends Card

{

private static final int annualfee = 10;

//adding Constructor

public SilverCard (String id, String name, String date, address address, int balance)

{

super(id, name, date, address, balance);

discount = 0.05;

CardType = "Silver";

//Calculate discount rate

if (balance < 3500)

{

couponRate = 0.04;

}

else if ( balance >= 3500 && years < 3)

{

couponRate = 0.05;

}

else if ( balance >= 3500 && years >= 3)

{

couponRate = 0.06;

}

fullDetail.append("Your annual fee is " + annualfee + "$").append(System.getProperty("line.separator"));

fullDetail.append("Your coupon rate is " + couponRate + "%").append(System.getProperty("line.separator"));

}

}

GoldCard.java

//adding Gold card subclass

public class GoldCard extends Card

{

private double yearlyBalance;

//adding Constructor

public GoldCard (String id, String name, String date, address address, int balance, double yearlyBalance)

{

super(id, name, date,address,balance);

discount = 0.1;

CardType = "Gold";

this.yearlyBalance = yearlyBalance;

//Calculate discount rate

if (yearlyBalance < 3000)

{

couponRate = 0.1;

}

else if ( yearlyBalance >= 3000 && years < 3)

{

couponRate = 0.15;

}

else if ( yearlyBalance >= 3000 && years >= 3)

{

couponRate = 0.2;

}

fullDetail.append("Your yearly balance is " + yearlyBalance + "$").append(System.getProperty("line.separator"));

fullDetail.append("Your coupon rate is " + couponRate + "%").append(System.getProperty("line.separator"));

}

}

Address.java

import java.util.Date;

public class address

{

//adding core variables

private int streetnum;

private String streetname;

private String suburb;

private String city;

private String state;

private int postcode;

//adding class constructor using string as input

public address(String full)

{

String[] address = full.split("/",6);

this.streetnum = Integer.parseInt(address[0]);

this.streetname = address[1];

this.suburb = address[2];

this.city = address[3];

this.state = address[4];

this.postcode = Integer.parseInt(address[5]);

}

//toString() method

public String toString()

{

return (streetnum + " " + streetname + " " + suburb + " " + city + " " + state + " " + postcode);

}

}

Test.java

import java.util.\*;

public class test{

public static void main(String[] args)

{

//initializing all the cards

Card basic1=new BasicCard("12","Andrew High","12-12-2015", new address("12/Lorne Avenue/Warwick Farm/Darwin/NSW/2000"),1800);

Card basic2=new BasicCard("14","Andrew is High","12-12-2011", new address("89/George street/Victoria/Kusher/VIC/2607"),4000);

Card silver1=new SilverCard("15","Andrew is not High","23-11-2009", new address("90/Jemama street/Tash/Tashmania/KAW/2309"),3000);

Card silver2=new SilverCard("16","Andrew is not very High","23-11-2007", new address ("90/Auntie street/Mascot/Brishbain/CAI/2358"),4000);

Card gold1 = new GoldCard("17", "Andrews are high","23-09-2001", new address("120/Freak street/Central/Gold Coast/CAR/8703"), 50000, 2000);

Card gold2 = new GoldCard("18", "Andrews are not high", "18-07-2002", new address("183/Garden street/Greener/Town Hall/SHO/9052"), 8000, 5000);

//Run calCoupon to calculate coupon

basic1.calCoupon();

basic2.calCoupon();

silver1.calCoupon();

silver2.calCoupon();

gold1.calCoupon();

gold2.calCoupon();

//Running printCoupon to get all the information about the coupons and the cards

basic1.printCoupon();

System.out.println();

basic2.printCoupon();

System.out.println();

silver1.printCoupon();

System.out.println();

silver2.printCoupon();

System.out.println();

gold1.printCoupon();

System.out.println();

gold2.printCoupon();

//Creating the Card ArrayList and adding all the cards to the ArrayList

ArrayList<Card> calculate = new ArrayList<Card>();

calculate.add(basic1);

calculate.add(basic2);

calculate.add(silver1);

calculate.add(silver2);

calculate.add(gold1);

calculate.add(gold2);

System.out.println();

//initializing the total of all the discount

int total = 0;

for (Card i : calculate)

{

total += i.getBalance()\*i.getDiscount();

System.out.println("Card " + i.getID() + " discount " + (i.getBalance()\*i.getDiscount()));

}

System.out.println("Total discount is " + total + "$");

}

}

UML diagram

