**Vivek**

public abstract class Bero {

protected String beroType;

protected String beroColour;

protected double price;

Bero(String beroType, String beroColour){

this.beroType=beroType;

this.beroColour=beroColour;

}

public String getBeroType(){

return beroType;

}

public void setBeroType(String beroType){

this.beroType=beroType;

}

public String getBeroColour(){

return beroColour;

}

public void setBeroColour(String beroColour){

this.beroColour=beroColour;

}

public double getPrice(){

return price;

}

public void setPrice(double price){

this.price=price;

}

public abstract void calculatePrice();

}

public class CustomerDetails {

private String customerName;

private long phoneNumber;

private String address;

public CustomerDetails(String customerName, long phoneNumber, String address){

this.customerName=customerName;

this.phoneNumber=phoneNumber;

this.address=address;

}

public String getCustomerName(){

return customerName;

}

public void setCustomerName(String customerName){

this.customerName=customerName;

}

public long getPhoneNumber(){

return phoneNumber;

}

public void setPhoneNumber(long phoneNumber){

this.phoneNumber=phoneNumber;

}

public String getAddress(){

return address;

}

public void setAddress(String address){

this.address=address;

}

}

public class Discount {

public double calculateDiscount(Bero bObj) {

double discount=0;

if(bObj instanceof SteelBero){

discount=.10\*bObj.getPrice();

}

else if (bObj instanceof WoodenBero){

discount=.15\*bObj.getPrice();

}

return discount;

}

}

public class SteelBero extends Bero{

private int beroHeight;

public int getBeroHeight(){

return beroHeight;

}

public void setBeroHeight(int beroHeight){

this.beroHeight=beroHeight;

}

public SteelBero(String beroType,String beroColour,int beroHeight){

super(beroType,beroColour);

this.beroHeight=beroHeight;

}

public void calculatePrice() {

double totalPrice=0;

if(beroHeight==3){

totalPrice=5000;

}

else if (beroHeight==5){

totalPrice=8000;

}

else if (beroHeight==7){

totalPrice=10000;

}

setPrice(totalPrice);

}

}

public class WoodenBero extends Bero{

private String woodType;

public WoodenBero(String beroType, String beroColour, String woodType){

super(beroType,beroColour);

this.woodType=woodType;

}

public void setWoodType(String woodType){

this.woodType=woodType;

}

public String getWoodType(){

return woodType;

}

public void calculatePrice() {

double totalPrice=0;

if(woodType.equals("Ply Wood")){

totalPrice=15000;

}

else if (woodType.equals("Teak Wood")){

totalPrice=12000;

}

else if (woodType.equals("Engineered Wood")){

totalPrice=10000;

}

setPrice(totalPrice);

}

}

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

double TotalPrice=0;

Discount d=new Discount();

Scanner sc = new Scanner(System.in);

System.out.println("Enter Customer Name");

String cname=sc.nextLine();

System.out.println("Enter Phone Number");

long phno=Long.parseLong(sc.nextLine());

System.out.println("Enter address");

String ads=sc.nextLine();

System.out.println("Enter Bero Type");

String btype=sc.nextLine();

System.out.println("Enter Bero Colour");

String bColour=sc.nextLine();

if(btype.equals("Wooden Bero")){

System.out.println("Enter Wood Type");

String wType=sc.nextLine();

WoodenBero wb = new WoodenBero(btype,bColour,wType);

wb.calculatePrice();

TotalPrice= wb.getPrice()-d.calculateDiscount(wb);

System.out.printf("Amount needs to be paid Rs.%.2f",TotalPrice);

}

else if (btype.equals("Steel Bero")){

System.out.println("Enter Bero Height");

int height=Integer.parseInt(sc.nextLine());

SteelBero sb =new SteelBero(btype,bColour,height);

sb.calculatePrice();

TotalPrice=sb.getPrice()-d.calculateDiscount(sb);

System.out.printf("Amount needs to be paidRs.%.2f",TotalPrice);

}

}

}

public interface BonusPoints {

double calculateBonusPoints();

}

public class CustomerDetails implements BonusPoints, DoorDelivery{

private String customerName;

private String phoneNumber;

private String streetName;

private double billAmount;

private int distance;

public CustomerDetails(String customerName, String phoneNumber, String

streetName, double billAmount, int distance){

this.customerName=customerName;

this.phoneNumber=phoneNumber;

this.streetName=streetName;

this.billAmount=billAmount;

this.distance=distance;

}

public String getCustomerName(){

return customerName;

}

public void setCustomerName(String customerName){

this.customerName=customerName;

}

public String getPhoneNumber(){

return phoneNumber;

}

public void setPhoneNumber(String phoneNumber){

this.phoneNumber=phoneNumber;

}

public String getStreetName(){

return streetName;

}

public void setStreetName(String streetName){

this.streetName=streetName;

}

public double getBillAmount(){

return billAmount;

}

public void setBillAmount(double billAmount){

this.billAmount=billAmount;

}

public int getDistance(){

return distance;

}

public void setDistance(int distance){

this.distance=distance;

}

public double calculateBonusPoints() {

if(billAmount>=250){

return billAmount/10;

}

return 0;

}

public double deliveryCharge() {

if(distance>=25){

return distance\*8;

}

else if (distance >= 15 && distance <25){

return distance\*5;

}

return distance\*2;

}

}

public interface DoorDelivery {

double deliveryCharge();

}

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner (System.in);

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

String cname=sc.nextLine();

System.out.println("Enter the phone number");

String pno=sc.nextLine();

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

String streetname=sc.nextLine();

System.out.println("Enter the bill Amount");

double billAmount=Double.parseDouble(sc.nextLine());

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

int distance=Integer.parseInt(sc.nextLine());

CustomerDetails cd=new

CustomerDetails(cname,pno,streetname,billAmount,distance);

System.out.println("Customer name "+cd.getCustomerName());

System.out.println("phone number "+cd.getPhoneNumber());

System.out.println("Street name "+cd.getStreetName());

System.out.println("Bonus points "+cd.calculateBonusPoints());

System.out.println("Delivery charge "+cd.deliveryCharge());

}

}

public class DayScholar extends Student{

private int busNumber;

private float distance;

public DayScholar(int studentId, String studentName, String department,

String gender, String category, double collegeFee, int busNumber, float distance){

super(studentId,studentName,department,gender,category,collegeFee);

this.busNumber=busNumber;

this.distance=distance;

}

public int getBusNumber(){

return busNumber;

}

public void setBusNumber(int busNumber){

this.busNumber=busNumber;

}

public float getDistance(){

return distance;

}

public void setDistance(float distance){

this.distance=distance;

}

public double calculateTotalFee() {

int busFee=0;

if(distance>30 && distance <=40){

busFee=28000;

}

else if (distance>20 && distance<=30){

busFee=20000;

}

else if(distance >10 && distance <= 20){

busFee=12000;

}

else{

busFee=6000;

}

return (collegeFee+busFee);

}

}

public abstract class Student {

protected int studentId;

protected String studentName;

protected String department;

protected String gender;

protected String category;

protected double collegeFee;

public Student(int studentId, String studentName, String department, String

gender, String category, double collegeFee){

this.studentId=studentId;

this.studentName=studentName;

this.department=department;

this.gender=gender;

this.category=category;

this.collegeFee=collegeFee;

}

public int getStudentId(){

return studentId;

}

public void setStudentId(int studentId){

this.studentId=studentId;

}

public String getStudentName(){

return studentName;

}

public void setStudentName(String studentName){

this.studentName=studentName;

}

public String getDepartment(){

return department;

}

public void setDepartment(String department){

this.department=department;

}

public String getGender(){

return gender;

}

public void setGender(String gender){

this.gender=gender;

}

public String getCategory(){

return category;

}

public void setCategory(String category){

this.category=category;

}

public double getCollegeFee(){

return collegeFee;

}

public abstract double calculateTotalFee();

}

import java.util.Scanner;

public class UserInterface {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter Student Id");

int studentId= Integer.parseInt(sc.nextLine());

System.out.println("Enter Student name");

String name=sc.nextLine();

System.out.println("Enter Department name");

String deptName=sc.nextLine();

System.out.println("Enter gender");

String gender=sc.nextLine();

System.out.println("Enter category");

String category=sc.nextLine();

System.out.println("Enter College fee");

double collegeFee=Double.parseDouble(sc.nextLine());

if(category.equals("DayScholar")){

System.out.println("Enter Bus number");

int busNumber=Integer.parseInt(sc.nextLine());

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

int distance=Integer.parseInt(sc.nextLine());

DayScholar dayScholar=new DayScholar(studentId, name, deptName, gender,category, collegeFee, busNumber, distance);

System.out.println("Total College fee is"+dayScholar.calculateTotalFee());

}

else{

System.out.println("Enter the room number");

int roomNumber=Integer.parseInt(sc.nextLine());

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

char blockName=sc.nextLine().charAt(0);

System.out.println("Enter the room type");

String roomType=sc.nextLine();

Hosteller hosteller=new

Hosteller(studentId,name,deptName,gender,category,collegeFee,roomNumber,blockName,roomType);

System.out.println("Total College fee is"+hosteller.calculateTotalFee());

}

}

}

public class Hosteller extends Student{

private int roomNumber;

private char blockName;

private String roomType;

public Hosteller(int studenId,String studentName, String department, String

gender, String category, double collegeFee, int roomNumber, char blockName, String

roomType){

super(studenId,studentName,department,gender,category,collegeFee);

this.roomNumber=roomNumber;

this.blockName=blockName;

this.roomType=roomType;

}

public int getRoomNumber(){

return roomNumber;

}

public void setRoomNumber(int roomNumber){

this.roomNumber=roomNumber;

}

public char getBlockName(){

return blockName;

}

public void setBlockName(char blockName){

this.blockName=blockName;

}

public String getRoomType(){

return roomType;

}

public void setRoomType(String roomType){

this.roomType=roomType;

}

public double calculateTotalFee(){

int roomFee=0;

int hostelFee=0;

if(blockName=='A'){

hostelFee=60000;

if(roomType.equals("AC")){

roomFee=8000;

}

}

else if (blockName=='B'){

hostelFee=50000;

if(roomType.equals("AC")){

roomFee=5000;

}

}

else if (blockName=='C'){

hostelFee=40000;

if(roomType.equals("AC")){

roomFee=2500;

}

}

return collegeFee+hostelFee+roomFee;

}

}

import java.util.ArrayList;

public interface EmployeeAudit {

public ArrayList<String> fetchEmployeeDetails(double salary);

}

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class Main {

private static Map <String,Double> employeeMap = new

HashMap<String,Double>();

public Map<String, Double> getEmployeeMap() {

return employeeMap;

}

public void setEmployeeMap(Map<String, Double> employeeMap) {

this.employeeMap = employeeMap;

}

public void addEmployeeDetails(String employeeName, double salary)

{

employeeMap.put(employeeName,salary);

}

public static EmployeeAudit findEmployee()

{

ArrayList<String> name= new ArrayList<String>();

EmployeeAudit employeeAudit = (search)->{

for (Map.Entry<String,Double> i:employeeMap.entrySet() )

if(i.getValue()<=search){

name.add(i.getKey());

}

return name;

};

return employeeAudit;

}

public static void main(String[] args)

{

Main emp = new Main();

Scanner sc=new Scanner(System.in);

int choice=0;

do{

System.out.println("1. Add Employee details");

System.out.println("2. Find Employee details");

System.out.println("3. Exit");

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

choice=Integer.parseInt(sc.nextLine());

switch(choice){

case 1:

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

String name=sc.nextLine();

System.out.println("Enter the Employee Salary");

double salary=Double.parseDouble(sc.nextLine());

emp.addEmployeeDetails(name,salary);

break;

case 2:

System.out.println("Enter the salary to be searched");

double search=Double.parseDouble(sc.nextLine());

ArrayList<String>

nameList=findEmployee().fetchEmployeeDetails(search);

if(nameList.isEmpty()){

System.out.println("No employee found");

}

else{

System.out.println("Employee List");

for(String empName: nameList){

System.out.println(empName);

}

}

break;

default:

break;

}}while(choice!=3);

System.out.println("Let's complete the session");

}

}

public interface NumberCategory{

public boolean checkNumberCategory(int num1,int num2);

}

import java.util.\*;

public class NumberCategoryUtility{

static int findFactor(int n){

int i;

int sum=0;

for ( i=1; (i\*i)<n; i++){

if(n%i==0){

sum+=i;

}

}

if(i-(n/i)==1){

i--;

}

for (;i>1; i--){

if(n%i==0){

sum+=(n/i);

}

}

return sum;

}

public static boolean isPalindrome(int num){

String n=String.valueOf(num);

int i=0;

int j=n.length()-1;

while(i<j){

if(n.charAt(i)==n.charAt(j)){

i++;

j--;

continue;

}return false;

}return true;

}

public static NumberCategory checkAmicable(){

NumberCategory amicable=((number1,number2)->{

int n1=findFactor(number1);

int n2=findFactor(number2);

if(number1==n2 && number2==n1){

return true;

}

return false;

});

return amicable;

}

public static NumberCategory checkPalindrome(){

NumberCategory

palindrome=(((number1,number2)->isPalindrome(number1\*number2)));

return palindrome;

}

public static void main(String [] args)

{

Scanner sc=new Scanner(System.in);

int num1=Integer.parseInt(sc.nextLine());

int num2=Integer.parseInt(sc.nextLine());

boolean isAmicable=checkAmicable().checkNumberCategory(num1,num2);

boolean isPalindrome=checkPalindrome().checkNumberCategory(num1,num2);

if(isAmicable){

System.out.println("The numbers are amicable");

}

else{

System.out.println("The numbers are not amicable");

}

if(isPalindrome){

System.out.println("Product do produces a palindrome");

}

else{

System.out.println("Product does not produce a palindrome");

}

}

}

public interface CommissionInfo{

public double calculateCommissionAmount(Ticket ticketObj);

}

public class Ticket {

private long pnrNo;

private String passengerName;

private int seatNo;

private String classType;

private double ticketFare;

public long getPnrNo() {

return pnrNo;

}

public void setPnrNo(long pnrNo) {

this.pnrNo = pnrNo;

}

public String getPassengerName() {

return passengerName;

}

public void setPassengerName(String passengerName) {

this.passengerName = passengerName;

}

public int getSeatNo() {

return seatNo;

}

public void setSeatNo(int seatNo) {

this.seatNo = seatNo;

}

public String getClassType() {

return classType;

}

public void setClassType(String classType) {

this.classType = classType;

}

public double getTicketFare() {

return ticketFare;

}

public void setTicketFare(double ticketFare) {

this.ticketFare = ticketFare;

}

public Ticket(long pnrNo, String passengerName, int seatNo, String classType,

double ticketFare){

this.pnrNo=pnrNo;

this.passengerName=passengerName;

this.seatNo=seatNo;

this.classType=classType;

this.ticketFare=ticketFare;

}

}

import java.util.\*;

public class UserInterface{

public static CommissionInfo generateCommissionObtained(){

CommissionInfo commissionInfo=(ticketObj-> {

double commissionAmt=0;

if(ticketObj.getClassType().equalsIgnoreCase("sl")||ticketObj.getClassType().equalsIgnoreCase("2s")){

commissionAmt+=60;

}

else{

commissionAmt+=100;

}

return commissionAmt;

});

return commissionInfo;

}

public static void main(String [] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the no of passengers");

int count=Integer.parseInt(sc.nextLine());

Ticket[] tickets=new Ticket[count];

for (int i=1; i<=count; i++){

Ticket ticket;

System.out.printf("Details of Passenger %d:\n",i);

System.out.println("Enter the pnr no:");

long pnrNo = Long.parseLong(sc.nextLine());

System.out.println("Enter passenger name:");

String passengerName=sc.nextLine();

System.out.println("Enter seat no:");

int setSeatNo=Integer.parseInt(sc.nextLine());

System.out.println("Enter class type:");

String setClassType=sc.nextLine();

System.out.println("Enter ticket fare:");

double setTicketFare=Double.parseDouble(sc.nextLine());

tickets[i-1]=new

Ticket(pnrNo,passengerName,setSeatNo,setClassType,setTicketFare);

}

System.out.println("Commission Obtained");

double commission=0;

for (int i=0; i<tickets.length;i++){

commission+=generateCommissionObtained().calculateCommissionAmount(tickets[i]);

}

System.out.printf("Commission obtained per each person:Rs.%.2f",commission);

}

}

import java.util.Scanner;

public class Main

{

static int availableTickets;

public static void main(String[] arg)

{

Scanner sc=new Scanner(System.in);

int n,nt;

String name="";

int ticketid, price;

System.out.println("Enter movie name");

name=sc.next();

System.out.println("Enter no of bookings");

n=sc.nextInt();

System.out.println("Enter the available tickets");

availableTickets=sc.nextInt();

for (int i=0; i<n;i++ ){

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

ticketid=sc.nextInt();

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

price=sc.nextInt();

System.out.println("Enter the no of tickets");

nt=sc.nextInt();

Ticket o1=new Ticket();

o1.setTicketId(ticketid);

o1.setPrice(price);

o1.setAvailableTickets(availableTickets);

System.out.println("Available tickets: "+availableTickets);

if(availableTickets>=nt){

System.out.println("Total amount: "+o1.calculateTicketCost(nt));

availableTickets=availableTickets-nt;

if(availableTickets!=0){

System.out.println("Available ticket after booking:"+availableTickets);

}else{

System.out.println("House full");

break;

}

}

else{

System.out.println("Tickets are not available");

}

}

}

}

public class Ticket

{

private int ticketid;

private int price;

private static int availableTickets;

public void setTicketId(int ticketid){

this.ticketid=ticketid;

}

public int getTicketId(){

return ticketid;

}

public void setPrice(int price){

this.price=price;

}

public int getPrice(){

return price;

}

public void setAvailableTickets(int availableTickets){

this.availableTickets=availableTickets;

}

public int getAvailableTickets(){

return availableTickets;

}

public int calculateTicketCost(int nooftickets)

{

if(availableTickets>=nooftickets){

availableTickets=availableTickets - nooftickets;

return (nooftickets\*price);

}

else if (availableTickets==0){

return -1;

}

else if (availableTickets< nooftickets){

return -1;

}

return 0;

}

}

public class CustomerDetails

{

private String customerId;

private String customerName;

private long phoneNumber;

private String city;

private double unitConsumed;

private double costPerUnit;

public void setCustomerId(String customerId){

this.customerId=customerId;

}

public void setCustomerName(String customerName){

this.customerName=customerName;

}

public void setPhoneNumber(long phoneNumber){

this.phoneNumber=phoneNumber;

}

public void setCity(String city){

this.city=city;

}

public void setUnitConsumed(double unitConsumed){

this.unitConsumed=unitConsumed;

}

public void setCostPerUnit(double costPerUnit){

this.costPerUnit=costPerUnit;

}

public String getCustomerId(){

return customerId;

}

public String getCustomerName(){

return customerName;

}

public long getPhoneNumber(){

return phoneNumber;

}

public String getCity(){

return city;

}

public double getUnitConsumed(){

return unitConsumed;

}

public double getCostPerUnit(){

return costPerUnit;

}

public CustomerDetails(String customerId,String customerName,long phoneNumber,String city,double unitConsumed,double costPerUnit)

{

this.customerId=customerId;

this.customerName=customerName;

this.phoneNumber=phoneNumber;

this.city=city;

this.unitConsumed=unitConsumed;

this.costPerUnit=costPerUnit;

}

public double calculateElectricityBill(){

return(unitConsumed\*costPerUnit);

}

}

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

CustomerDetails c = new CustomerDetails("Sample","Sample",123456789,"Sample",12.00,13.00);

System.out.println("Enter Customer Id");

c.setCustomerId(sc.nextLine());

System.out.println("Enter Customer Name");

c.setCustomerName(sc.nextLine());

System.out.println("Enter Phone Number");

c.setPhoneNumber(sc.nextLong());

System.out.println("Enter City");

c.setCity(sc.next());

System.out.println("Enter Units Consumed");

c.setUnitConsumed(sc.nextDouble());

System.out.println("Enter Cost per Units");

c.setCostPerUnit(sc.nextDouble());

double amount=c.calculateElectricityBill();

System.out.printf("Amount to be paid is Rs.%.2f",amount);

}

}