package july;

import java.util.\*;

abstract class areaVolume{

abstract void calArea();

abstract void calVolume();

}

class sphere extends areaVolume{

double r;

sphere(double r){

this.r=r;

}

void calArea() {

double A=3.14\*r\*r;

System.*out*.println("Area "+A);

}

void calVolume(){

double v=3.14\*r\*r\*r;

System.*out*.println("Volume "+v);

}

}

class box extends areaVolume{

double h,b,l;

box(double h,double b,double l){

this.h=h;

this.b=b;

this.l=l;

}

void calArea() {

double a=l\*b;

System.*out*.println("Area "+a);

}

void calVolume() {

double v=l\*b\*h;

System.*out*.println("Volume "+v);

}

}

class cone extends areaVolume{

double r,h;

cone(double r,double h){

this.r=r;

this.h=h;

}

void calArea() {

double a=3.14\*r\*(Math.*sqrt*((r\*r)+(h\*h)))+3.14\*r\*r;

System.*out*.println("Area "+a);

}

void calVolume() {

double v=3.14\*r\*r\*h/3;

System.*out*.println("Volume "+v);

}

}

class cylinder extends areaVolume{

double r,h;

cylinder(double r,double h){

this.r=r;

this.h=h;

}

void calArea() {

double a=2\*3.14\*r\*h+2\*3.14\*r\*r;

System.*out*.println("Area "+a);

}

void calVolume() {

double v=3.14\*r\*r\*h;

System.*out*.println("Volume "+v);

}

}

abstract class staff{

protected int id;

protected String name;

staff(int id,String name){

this.id=id;

this.name=name;

}

abstract void display();

}

class officeStaff extends staff{

String department;

officeStaff(int id,String name,String department){

super(id,name);

this.department=department;

}

public void display() {

System.*out*.println("ID "+id+"\tName "+name+"\tDepartment "+department);

}

}

abstract class order{

int id;

String desc;

abstract void show();

}

class PurchaseOrder extends order{

Scanner sc=new Scanner(System.*in*);

String vendorName;

void accept() {

System.*out*.println("Enter id, description, vendor name");

id=sc.nextInt();

desc=sc.next();

vendorName=sc.next();

}

void show() {

System.*out*.println("ID "+id+"\nDescription "+desc+"\nVendor Name "+vendorName);

}

}

class SalesOrder extends order{

Scanner sc=new Scanner(System.*in*);

String customerName;

void accept() {

System.*out*.println("Enter id, description, customer name");

id=sc.nextInt();

desc=sc.next();

customerName=sc.next();

}

void show() {

System.*out*.println("ID "+id+"\nDescription "+desc+"\nCustomer Name "+customerName);

}

}

abstract class account{

String name;

int accNo;

abstract void deposit();

abstract void displayBalance();

abstract void withDraw();

}

class savAcc extends account{

Scanner sc=new Scanner(System.*in*);

double balance=200000,penalty=0,deposit,interest=0.10,withdraw,nb,t;

void deposit(){

System.*out*.println("Enter Amount to be deposited");

deposit=sc.nextDouble();

if(penalty!=0) {

t=(deposit-penalty);

if(t<=0) {

penalty-=deposit;

System.*out*.println("Penalty remaining "+penalty);

}

else {

balance+=t;

displayBalance();

}

}

else {

balance+=deposit;

displayBalance();

}

}

void displayBalance() {

System.*out*.println("Balance "+balance);

}

void interest() {

System.*out*.println("Current balance "+balance+"\tInterest on balance "+(interest\*balance));

balance+=(interest\*balance);

displayBalance();

}

void withDraw() {

System.*out*.println("Current balance "+balance);

System.*out*.println("Enter withdrawal amount ");

withdraw=sc.nextDouble();

nb=balance-withdraw;

if(nb<=0)

System.*out*.println("Less balance.\ntransaction cancelled");

else if(nb<=2000) {

System.*out*.println("Your account will be having a penalty.Still want to withdraw(y/n)");

char ch=sc.next().charAt(0);

if(ch=='y' || ch=='Y') {

balance=nb;

penalty+=2000;

System.*out*.println("Transaction successfull.\nCurrent balance "+balance+"\nPenalty "+penalty);

}

else

System.*out*.println("Transaction cancelled");

}

else {

balance=nb;

System.*out*.println("Transaction successfull.\nCurrent balance "+balance);

}

}

}

class currAcc extends account{

Scanner sc=new Scanner(System.*in*);

double balance=200000,penalty=0,deposit,interest=0.10,withdraw,nb,t;

void deposit(){

System.*out*.println("Enter Amount to be deposited");

deposit=sc.nextDouble();

if(penalty!=0) {

t=(deposit-penalty);

if(t<=0) {

penalty-=deposit;

System.*out*.println("Penalty remaining "+penalty);

}

else {

balance+=t;

displayBalance();

}

}

else {

balance+=deposit;

displayBalance();

}

}

void displayBalance() {

System.*out*.println("Balance "+balance);

}

void withDraw() {

System.*out*.println("Current balance "+balance);

System.*out*.println("Enter withdrawal amount ");

withdraw=sc.nextDouble();

nb=balance-withdraw;

if(nb<=0)

System.*out*.println("Less balance.\ntransaction cancelled");

else if(nb<=2000) {

System.*out*.println("Your account will be having a penalty.Still want to withdraw(y/n)");

char ch=sc.next().charAt(0);

if(ch=='y' || ch=='Y') {

balance=nb;

penalty+=2000;

System.*out*.println("Transaction successfull.\nCurrent balance "+balance+"\nPenalty "+penalty);

}

else

System.*out*.println("Transaction cancelled");

}

else {

balance=nb;

System.*out*.println("Transaction successfull.\nCurrent balance "+balance);

}

}

}

public class July\_26 {

public static void main(String[] args) {

Scanner sc=new Scanner(System.*in*);

/\*int ch;

double r,b,h,l;

do {

System.out.println("1.Area and Volume of Sphere\n2.Area and Volume of Box\n3.Area and Volume of Cone\n4.Area and Volume of Cylinder\n5.Exit");

ch=sc.nextInt();

switch(ch) {

case 1:

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

r=sc.nextDouble();

sphere s=new sphere(r);

s.calArea();

s.calVolume();

break;

case 2:

System.out.println("Enter height, breadth, length");

h=sc.nextDouble();

b=sc.nextDouble();

l=sc.nextDouble();

box b1=new box(h,b,l);

b1.calArea();

b1.calVolume();

break;

case 3:

System.out.println("Enter radius, height");

r=sc.nextDouble();

h=sc.nextDouble();

cone c=new cone(r,h);

c.calArea();

c.calVolume();

break;

case 4:

System.out.println("Enter radius, height");

r=sc.nextDouble();

h=sc.nextDouble();

cylinder c1=new cylinder(r,h);

c1.calArea();

c1.calVolume();

break;

case 5:

System.exit(1);

break;

default:

System.out.println("Enter valid option");

break;

}

}while(ch!=5);

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

int n=sc.nextInt();

officeStaff f[]=new officeStaff[n];

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

System.out.println("Enter id, name, department");

int id=sc.nextInt();

String name=sc.next();

String department=sc.next();

f[i]=new officeStaff(id,name,department);

}

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

f[i].display();

PurchaseOrder p[]=new PurchaseOrder[3];

System.out.println("Purchase Order");

for(int i=0;i<3;i++) {

p[i]=new PurchaseOrder();

p[i].accept();

}

for(int i=0;i<3;i++)

p[i].display();

SalesOrder s[]=new SalesOrder[3];

System.out.println("Sales Order");

for(int i=0;i<3;i++) {

s[i]=new SalesOrder();

s[i].accept();

}

for(int i=0;i<3;i++)

s[i].display();\*/

int ch,choice;

savAcc s=new savAcc();

currAcc c=new currAcc();

do {

System.*out*.println("1.Current Account\n2.Savings Account\n3.Exit");

ch=sc.nextInt();

switch(ch) {

case 1:

System.*out*.println("1.Deposit\n2.Check Balance\n3.Withdraw");

choice=sc.nextInt();

switch(choice) {

case 1:

c.deposit();

break;

case 2:

c.displayBalance();

break;

case 3:

c.withDraw();

break;

default:

System.*out*.println("Enter valid option");

break;

}

break;

case 2:

System.*out*.println("1.Deposit\n2.Check Balance\n3.Compute Interest\n4.Withdraw");

choice=sc.nextInt();

switch(choice) {

case 1:

s.deposit();

break;

case 2:

s.displayBalance();

break;

case 3:

s.interest();

break;

case 4:

s.withDraw();

break;

default:

System.*out*.println("Enter valid option");

break;

}

break;

case 3:

System.*exit*(1);

break;

default:

System.*out*.println("Enter valid option");

break;

}

}while(ch!=3);

}

}