

Lab Program 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class Acc that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

- a) Accept deposit from customers and update the balance
- b) Display the balance
- c) Compute and deposit interest
- d) Permit withdrawal and update balance
- e) Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
```

```
class account
```

```
{
```

```
    String name,
```

```
    int accno;
```

```
    String type;
```

```
    double balance;
```

```
    account(String name, int accno,  
            String type, double balance)
```

```
{
```

```
        this.name = name;
```

```
        this.accno = accno;
```

```
        this.type = type;
```

```
        this.balance = balance;
```

```
}
```

```
    void deposit(double amount)
```

```
{
```

```
        balance += amount;
```

```
}
```

```
    void withdraw(double amount)
```

```
{
```

~~```
 if((balance - amount) >= 0)
```~~

```
 balance -= amount;
```

```
}
```

```
else
```

```
{
```

```
 System.out.println("insufficient
balance, can't withdraw");
```

```
}
```

```
}
```

void display()

{

System.out.println("name:" + name,  
"accno:" + accno + "type:" + type +  
"balance:" + balance);

}

class savAcct extends account

{

private static double rate = 5;  
savAcct (String name, int accno,  
double balance)

{

super(name, accno, "savings",  
balance);

}

void interest()

{

balance += balance \* (rate) / 100;

System.out.println("balance:"  
+ balance);

}

class curAcct extends account

{

private double minBal = 500;

private double serviceCharger = 50;

{

curAcct (String name, int accno,  
double balance)

super(name, accno, "current",  
balance);

}

```
void checkmin()
```

```
{ if (balance < minBal)
```

```
System.out.println("balance is less than min balance, service charges imposed: " + serviceCharges);
balance -= serviceCharges;
```

```
System.out.println("Balance is: " + balance);
```

3  
3

```
class accountMain
```

```
{
```

```
public static void main (String a[])
```

```
Scanner s = new Scanner (System.in);
```

```
System.out.println ("enter the name");
```

```
String name = s.next();
```

```
System.out.println ("enter the type
(current/savings): ");
```

```
String type = s.next();
```

```
System.out.println ("enter the
account number: ");
```

```
int accno = s.nextInt();
```

```
System.out.println ("enter the
initial balance: ");
```

```
double balance = s.nextDouble();
```

```
int ch;
```

```
double amount1, amount2;
```

```
account acc = new account (name,
accno, type, balance);
```

```
savAcc sa = new savAcc(name,
accno, balance);
```

```
currAcc ca = new currAcc(name,
accno, balance);
```

```
while(true)
```

```
{
```

```
 if (acc.type.equals("saving"))
```

```
 System.out.println("Menu");
```

```
 \n1. deposit 2. withdraw 3. compute
 interest 4. display");
```

```
 System.out.println("enter the
 choice:");
```

```
 ch = s.nextInt();
```

```
 switch(ch)
```

```
 case 1: System.out.println("Enter the amount:");
```

```
 amount1 = s.nextInt();
```

```
 sa.deposit(amount1);
```

```
 break;
```

```
 case 2: System.out.println("Enter
 the amount:");
```

 ~~amount2 = s.nextInt();~~ ~~sa.withdraw(amount2);~~ ~~break;~~

```
 case 3: sa.interest();
```

```
 break;
```

 ~~case 4: System.exit(0);~~

```
 case 4: sa.display();
```

```
 break;
```

```
 case 5: System.exit(0);
```

default: System.out.println("invalid input");  
break;

}  
else  
{

System.out.println("menu [1. deposit  
2. withdraw 3. display"]");

System.out.print("enter the choice");  
ch = s.nextInt();  
switch(ch){

case 1: System.out.print("enter the  
amount");

amount1 = s.nextInt();  
ca.deposit(amount1);

break;

case 2: System.out.print("enter  
the amount");

amount2 = s.nextInt();  
ca.withdraw(amount2);

ca.checkmin();

break;

case 3: ca.display();

break;

case 4: System.exit(0);

default: System.out.println("invalid  
input");

break;

}  
}

}

Output :

Enter the name, type (current/savings), account number, initial balance,

Sakshi

savings

321

60000

menu

1. Deposit
2. Withdraw
3. Compute interest
4. Display

Enter the choice :

1

Enter the amount :

500

Menu

1. Deposit
2. Withdraw
3. Compute interest
4. Display

2

Enter the amount

500

Menu

1. Deposit
2. Withdraw
3. Compute interest
4. Display

3.

Balance : 57006.0

Menu

1. Deposit
2. Withdraw
3. Compute interest
4. Display

Enter the choice

4.

Name: Sakshi

Acc no: 3210

Type: Savings

Balance: 59006.0

Enter the name, type (current/saving),  
acc no, initial balance

Sakshi

Current

101

50000

Menu

1. Deposit
2. Withdraw
3. Display

Enter the choice

1

Enter the amount

5000

Menu

1. Deposit
2. Withdraw
3. Display

Enter the choice

2

~~Enter the amount~~

500

Menu

1. Deposit
2. Withdraw
3. Display

Enter the choice

3

Name : Sakshi

Acc no: 101

Type : current

Balance : 504500.0

Sakshi - B.R

IBM02CS033

SS  
16/1/2024

101

00002