

~~Lab programs~~
9/03/21

- 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 the level, a service charge is imposed.
- Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-act and Sav-act to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:
 - a) Accept deposit from customer and update the balance.
 - b) Display the balance.
 - c) Compute and deposit interest.
 - d) Permit withdrawal and update the balance.
- 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)

{ constructor statement
(initialization) this.name = name; this.accno = accno;

(initialization) this.type = type; this.balance = balance;

void deposit (double amount)

{ (deposit) this.balance += amount;
 this.balance + amount = balance;

void withdraw (double amount)

{ if ((balance - amount) >= 0)

{ (balance - amount) = balance;
 balance = amount;

(initialization) balance = amount;

else

(withdrawal) 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 is: " + balance);
}

class CurrAcct extends account {
    private double minBal = 500;
    private double serviceCharges = 50;
    CurrAcct (String name, int accno, double balance)
        Super (name, accno, "current", balance);

    void checkmin()
    {
        if (balance < minBal)
            System.out.println ("balance is less than min
" + minBal + " imposed: " + serviceCharges);
        balance -= serviceCharges;
        System.out.println ("balance is: " + balance);
    }
}

```

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

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

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

```
double amount1, amount2;
```

```
account acc = new
```

```
account ( name, acno, type, balance );
```

```
SavAcct sa = new
```

```
SavAcct ( name, acno, balance );
```

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

```
newAcc ( name, acno, balance );
```

```
Vehicle ( E )
```

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

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

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

```
1. deposit 2. withdraw 3. compute interest
```

```
( choice )
```

```
if ( choice == 1 )
```

```
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 ();

(case 4: sa.display ();

(case 5: System.exit (0);

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

(break);

exist = 100 thousand

(choice, exist, min) thousand
else exist = 10 thousand

(initial, min, max) thousand

System.out.println ("\n1. deposit

(1. deposit, 2. withdraw, 3. display");

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

ch = s.nextInt ();

((choice == 1) {

switch (ch) {

((choice == 1) {

(case 1: System.out.println ("Enter the amount!");
amount1 = s.nextInt ();

sa.deposit (amount1);

(break);

((choice == 2) {

```
case 2: System.out.println("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/Saving(), account number, initial balance;

Sakshi

~~savings (private firms) etc~~

123

50000

Menu

~~1. Deposit 2. Withdraw 3. Compute Interest
4. Display~~

Enter the choice :

1.

Enter the amount:

5000

Menu

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

2 : (1) Enter > = 5000

Enter the amount in LHS > 5000

5000 : (2) Enter > 5000

Menu:

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

3 : (3) Enter > 5000

Balance : 57225.00

Menu:

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

Enter the choice:

4.

Name: Sakshi

Accno: 123

Type: Savings

Balance: 57225.60

Enter the name, type (current/saving), account number

Initial balance,

Sakshi

current

101

500000

Menu

1. Deposit 2. Withdraw 3. display

Enter the choice

1.

Enter the amount:

5000

20/11/2022

1. Menu 2. Withdrawal 3. Deposit

1. Deposit 2. Withdrawals 3. display

Enter the choice: 1

2. withdrawal amount from account

Enter the amount: amount of AP Bank

500

amount of withdrawal

amount of withdrawal

1. Deposit 2. Withdrawals 3. display

Enter the choice: 3

3. display

Name: Sakshi

Acno: 101

Type: current

Balance: 504500.0

name: SHRI. SAKSHI SHETTY
IBN22CS234.

(1) points card = 0.20 points lost after

(2) points card = every points lost after

and 1.9 lost after

2. (selected bank) withdraw amount

(("selected") amount) amount = 22 amount

("(" withdraw " ") withdraw amount)

(((" withdraw ") withdraw amount))

(((" withdraw ") withdraw amount))