PROGRAM-5

19/01/2024.

```
import java.util.Scanner;

class Account {
    String customerName;
    long accno;
    String accountType;
    double balance;
    public Account (String customerName, long accno,
                    String accountType) {
    {
        this.customerName = customerName;
        this.accno = accno;
        this.accountType = accountType;
        this.balance = 0.0;
    }
    public void displayBalance() {
        System.out.print("Account number " + accno);
        System.out.println(" Customer name " + CustomerName);
    System.out.println("Account Type :" + accountType);
    System.out.println("Balance : $" + balance);
    }
}
class CurrentAcc extends Account {
double minBalance;
double serviceCharge;
public CurAcct (String customerName, long accno)
{
    super(CustomerName, accno, "Current");
    this.minBalance = 500.0
```

```
this.serviceCharge = 50.0;
Set service charge
}
public void withdraw (double amount){
    if (balance - amount >= minBalance)
    {
        balance -= amount;
    System.out.println ("Withdrawal Successful. Current
                balance : " + balance);
    }
    else {
        System.out.println ("Insufficient Balance,
    withdrawal unsuccessful");
    }
}
Public imposeServiceCharge () {
    if (Balance < minBalance) {
        Balance -= Service Charge();
    System.out.println ("Service charge imposed } Current
        balance " + balance);
    }
}
}
Public static void demo
Class SavingsAcc extends Account {
        double interestRate;

Public SavingsAcc (String customerName, long accNo)
{
    Super (CustomerName, accNo, "Savings");
        this.interestRate = 0.05;
}
```

```
public void depositInterest() {
    double interest = balance * interestRate;
    balance += interest;
System.out.println("Interest deposited. Current
        Balance : $" + balance);
}
public void CompoundInterest (double initial
 Amount * Math.pow((1 + interestRate), term) -
    initialAmount;
  balance += compoundInterest;
System.out.println("Compound Interest deposited.
   Current Balance : Rs." + balance);
  }
}
public class Bank {
    public static void main (String[] args){
        Scanner scanner = new Scanner(System.in);
System.out.println("Choose account type :");
System.out.println("1. Current");
System.out.println("2. Savings");
System.out.println("3. Enter choice (1 or 2) :");
  int choice = scanner.nextInt();
System.out.print("Enter Customer name : ");
String customerName = scanner.next();
System.out.println("Enter account number");
long accno = scanner.nextLong();
    if (choice == 1) {
        CurAcct curAccount = new
            CurAcct(CustomerName, accno);
System.out.println("Enter initial balance : $");
  double initialBalance = scanner.nextDouble();
   curAccount.balance = initialBalance;
```

```
System.out.println("Enter withdrawal amount : $ ");
double withdrawalAmount = Scanner.nextDouble();
curAccount.withdrawal(withdrawalAmount);
curAccount.imposeServiceCharge();
curAccount.displayBalance();
}
else if(choice==2){
    SavAcct savAccount = new SavAcct(customerName,
                                     accNo);
System.out.println("Enter initial balance : $");
double initialBalance = scanner.nextDouble();
SavAccount.balance = initialBalance;
System.out.print("Enter withdrawal amount : $ ");
double withdrawalAmount = scanner.nextDouble();
SavAccount.balance -= withdrawalAmount;
System.out.println("Withdrawal successful. Current
Balance : $" + savAccount.balance);
System.out.println("Enter the interest rate : ");
double interestRate = Scanner.nextDouble();
savAccount.interestRate = interestRate;
savAccount.displayBalance();
    System.out.print("Enter term (in years) for
    compound interest calculation : ");
    int term = Scanner.nextInt();
savAccount.compoundInterest(initialBalance, term);
savAccount.displayBalance();
}
else{
    System.out.println("Invalid choice");
}
}
}
```

Algorithm

Step 1: Start

Step 2: Create class account, initialize constructor accordingly

Step 3: Create a method display balance with the class account

Step 4: Extend class Account to currentacc

Step 5: Create method withdraw with current account

Step 6: Create method imposeservicecharge.

Step 7: Create class savings which extends Account

Step 8: Create method deposit Interest inside the class

Step 9: Create method compound interest for calculating compound interest

Step 10: Create class bank

Step 11: Take inputs from the user like name, account number, amount, balance, withdraw amount and execute the methods

Step 12: Stop

02.02.24