Assignment 2

|  |
| --- |
| **Class: BankAccount** |
| - balance : double  - num\_Of\_Deposit : int  - num\_Of\_Withdrawals : int  - annual\_Int\_Rate : double  - monthly\_Ser\_Charge : double |
| + BankAccount()  + BankAccount(balance : double, annual\_Int\_Rate : double, monthly\_Ser\_Charge : double)  + getBalance : double  + getNum\_Of\_Deposit : int  + getNum\_Of\_Withdrawals : int  + getAnnual\_Int\_Rate : double  + getMonthly\_Ser\_Charge : double  + setBalance(double balance) : void  + setNum\_Of\_Deposit(int num\_Of\_Deposit) : void  + setNum\_Of\_Withdrawals(int num\_Of\_Withdrawals) : void  + setAnnual\_Int\_Rate(double annual\_Int\_Rate) : void  + setMonthly\_Ser\_Charge(double monthly\_Ser\_Charge) : void  + deposit(double amount\_Of\_Deposit) : void  + withdraw(double amount\_Of\_Withdraw) : void  + calcInterest() : double  + monthlyProcess() : void |

|  |
| --- |
| **Class : SavingsAccount** |
| - status : Boolean |
| + SavingsAccount(balance : double, monthly\_Ser\_Charge : double, annual\_Int\_Rate : double)  + deposit(double amount\_Of\_Deposit) : void  + withdraw(double amount\_Of\_Withdraw) : void |

**Program BankAccount :**

**package Inheritance;**

**/\*\***

**\* @author Jay Patel**

**\* @version 1.0**

**\***

**\*/**

**public class BankAccount {**

**//Variables**

**private double balance;**

**private int num\_Of\_Deposit;**

**private int num\_Of\_Withdrawals;**

**private double annual\_Int\_Rate;**

**private double monthly\_Ser\_Charge;**

**//Default Constructor**

**BankAccount(){**

**balance = 0.0;**

**num\_Of\_Deposit = 0;**

**num\_Of\_Withdrawals = 0;**

**annual\_Int\_Rate = 0.0;**

**monthly\_Ser\_Charge = 0.0;**

**}**

**//Parameterized Constructor**

**BankAccount(double balance,double monthly\_Ser\_Charge,**

**double annual\_Int\_Rate){**

**this.balance = balance;**

**this.monthly\_Ser\_Charge = monthly\_Ser\_Charge;**

**this.annual\_Int\_Rate = annual\_Int\_Rate;**

**}**

**//accessor methods**

**/\*\***

**\***

**\* @return Balance- double value**

**\*/**

**public double getBalance()**

**{**

**return balance;**

**}**

**/\*\***

**\***

**\* @return Number Of Deposit - int value**

**\*/**

**public int getNum\_Of\_Deposit()**

**{**

**return num\_Of\_Deposit;**

**}**

**/\*\***

**\***

**\* @return Number of Withdrawals - int value**

**\*/**

**public int getNum\_Of\_Withdrawals() {**

**return num\_Of\_Withdrawals;**

**}**

**/\*\***

**\***

**\* @return Annual Interest Rate - double value**

**\*/**

**public double getAnnual\_Int\_Rate()**

**{**

**return annual\_Int\_Rate;**

**}**

**/\*\***

**\***

**\* @return Monthly Service Charge - double value**

**\*/**

**public double getMonthly\_Ser\_Charge()**

**{**

**return monthly\_Ser\_Charge;**

**}**

**// mutator methods**

**/\*\***

**\***

**\* @param balance - double value**

**\*/**

**public void setBalance(double balance)**

**{**

**this.balance = balance;**

**}**

**/\*\***

**\***

**\* @param num\_Of\_Deposit - int value**

**\*/**

**public void setNum\_Of\_Deposit(int num\_Of\_Deposit)**

**{**

**this.num\_Of\_Deposit = num\_Of\_Deposit;**

**}**

**/\*\***

**\***

**\* @param num\_Of\_Withdrawals - int value**

**\*/**

**public void setNum\_Of\_Withdrawals(int num\_Of\_Withdrawals)**

**{**

**this.num\_Of\_Withdrawals = num\_Of\_Withdrawals;**

**}**

**/\*\***

**\***

**\* @param annual\_Int\_Rate - double value**

**\*/**

**public void setAnnual\_Int\_Rate(double annual\_Int\_Rate) {**

**this.annual\_Int\_Rate = annual\_Int\_Rate;**

**}**

**/\*\***

**\***

**\* @param monthly\_Ser\_Charge - double value**

**\*/**

**public void setMonthly\_Ser\_Charge(double monthly\_Ser\_Charge) {**

**this.monthly\_Ser\_Charge = monthly\_Ser\_Charge;**

**}**

**// methods**

**/\*\***

**\***

**\* @param amount\_Of\_Deposit -double value**

**\*/**

**public void deposit(double amount\_Of\_Deposit){**

**balance += amount\_Of\_Deposit;**

**num\_Of\_Deposit++;**

**System.out.println("Number Of Deposits : "+getNum\_Of\_Deposit() +**

**"; Number Of Withdraws : "+getNum\_Of\_Withdrawals()+"; Balance : " + getBalance());**

**}**

**/\*\***

**\***

**\* @param amount\_Of\_Withdraw - double value**

**\*/**

**public void withdraw(double amount\_Of\_Withdraw){**

**if (balance < amount\_Of\_Withdraw){**

**System.out.println("Sorry! Your Balance is too low for this withdrawal!");**

**}else**

**{**

**balance -= amount\_Of\_Withdraw;**

**num\_Of\_Withdrawals++;**

**if (balance<50)**

**System.out.println("Your Account balance is to low.\n" +**

**"And if you will not deposit a few money within few months then " +**

**"your account will be deactivated within 45 days.");**

**}**

**System.out.println("Number Of Deposits : " +getNum\_Of\_Deposit()+**

**"; Number Of Withdraws : " +getNum\_Of\_Withdrawals()+**

**"; Balance : " +getBalance());**

**}**

**public double calcInterest(){**

**double monthly\_Int\_Rate,monthly\_Interest;**

**monthly\_Int\_Rate = annual\_Int\_Rate/12;**

**monthly\_Interest = balance\*monthly\_Int\_Rate;**

**balance += monthly\_Interest;**

**return balance+monthly\_Interest;**

**}**

**public void monthlyProcess(){**

**balance -= monthly\_Ser\_Charge;**

**calcInterest();**

**num\_Of\_Withdrawals=0;**

**num\_Of\_Deposit=0;**

**monthly\_Ser\_Charge=0.0;**

**}**

**}**

**Program SavingsAccount :**

**package Inheritance;**

**/\*\***

**\* @author Jay Patel**

**\* @version 1.0**

**\***

**\*/**

**public class SavingsAccount extends BankAccount {**

**private Boolean status;**

**SavingsAccount(double balance,double monthly\_Ser\_Charge,**

**double annual\_Int\_Rate){**

**super(balance, monthly\_Ser\_Charge, annual\_Int\_Rate);**

**if(getBalance()<50){**

**status = false;**

**}else {**

**status = true;**

**}**

**}**

**//methods**

**public void deposit(double amount\_Of\_Deposit){**

**super.deposit(amount\_Of\_Deposit);**

**if (status==false && getBalance() > 50){**

**status=true;**

**}**

**}**

**public void withdraw(double amount\_Of\_Withdraw){**

**if (status==false){**

**System.out.println("Sorry!! No withdrawals allowed As Your Account is deactivated. ");**

**System.out.println("Number Of Deposits : " +getNum\_Of\_Deposit()+**

**"; Number Of Withdraws : " +getNum\_Of\_Withdrawals()+**

**"; Balance : " +getBalance());**

**}else{**

**super.withdraw(amount\_Of\_Withdraw);**

**if (getBalance()<50){**

**status=false;**

**}**

**}**

**}**

**}**

**Program AccountDemo :**

**package Inheritance;**

**import java.util.Scanner;**

**public class AccountDemo {**

**public static void main(String[] args) {**

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

**double balance, annual\_Int\_Rate,monthly\_Ser\_Charge,**

**amount\_Of\_Deposit,amount\_Of\_Withdraw;**

**System.out.println("Testing Bank Account Class");**

**System.out.println("Enter Your Balance");**

**balance = input.nextDouble();**

**System.out.println("Enter Interest Rate");**

**annual\_Int\_Rate = input.nextDouble();**

**System.out.println("Enter Monthly Charges");**

**monthly\_Ser\_Charge = input.nextDouble();**

**System.out.println("Enter Amount Of Deposit You Want To Deposit");**

**amount\_Of\_Deposit = input.nextDouble();**

**BankAccount ba = new BankAccount(balance,**

**monthly\_Ser\_Charge,annual\_Int\_Rate);**

**System.out.println("After Calling Deposit");**

**ba.deposit(amount\_Of\_Deposit);**

**System.out.println("Enter Amount Of Withdraw you Want To Withdraw");**

**amount\_Of\_Withdraw = input.nextDouble();**

**System.out.println("After Calling Withdraw");**

**ba.withdraw(amount\_Of\_Withdraw);**

**System.out.println("\nTesting Savings Account Class");**

**System.out.println("Enter Your Balance");**

**balance = input.nextDouble();**

**System.out.println("Enter Interest Rate");**

**annual\_Int\_Rate = input.nextDouble();**

**System.out.println("Enter Monthly Charges");**

**monthly\_Ser\_Charge = input.nextDouble();**

**System.out.println("Enter Amount Of Deposit You Want To Deposit");**

**amount\_Of\_Deposit = input.nextDouble();**

**SavingsAccount sa = new SavingsAccount(balance,monthly\_Ser\_Charge,annual\_Int\_Rate);**

**System.out.println("After Calling Deposit");**

**sa.deposit(amount\_Of\_Deposit);**

**System.out.println("Enter Amount Of Withdraw you Want To Withdraw");**

**amount\_Of\_Withdraw = input.nextDouble();**

**System.out.println("After Calling Withdraw");**

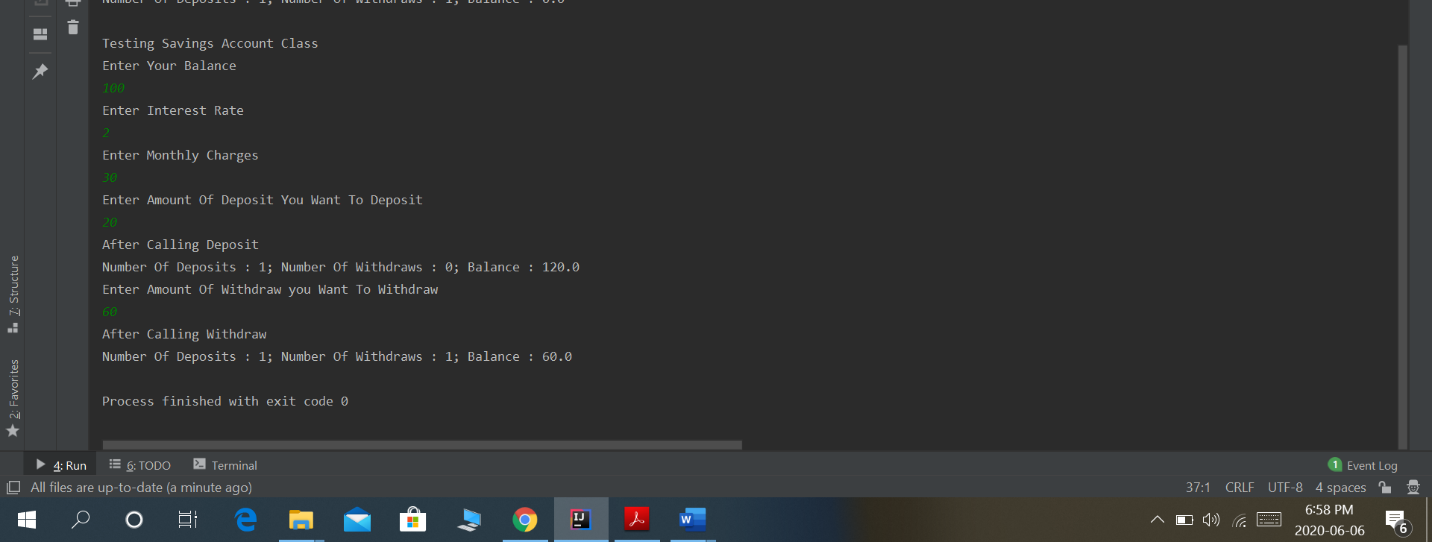
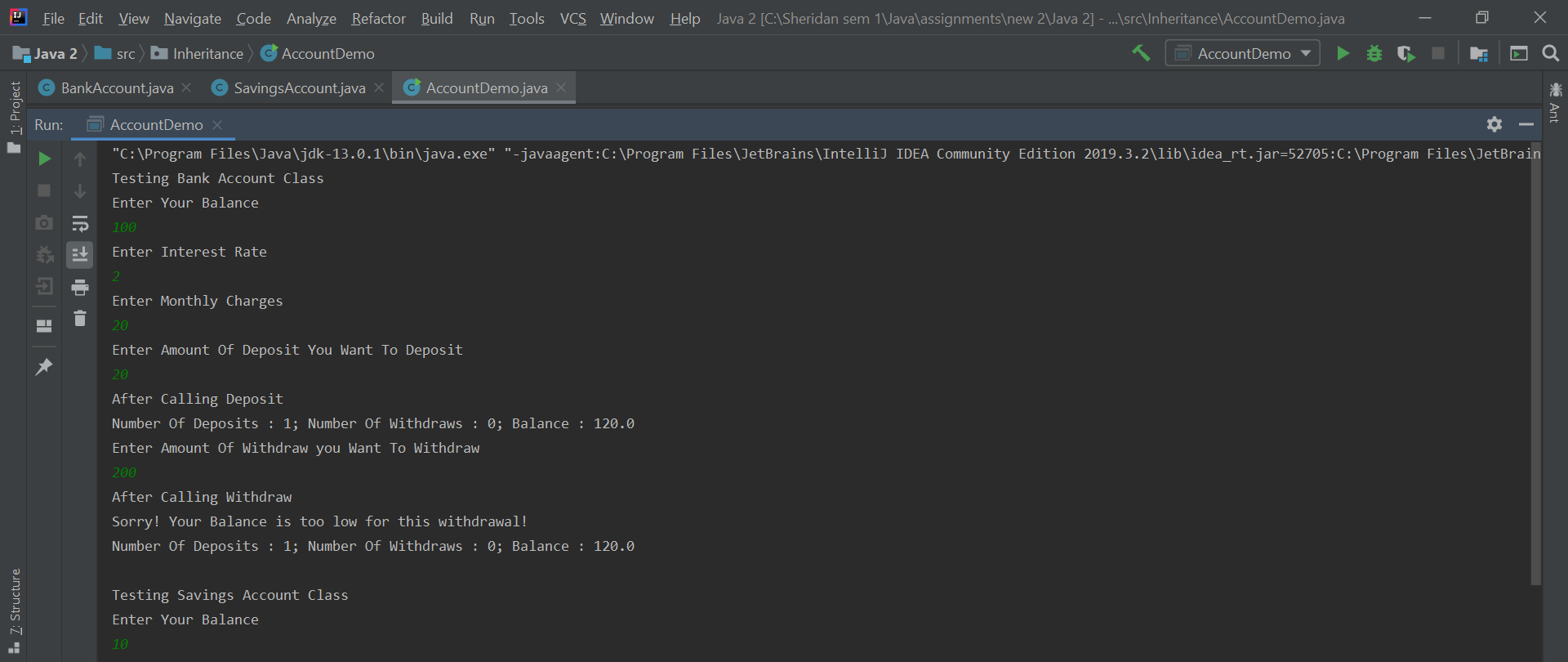
**sa.withdraw(amount\_Of\_Withdraw);**

**}**

**}**

**Screenshots :**

**Sample Run1:**



**Sample Run2 :**

