Aayush Maheshwari

2047152

Design a interface Bank with the following members

Member Functions: CreateAccount( ) - > Creating new Account for the particular bank

Deposit (double amount), Withdraw (double amount),Getbalance( ),GetAnnualInterest ( ) -> Returns Annual Interest Rate

Design a three class called SBI, HDFC and ICICI which implements from BANK interface (Assume that each bank have different

interest rate) with the following members

Data Members: Bankid, Branch, Location

Member function: SetMonthlyInterest ( ) ->Set Different monthly interest rate for each bank

Design a class Person inherits from above three class (SBI or HDFC or ICICI) with the following members

Data Members: PersonId, Name, Account Number, TypeofAccount, Email

Member functions: AccountDetails () -> Display the following details for the each account holders

PersonId, Name, Bankid, Bankname, Balance, AnnualInterest (Note that AnnualInterestRate is a percentage,e.g.,

like 4.5%. You need to divide it by 100)

The program should have the following features

1. Create user defined exception called Insufficientbalance whenever user withdrawing amount more than his/her balance.

2. Convert account holders name from lowercase to uppercase except first character (Note: First character should be capital letter)

3. Display the person name who is having maximum amount in the particular bank (use math class)

4. Use of exception wherever possible

Note: You can add additional functions in the classes wherever required

\*/\

CODE:

import java.util.Scanner;

import java.util.\*;

interface Bank {

   // void create\_account();

    void deposit(double amount);

    void withdraw(double amount) throws InsuffiecientBalanceException;

    double getBalance();

    double getAnnualInterest();

}

class SBI implements Bank {

    private int bankId;

    private String branch;

    private String location;

    private double interestRate;

    private double totalAmount;

    public SBI() {

        interestRate = 4.5;

    }

    public void setBankID(int id) {

        bankId = id;

    }

    public int getBankId() {

        return bankId;

    }

    public String getBankName() {

        return "SBI";

    }

    public void deposit(double amount) {

        totalAmount += amount;

    }

    public void withdraw(double amount) throws InsuffiecientBalanceException {

        if (amount > totalAmount) {

            throw new InsuffiecientBalanceException("Insufficient Balance!!");

        } else {

            totalAmount -= amount;

        }

    }

    public double getBalance() {

        return totalAmount;

    }

    public double getAnnualInterest() {

        return interestRate;

    }

}

class HDFC implements Bank {

    private int bankId;

    private String branch;

    private String location;

    private double interestRate;

    private double totalAmount;

    public HDFC() {

        interestRate = 5.5;

    }

    public void setBankID(int id) {

        bankId = id;

    }

    public int getBankId() {

        return bankId;

    }

    public String getBankName() {

        return "HDFC";

    }

    public void deposit(double amount) {

        totalAmount += amount;

    }

    public void withdraw(double amount) throws InsuffiecientBalanceException {

        if (amount > totalAmount) {

            throw new InsuffiecientBalanceException("Insufficient Balance!!");

        } else {

            totalAmount -= amount;

        }

    }

    public double getBalance() {

        return totalAmount;

    }

    public double getAnnualInterest() {

        return interestRate;

    }

}

class ICICI implements Bank {

    private int bankId;

    private String branch;

    private String location;

    private double interestRate;

    private double totalAmount;

    public ICICI() {

        interestRate = 6.5;

    }

    public void setBankID(int id) {

        bankId = id;

    }

    public int getBankId() {

        return bankId;

    }

    public String getBankName() {

        return "ICICI";

    }

    public void deposit(double amount) {

        totalAmount += amount;

    }

    public void withdraw(double amount) throws InsuffiecientBalanceException {

        if (amount > totalAmount) {

            throw new InsuffiecientBalanceException("Insufficient Balance!!");

        } else {

            totalAmount -= amount;

        }

    }

    public double getBalance() {

        return totalAmount;

    }

    public double getAnnualInterest() {

        return interestRate;

    }

}

class InsuffiecientBalanceException extends Exception {

    String message;

    public InsuffiecientBalanceException(String message) {

        super(message);

        this.message = message;

    }

    public String getMessage() {

        return message;

    }

}

class Person extends SBI {

    private int personID;

    private String name;

    private int accountNumber;

    private String typeOfAccount;

    private String email;

    private String bankName;

    Person(int pId, String pName, int accountNo, String accountType, String pEmail, String pBankName) {

        personID = pId;

        name = pName;

        name = name.toUpperCase();

        accountNumber = accountNo;

        typeOfAccount = accountType;

        email = pEmail;

        bankName = pBankName;

        if (bankName == "SBI") {

            setBankID(1);

        } else if(bankName == "HDFC") {

            setBankID(2);

        }else {

            setBankID(3);

        }

    }

    public int getPersonId() {

        return personID;

    }

    public String getName() {

        return name;

    }

    public int getAccountNumber() {

        return accountNumber;

    }

    public void accountDetails() {

        System.out.println("Person id : " + getPersonId());

        System.out.println("Person name : " + getName());

        //System.out.println("Bank id : " + getBankId());

        System.out.println("Bank name : " + getBankName());

        System.out.println("Balance : " + getBalance());

        System.out.println("Annual Interest : " + getAnnualInterest());

    }

}

public class Classwork\_Bank {

    public static void main(String[] args) {

        Scanner scn = new Scanner(System.in);

        System.out.print("Enter number of records you want to enter : ");

        int n = scn.nextInt();

        Person personWithMaxAmount = null;

        double maxAmount = 0;

        for (int i=1; i<=n; i++) {

            System.out.println("Enter Person id : ");

            int pId = scn.nextInt();

            System.out.println("Enter Person name : ");

            scn.nextLine();

            String pName = scn.nextLine();

            System.out.println("Enter Person email : ");

            String email = scn.nextLine();

            System.out.println("Enter your type of account : ");

            String typeOfAccount = scn.nextLine();

            System.out.println("Enter Person account number : ");

            int accountNumber = scn.nextInt();

            System.out.println("Enter your bank name : ");

            String bankName = scn.nextLine();

            Person p = new Person(pId, pName, accountNumber, typeOfAccount, email, bankName);

            int ch;

            do {

                System.out.println("Main Menu\n1.Deposit\n2.Withdraw\n3.Check Balance\n4.Print Account Details\n5.Exit");

                System.out.println("Your choice : ");

                ch = scn.nextInt();

                switch(ch) {

                    case 1:

                        System.out.println("Enter amount to deposit : ");

                        double amount = scn.nextDouble();

                        p.deposit(amount);

                        break;

                    case 2:

                        System.out.println("Enter amount to withdraw : ");

                        double money = scn.nextDouble();

                        try {

                           p.withdraw(money);

                        } catch(InsuffiecientBalanceException e) {

                            System.out.println(e.getMessage());

                        }

                        break;

                    case 3:

                        System.out.println("Balance : " + p.getBalance());

                        break;

                    case 4:

                        p.accountDetails();

                        break;

                    case 5:

                         System.out.println("Good bye...");

                        break;

                }

                if (p.getBalance() > maxAmount) {

                    maxAmount = p.getBalance();

                    personWithMaxAmount = p;

                }

            } while (ch != 5);

        }

        System.out.println("Person with maximum amount : " + personWithMaxAmount.getName());

    }

}







