import java.util.Scanner;

class Account {

    protected String customerName;

    protected String accountNumber;

    protected double balance;

    public Account(String customerName, String accountNumber, double initialBalance) {

        this.customerName = customerName;

        this.accountNumber = accountNumber;

        this.balance = initialBalance;

    }

    public void deposit(double amount) {

        balance += amount;

        System.out.println("Deposited: " + amount);

    }

    public void displayBalance() {

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

    }

    public void withdraw(double amount) {

        if (amount > balance) {

            System.out.println("Insufficient balance!");

        } else {

            balance -= amount;

            System.out.println("Withdrew: " + amount);

        }

    }

}

class SavAcct extends Account {

    private double interestRate;

    public SavAcct(String customerName, String accountNumber, double initialBalance, double interestRate) {

        super(customerName, accountNumber, initialBalance);

        this.interestRate = interestRate;

    }

    public void computeAndDepositInterest(int years) {

        double interest = balance \* Math.pow((1 + interestRate / 100), years) - balance;

        deposit(interest);

        System.out.println("Interest for " + years + " years deposited: " + interest);

    }

}

class CurAcct extends Account {

    private double minimumBalance;

    private double serviceCharge;

    public CurAcct(String customerName, String accountNumber, double initialBalance, double minimumBalance, double serviceCharge) {

        super(customerName, accountNumber, initialBalance);

        this.minimumBalance = minimumBalance;

        this.serviceCharge = serviceCharge;

    }

    @Override

    public void withdraw(double amount) {

        if (amount > balance) {

            System.out.println("Insufficient balance!");

        } else {

            balance -= amount;

            System.out.println("Withdrew: " + amount);

            checkMinimumBalance();

        }

    }

    private void checkMinimumBalance() {

        if (balance < minimumBalance) {

            balance -= serviceCharge;

            System.out.println("Minimum balance not maintained. Service charge of " + serviceCharge + " applied.");

        }

    }

}

public class Bank {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Account account = null;

        System.out.println("Welcome to the Bank!");

        System.out.print("Enter customer name: ");

        String name = sc.nextLine();

        System.out.print("Enter account number: ");

        String accountNumber = sc.nextLine();

        System.out.print("Choose account type (1 for Savings, 2 for Current): ");

        int accountType = sc.nextInt();

        if (accountType == 1) {

            System.out.print("Enter initial balance: ");

            double initialBalance = sc.nextDouble();

            System.out.print("Enter interest rate: ");

            double interestRate = sc.nextDouble();

            account = new SavAcct(name, accountNumber, initialBalance, interestRate);

        } else if (accountType == 2) {

            System.out.print("Enter initial balance: ");

            double initialBalance = sc.nextDouble();

            System.out.print("Enter minimum balance: ");

            double minimumBalance = sc.nextDouble();

            System.out.print("Enter service charge: ");

            double serviceCharge = sc.nextDouble();

            account = new CurAcct(name, accountNumber, initialBalance, minimumBalance, serviceCharge);

        } else {

            System.out.println("Invalid account type.");

            return;

        }

        int choice;

        do {

            System.out.println("\nMenu:");

            System.out.println("1. Deposit");

            System.out.println("2. Display Balance");

            System.out.println("3. Withdraw");

            System.out.println("4. Compute and Deposit Interest (Savings only)");

            System.out.println("5. Exit");

            System.out.print("Enter your choice: ");

            choice = sc.nextInt();

            switch (choice) {

                case 1:

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

                    double depositAmount = sc.nextDouble();

        case 2: account.displayBalance();

             break;

        case 3: System.out.print("Enter amount to withdraw: ");

            double withdrawAmount = sc.nextDouble();

            account.withdraw(withdrawAmount);

            break;

        case 4: if (account instanceof SavAcct)

                        { System.out.print("Enter number of years for interest calculation: ");

                        int years = sc.nextInt();

                        ((SavAcct) account).computeAndDepositInterest(years);

                        } else { System.out.println("This option is only available for Savings accounts."); }

                        break;

                case 5: System.out.println("Exiting... Thank you for using our services.");

                        break;

               default: System.out.println("Invalid choice! Please select a valid option.");

                        break; } }

                        while (choice != 5);

                        sc.close(); } }





