package com.atmprocess;

import java.util.Arrays;

import java.util.List;

import java.util.Scanner;

public class Assign {

    static Assign atm = new Assign();

    static Create[] ATM = {

            new Create(2000,20,40000),

            new Create(500,10,5000),

            new Create(100,100,10000)

    };

    static List<Create> Atm\_attributes = Arrays.asList(ATM);

    static CustomerDetails[] customerDetails = {

            new CustomerDetails(101,"abd",7890,25234),

            new CustomerDetails(102,"joe",5432,34123),

            new CustomerDetails(103,"root",7854,26100),

            new CustomerDetails(104,"kane",2345,80000),

            new CustomerDetails(105,"williamson",1907,103400)

    };

    static List<CustomerDetails> CUS = Arrays.asList(custtomerDetails);

    public int checkBalance(int accountNumber,int pin)

    {

        for(CustomerDetails customer : cust){

            if(customer.getAccNo() == accountNumber && customer.getPin()==pin){

                return customer.getAccBal();

            }

        }

        return 0;

    }

    public boolean withDrawPossible(int accountNumber,int pin,int amt){

        for(CustomerDetails customer : cust){

            if(customer.getAccNo() == accountNumber && customer.getPin()==pin && customer.getAccBal()>=amt){

                customer.setAccBal(customer.getAccBal()-amt);

                return true;

            }

        }

        return false;

    }

    public boolean transferPossible(int accountNumber,int pin,int amt,int otherAccNo){

        for(CustomerDetails customer : cust){

            if(customer.getAccNo() == accountNumber && customer.getPin()==pin&&customer.getAccBal()>=amt){

                customer.setAccBal(customer.getAccBal()-amt);

            }

        }

        for(CustomerDetails customer : cust){

            if(customer.getAccNo() == otherAccNo){

                customer.setAccBal(customer.getAccBal()+amt);

                return true;

            }

        }

        return false;

    }

    public static boolean correctPinOrNot(int acc)

    {

        for(CustomerDetails customer : CUS){

            if(customer.getAccNo() == acc){

                return true;

            }

        }

        return false;

    }

    public static boolean correctPinOrNot(int acc,int pin)

    {

        for(CustomerDetails customer : CUS){

            if(customer.getAccNo() == acc && customer.getPin()==pin){

                return true;

            }

        }

        return false;

    }

    public static void main(String[] args) {

        int amount;

        Scanner scanner = new Scanner(System.in);

        boolean condition = true;

        int otherAccNo;

        int accNo,pin;

        while (condition) {

            System.out.println("1 . Check Balance");

            System.out.println("2 . Withdraw Amount");

            System.out.println("3 . Transfer Amount");

            System.out.println("4 . AMT Balance");

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

            System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

            System.out.println("Enter Your Choice: ");

            int ch = scanner.nextInt();

            switch (ch) {

                case 1-> {

                    System.out.println("Please Enter Account Number: ");

                    accNo = scanner.nextInt();

                    System.out.println("Enter Your Four Digit Pin: ");

                    pin = scanner.nextInt();

                    if (correctPinOrNot(accNo, pin)) {

                        int balance = atm.checkBalance(accNo, pin);

                        System.out.println("\nYour Current Balance is " + balance);

                    }

                }

                case 2->{

                    System.out.println("Enter Account Number: ");

                    accNo = scanner.nextInt();

                    System.out.println("Enter Your Four Digit Pin: ");

                    pin = scanner.nextInt();

                    System.out.println(" Amount of Money to be withdrawn: ");

                    amount=scanner.nextInt();

                    if(correctPinOrNot(accNo,pin)){

                        if(amount>=2000){

                            if(atm.withDrawPossible(accNo,pin,amount)){

                                int withdrawAmount=amount;

                                if(Atm\_attributes.stream().filter(atm -> atm.getValue()>=withdrawAmount).count()>1){

                                    Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setValue(atm.getValue()-withdrawAmount));

                                    Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setNumber(atm.getNumber()-(withdrawAmount/2000)));

                                    System.out.println("amount withdrawn");

                                }

                            }

                        }

                        else if(amount>=500){

                            if(atm.withDrawPossible(accNo,pin,amount)){

                                int withdrawAmount=amount;

                                if(Atm\_attributes.stream().filter(atm -> atm.getValue()>=withdrawAmount).count()>1){

                                    Atm\_attributes.stream().filter(atm -> atm.getDenomination()==500).forEach(atm->atm.setValue(atm.getValue()-withdrawAmount));

                                    Atm\_attributes.stream().filter(atm -> atm.getDenomination()==500).forEach(atm->atm.setNumber(atm.getNumber()-(withdrawAmount/500)));

                                    System.out.println("amount withdrawn");

                                }

                            }

                        }

                        else if(amount>=100){

                            if(atm.withDrawPossible(accNo,pin,amount)){

                                int withdrawAmount=amount;

                                if(Atm\_attributes.stream().filter(atm -> atm.getValue()>=withdrawAmount).count()>1){

                                    Atm\_attributes.stream().filter(atm -> atm.getDenomination()==100).forEach(atm->atm.setValue(atm.getValue()-withdrawAmount));

                                    Atm\_attributes.stream().filter(atm -> atm.getDenomination()==100).forEach(atm->atm.setNumber(atm.getNumber()-(withdrawAmount/100)));

                                    System.out.println("amount withdrawn");

                                }

                            }

                        }

                    }

                }

                case 3->{

                    System.out.println("enter your Account Number: ");

                    accNo = scanner.nextInt();

                    System.out.println("Enter Your Four Digit Pin: ");

                    pin = scanner.nextInt();

                    System.out.println("Enter Amount of Money to be transfer: ");

                    amount=scanner.nextInt();

                    if(correctPinOrNot(accNo,pin)){

                        if(amount<=10000){

                            System.out.println("Enter Account Number of Your Friend: ");

                            otherAccNo= scanner.nextInt();

                            if(correctPinOrNot(otherAccNo)){

                                if(amount>=2000){

                                    if(atm.transferPossible(accNo,pin,amount,otherAccNo)){

                                        int transferAmount=amount;

                                        if(Atm\_attributes.stream().filter(atm -> atm.getValue()>=transferAmount).count()>1){

                                            Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setValue(atm.getValue()+transferAmount));

                                            Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setNumber(atm.getNumber()+transferAmount/2000));

                                            System.out.println("AMUNT transferee");

                                        }

                                    }

                                }

                                else if(amount>=500){

                                    if(atm.transferPossible(accNo,pin,amount,otherAccNo)){

                                        int transferAmount=amount;

                                        if(Atm\_attributes.stream().filter(atm -> atm.getValue()>=transferAmount).count()>1){

                                            Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setValue(atm.getValue()+transferAmount));

                                            Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setNumber(atm.getNumber()+transferAmount/500));

                                            System.out.println("AMUNT transferee");

                                        }

                                    }

                                }

                                else if(amount>=100){

                                    if(atm.transferPossible(accNo,pin,amount,otherAccNo)){

                                        int transferAmount=amount;

                                        if(Atm\_attributes.stream().filter(atm -> atm.getValue()>=transferAmount).count()>1){

                                            Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setValue(atm.getValue()+transferAmount));

                                            Atm\_attributes.stream().filter(atm -> atm.getDenomination()==2000).forEach(atm->atm.setNumber(atm.getNumber()+transferAmount/100));

                                            System.out.println("AMUNT transferee");

                                        }

                                    }

                                }

                            }

                        }

                        else{

                            System.out.println("Amount Must be less than 10000");

                        }

                    }

                }

                case 4-> Atm\_attributes.forEach(System.out::println);

                case 5-> condition=false;

                default -> System.out.println("Enter from 1,2,3,4,5");

            }

        }

    }