

ชื่อ-นามสกุล ..... รหัสนิสิต ..... หมู่ .....

## Lab 4 : Class and Object

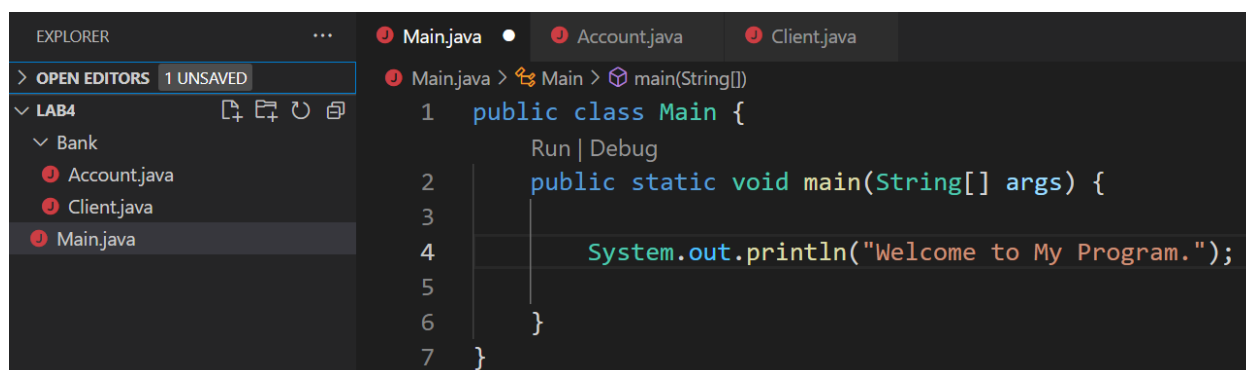
ให้นักนิสิตเขียนโปรแกรมภาษา Java แก้ปัญหาโจทย์ต่อไปนี้

### 1. สร้างไฟล์ให้พร้อมใช้งาน

1.1 ให้นักนิสิตสร้างโฟลเดอร์ใหม่ชื่อ “Lab4” ซึ่งในที่นี้จะกำหนดให้เป็นโฟลเดอร์หลัก และให้นักนิสิตเปิดโฟลเดอร์นั้นด้วย VS code

1.2 ให้นักนิสิตสร้างโฟลเดอร์ชื่อ “Bank” ขึ้นในโฟลเดอร์หลัก จากนั้นสร้างไฟล์ Account.java และ Client.java ขึ้นมาพร้อมกับกำหนด package และ class ให้กับทั้งสองไฟล์

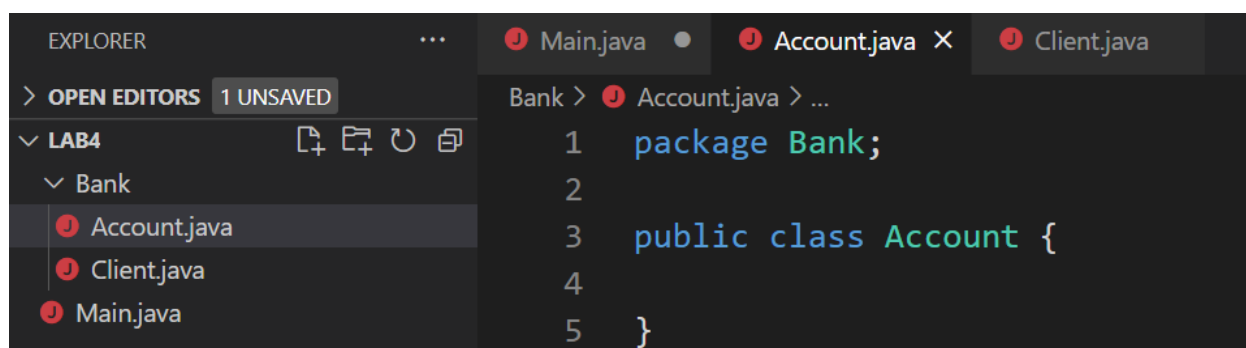
1.3 สร้างไฟล์ Main.java ในโฟลเดอร์หลักให้สามารถพิมพ์คำว่า “Welcome to My Program.” ออกทางจอภาพได้ ดังตัวอย่าง



```

EXPLORER
> OPEN EDITORS 1 UNSAVED
LAB4
  Bank
    Account.java
    Client.java
    Main.java

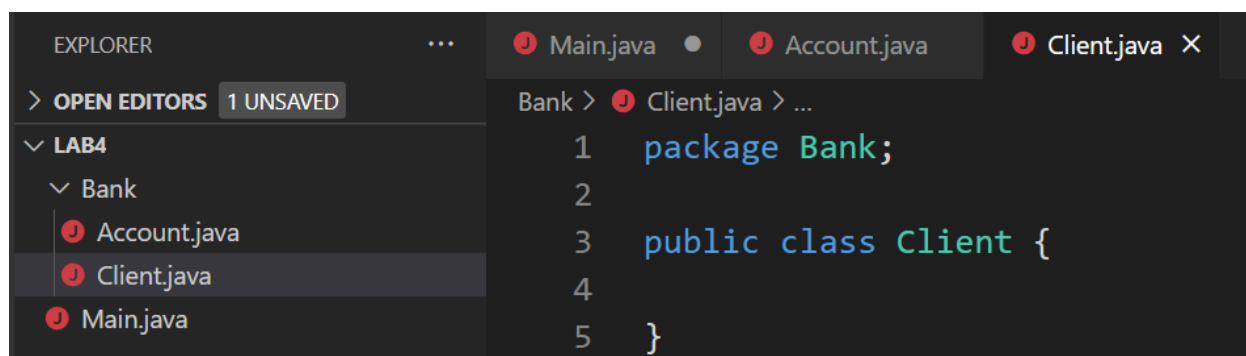
Main.java > Main > main(String[])
1  public class Main {
2      Run | Debug
3      public static void main(String[] args) {
4          System.out.println("Welcome to My Program.");
5      }
6  }
7  }
  
```



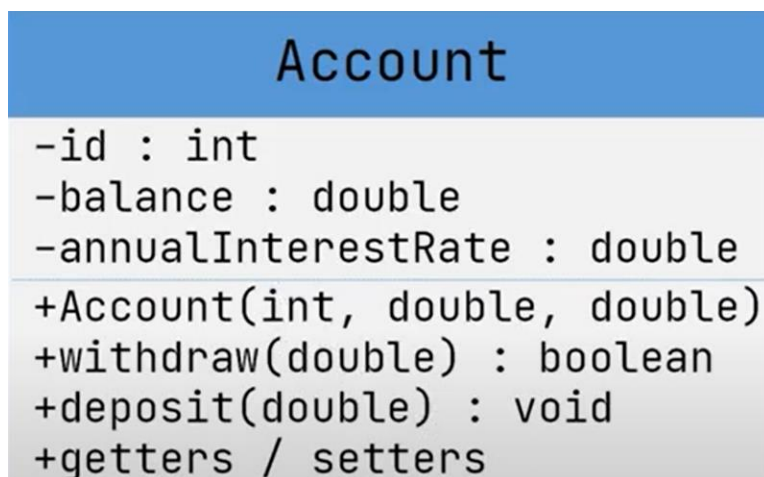
```

EXPLORER
> OPEN EDITORS 1 UNSAVED
LAB4
  Bank
    Account.java
    Client.java
    Main.java

Bank > Account.java > ...
1  package Bank;
2
3  public class Account {
4
5  }
  
```



2. สร้างคลาส Account (บัญชี) ให้พร้อมใช้งานตาม UML Class Diagram ต่อไปนี้



โดย id คือ เลขบัญชี

Balance คือ จำนวนเงินในบัญชี

annualInterestRate คือ อัตราดอกเบี้ย

Account คือ Constructor ที่กำหนดค่าเริ่มต้นให้ตัวแปรตามค่าที่รับเข้ามา

withdraw คือ การถอนเงิน

deposit คือ การฝากเงิน

getters / setters คือ เมธอดของการเรียกดูค่าและเซตค่าของทุกแอตทริบิวต์

## โปรแกรมคำตอบ

```
package Bank;

public class Account{

    private int id ;
    private double balance ;
    private double annualInterestRate ;

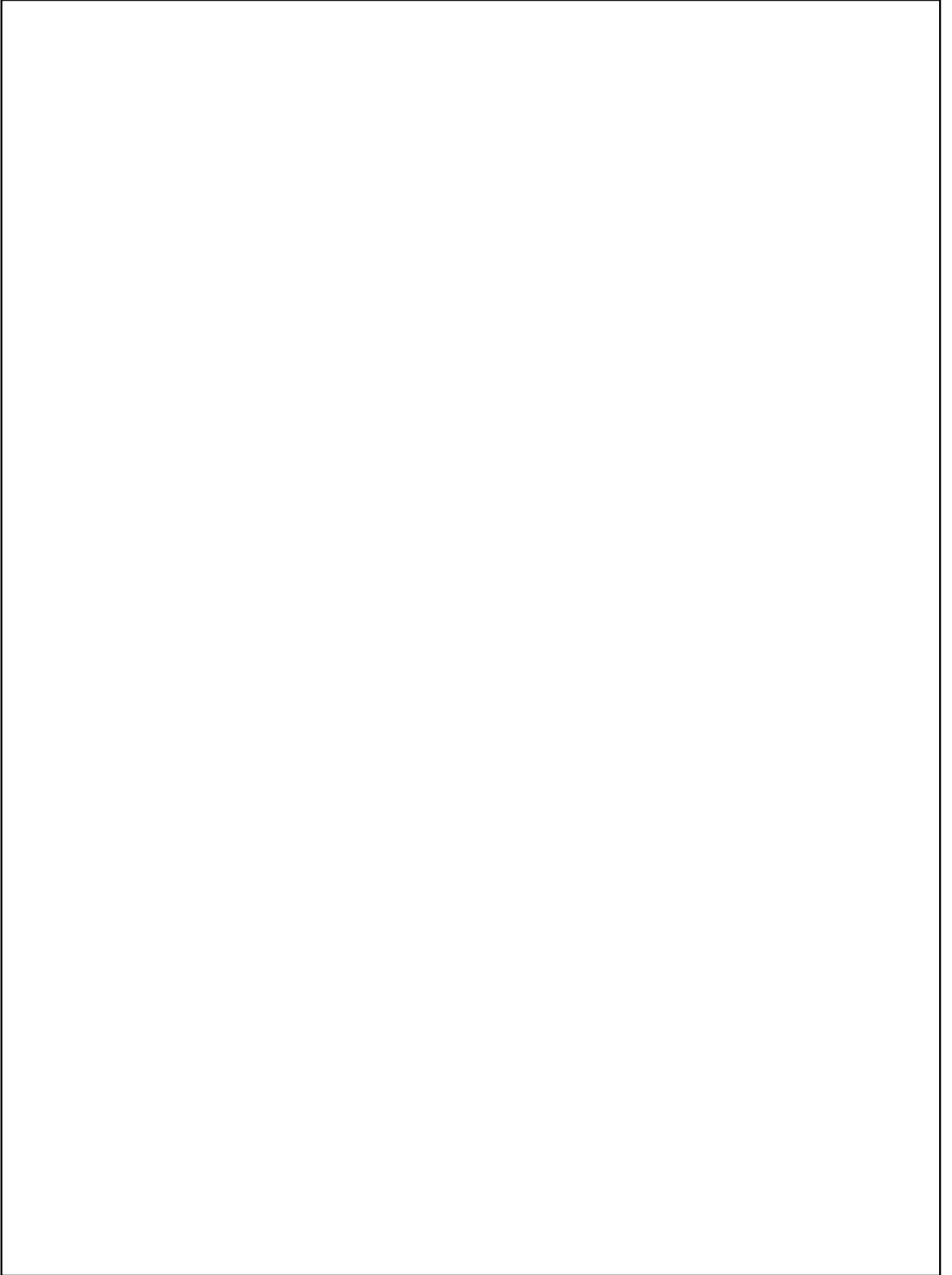
    public Account(int id,double balance,double annualInterestRate){
        this.id = id ;
        this.balance = balance ;
        this.annualInterestRate = annualInterestRate ;
    }

    public boolean withdraw(double value){

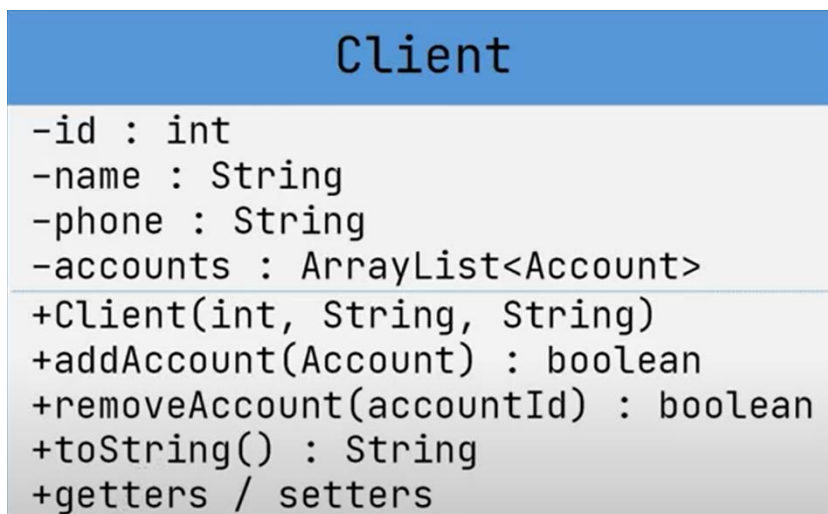
        if(this.balance >= value)
        {
            this.balance -= value ;
            System.out.println("Withdraw "+value+" from Account ID : "+this.id);
            return true ;
        }
        return false ;
    }

    public void deposit(double value){
        System.out.println("Deposit "+value+" to Account ID : "+this.id);
        this.balance += value ;
    }

    public int getId(){
        return this.id ;
    }
    public void setId(int id){
        this.id = id ;
    }
    public double getBalance(){
        return this.balance ;
    }
    public void setBalance(double balance){
        this.balance = balance ;
    }
    public double getAnnualInterestRate(){
        return this.annualInterestRate ;
    }
    public void setAnnualInterestRate(double annualInterestRate){
        this.annualInterestRate = annualInterestRate ;
    }
}
```



### 3. สร้างคลาส Client (ผู้ใช้งาน) ให้พร้อมใช้งานตาม UML Class Diagram ต่อไปนี้



โดย id คือ รหัสบัตรประชาชน

Name คือ ชื่อ-นามสกุล

Phone คือ เบอร์โทรศัพท์

Account คือ บัญชีธนาคารที่มี

Client คือ Constructor ที่กำหนดค่าเริ่มต้นให้ตัวแปรตามค่าที่รับเข้ามา

addAccount คือ การเพิ่มบัญชี

removeAccount คือ การลบบัญชี

toString() คือ การแสดงข้อมูลส่วนบุคคล และแสดงบัญชีทั้งหมดที่มี

(ในแต่ละบัญชีให้แสดง เลขบัญชี จำนวนเงิน และอัตราดอกเบี้ย)

getters / setters คือ เมธอดของการเรียกดูค่าและเซตค่าของ**ทุกแอตทริบิวต์**

## โปรแกรมคำตอบ

```

package Bank;

import java.util.ArrayList;
public class Client {

    private int id ;
    private String name ;
    private String phone ;
    public ArrayList<Account> allAccount ;

    public Client(int id,String name,String phone){
        this.id = id ;
        this.name = name ;
        this.phone = phone ;
        allAccount = new ArrayList<>();
    }

    public boolean addAccount(Account tmp){
        this.allAccount.add(tmp) ;
        return true ;
    }

    public boolean removeAccount(int accountId){

        for(int i=0 ; i < this.allAccount.size() ; i++) {
            if(this.allAccount.get(i).getId() == accountId)
            {
                System.out.println("Account ID : "+accountId+" has been removed.");
                this.allAccount.remove(i);
                return true ;
            }
        }
        return false ;
    }

    public String toString(){

        String ans = "";
        ans += "ID : "+this.id+" , Name : "+this.name+" , TEL : "+this.phone +"\n";
        ans += "Accounts \n" ;
        for (Account ac : this.allAccount) {
            ans += "Account ID = "+ac.getId()+" , Balance = "+ac.getBalance()+
                " , Annual Interest Rate = "+ac.getAnnualInterestRate()+"\n";
        }
        return ans ;
    }
}

```

```
public int getId(){
    return this.id ;
}
public void setId(int id){
    this.id = id ;
}
public String getName(){
    return this.name ;
}
public void setName(String name){
    this.name = name ;
}
public String getPhone(){
    return this.phone ;
}
public void setPhone(String phone){
    this.phone = phone ;
}
}
```

## 4. การใช้งาน Account และ Client ผ่าน Main.java

4.1 ให้นิสิตสร้างผู้ใช้งานสองคนโดย (ให้ชนิดข้อมูลเป็น Array of Client )

4.2 กำหนดให้คนที่ 1

ชื่อ Jame Panter รหัสบัตรประชาชน 144556677 เบอร์โทรศัพท์ 0811151511

มีบัญชีสองบัญชี คือ 1. เลขบัญชี 4450123 มีเงินจำนวน 20,000 บาท อัตราดอกเบี้ย 1.5%

2. เลขบัญชี 4450295 มีเงินจำนวน 90,000 บาท อัตราดอกเบี้ย 2.0%

4.2 กำหนดให้คนที่ 2

ชื่อ Camelo Antony รหัสบัตรประชาชน 122338899 เบอร์โทรศัพท์ 0899959599

มีบัญชีสองบัญชี คือ 1. เลขบัญชี 4550234 มีเงินจำนวน 10,000 บาท อัตราดอกเบี้ย 1.5%

2. เลขบัญชี 4550379 มีเงินจำนวน 80,000 บาท อัตราดอกเบี้ย 3.0%

3. เลขบัญชี 4550987 มีเงินจำนวน 50,000 บาท อัตราดอกเบี้ย 4.0%

4.3 ให้ทำการป็นข้อมูลทั้งหมดของทั้งสองคนออกมาโชว์ผ่านเมธอด toString()

4.4 ในส่วนของคุณ Camelo Antony ให้ทำการถอนเงินจากบัญชีที่ 1 ทั้งหมดไปฝากไว้ในบัญชีที่ 3 แล้วทำการลบบัญชีที่ 1 ที่

4.5 ให้ทำการป็นข้อมูลทั้งหมดของทั้งสองคนออกมาโชว์ผ่านเมธอด toString() อีกครั้ง



```

Welcome to My Program.
ID : 144556677 , Name : Jame Panter , TEL : 0811151511
Accounts
Account ID = 4450123 , Balance = 20000 , Annual Interest Rate = 1.5 %
Account ID = 4450295 , Balance = 90000 , Annual Interest Rate = 2.0 %
ID : 122338899 , Name : Camelo Antony , TEL : 0899959599
Accounts
Account ID = 4550234 , Balance = 10000 , Annual Interest Rate = 1.5 %
Account ID = 4550379 , Balance = 80000 , Annual Interest Rate = 3.0 %
Account ID = 4550987 , Balance = 50000 , Annual Interest Rate = 4.0 %

Withdraw 10000 from Account ID : 4550234
Deposit 10000 to Account ID : 4550987
Account 4550234 has been removed

ID : 144556677 , Name : Jame Panter , TEL : 0811151511
Accounts
Account ID = 4450123 , Balance = 20000 , Annual Interest Rate = 1.5 %
Account ID = 4450295 , Balance = 90000 , Annual Interest Rate = 2.0 %
ID : 122338899 , Name : Camelo Antony , TEL : 0899959599
Accounts
Account ID = 4550379 , Balance = 80000 , Annual Interest Rate = 3.0 %
Account ID = 4550987 , Balance = 60000 , Annual Interest Rate = 4.0 %

```

ตัวอย่างผลการรันโปรแกรม

โปรแกรมคำตอบ

```

import java.util.ArrayList;

import Bank.*;

public class Main {

    public static void main(String[] args) {

        System.out.println("Welcome to My Program");

        Client std[] = new Client[2];
        std[0] = new Client(144556677, "Jame Panter", "0811151511");
        std[0].addAccount(new Account(4450123, 20000, 1.5));
        std[0].addAccount(new Account(4450295, 90000, 2.0));

        std[1] = new Client(122338899, "Camelo Antony", "0899959599");
        std[1].addAccount(new Account(4550234, 10000, 1.5));
        std[1].addAccount(new Account(4550379, 80000, 3.0));
        std[1].addAccount(new Account(4550987, 50000, 4.0));

        System.out.println(std[0].toString()+std[1].toString());

        double max = std[1].allAccount.get(0).getBalance();
        std[1].allAccount.get(0).withdraw(max);
        std[1].allAccount.get(2).deposit(max);
        std[1].removeAccount( std[1].allAccount.get(0).getId() );

        System.out.println(std[0].toString()+std[1].toString());

        /*
        ArrayList<Client> Astd = new ArrayList<>();

        Astd.add(new Client(144556677, "Jame Panter", "0811151511"));
        Astd.get(0).addAccount(new Account(4450123, 20000, 1.5));
        Astd.get(0).addAccount(new Account(4450295, 90000, 2.0));

        Astd.add(new Client(122338899, "Camelo Antony", "0899959599"));
        Astd.get(1).addAccount(new Account(4550234, 10000, 1.5));
        Astd.get(1).addAccount(new Account(4550379, 80000, 3.0));
        Astd.get(1).addAccount(new Account(4550987, 50000, 4.0));
        */
    }
}

```