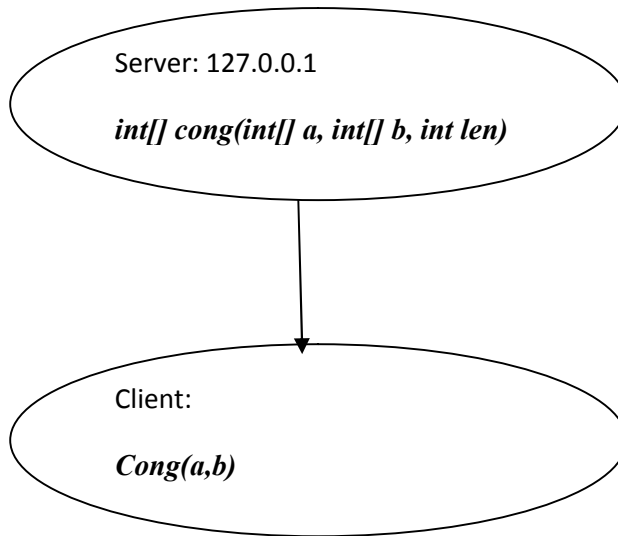


# BÀI TẬP RMI

## BÀI THỰC HÀNH 1: RMI

- Trên Server cung cấp hàm cộng 2 ma trận
- Client tạo 2 ma trận ngẫu nhiên và yêu cầu Server để cộng 2 ma trận này, Client hiển thị kết quả sau khi cộng.



### Các bước để thực hiện chương trình:

1. Trên Server tạo interface để khai báo hàm cộng 2 ma trận
2. Trên Server cài đặt giao diện để cộng 2 ma trận
3. Xây dựng lớp MyServer để quảng bá đối tượng để cộng 2 ma trận và đăng ký 1 địa chỉ với RMI.
4. Tạo file Server.bat bao gồm các lệnh để dịch chương trình Server
5. Xây dựng lớp Client để dò tìm địa chỉ mà Server đã đăng ký và gọi hàm cộng ma trận trên Server
6. Tạo file Client.bat bao gồm các lệnh để dịch chương trình Client

### Cài đặt các bước:

1. Trên Server tạo interface: **ITinhToan** để khai báo hàm cộng 2 ma trận

```
import java.rmi.*;
```

```
public interface ITinhToan extends Remote {  
    public int[] cong(int[] a, int[] b, int len) throws RemoteException;  
}
```

2. Trên Server tạo lớp **TinhToan** cài đặt giao diện để cộng 2 ma trận

```
import java.rmi.*;
```

```
public class TinhToan implements ITinhToan {  
    public int[] cong(int[] a, int[] b, int len) throws RemoteException  
    {
```

```

        int[] c=new int[len];
        for(int i=0;i<len;i++)//Cong 2 ma tra a va b
            c[i]=a[i]+b[i];
        for(int j=0;i<len;i++)//Hien thi ma tran c
        System.out.print(c[j]+" ");
        return c;
    }
}

```

**3. Xây dựng lớp MyServer để quảng bá đối tượng để cộng 2 ma trận và đăng ký 1 địa chỉ với RMI.**

```

import java.rmi.*;
import java.rmi.server.*;
public class MyServer {
    public static void main(String[] args) {
        try{
            TinhToan tt= new TinhToan();
            UnicastRemoteObject.exportObject(tt);
            InetAddress ip=InetAddress.getLocalHost();//Lay ip cua may
            Naming.bind("rmi://" + ip.getHostName() + "/" + "TinhToan",tt);
            System.out.print("Dang cho Client yeu cau: ");
        }catch(Exception tt) { System.out.print(tt); }
    }
}

```

4. Tạo file Server.dat bao gồm các lệnh để dịch chương trình Server

```

javac *.java
rmic TinhToan
start rmiregistry
java MyServer

```

Sao chép lớp: TinhToan\_Sub.class sang client

5. Xây dựng lớp Client để dò tìm địa chỉ mà Server đã đăng ký và gọi hàm cộng ma trận trên Server:

```

import java.rmi.*;
public class Client {
    public static void xuat(int[] a, int len)
    {

```

```

    for(int i=0;i<len;i++)
        System.out.print(a[i]+" ");
    System.out.println();
}

public static void main(String[] st)
{
    try{
        ITinhToan server = (ITinhToan) Naming.lookup("rmi://localhost/TinhToan");
        Random r=new Random();
        int len=10000;
        int[] a=new int[len];
        int[] b=new int[len];
        for(int i=0;i<len;i++)//Tạo 2 mảng ngẫu nhiên
        {
            a[i]=r.nextInt(100);
            b[i]=r.nextInt(100);
        }
        xuất(a,len);//Xuất mảng a
        xuất(b,len);//Xuất mảng b
        int[] c=server.cong(a,b,len);
        xuất(c,len);//Xuất mảng sau khi cộng
    }catch(Exception tt){ System.out.print(tt.getMessage());}
}
}

```

6. Tạo file Client.bat bao gồm các lệnh để dịch chương trình Client

```
javac *.java
```

```
java Client
```

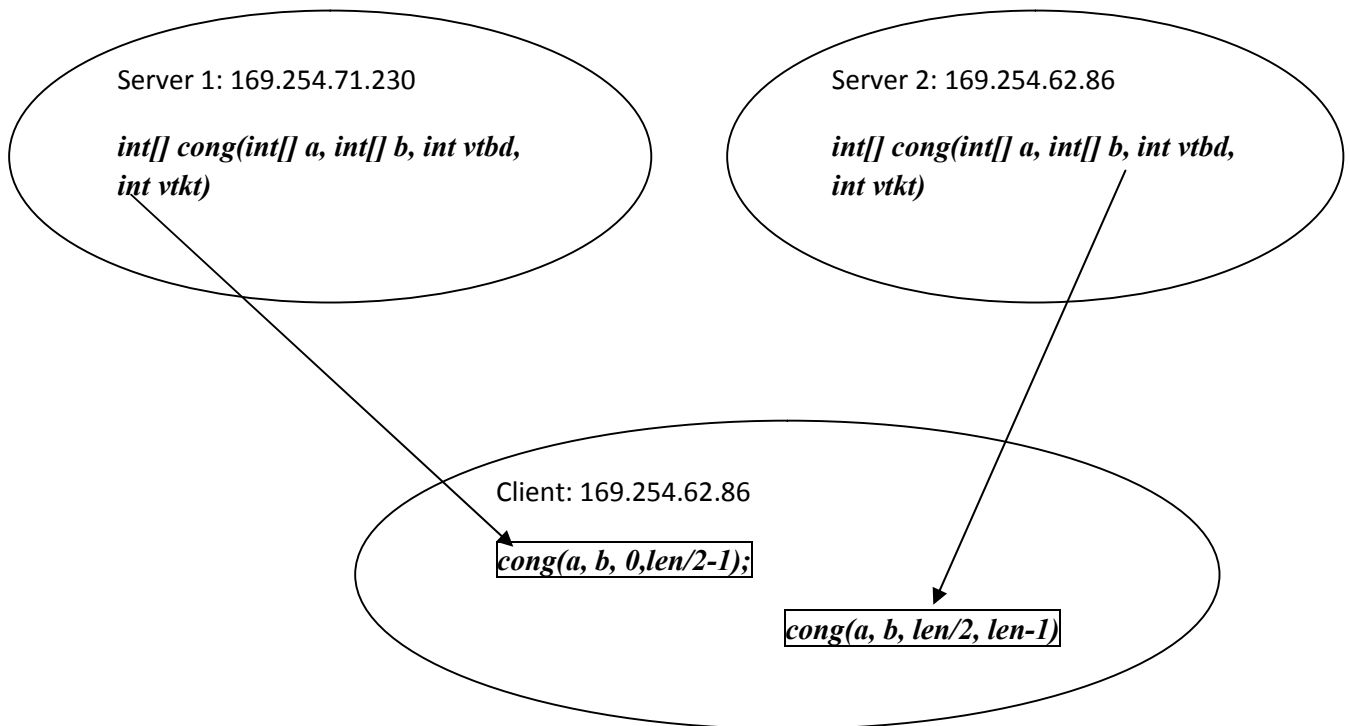
7. Chạy file Server.bat và Client.bat

8. Yêu cầu bổ sung

- Trên server cung cấp thêm hàm nhân 2 ma trận vuông a,b
- Trên Client gọi lại hàm Cong và nhân 2 ma trận

## BÀI THỰC HÀNH 2: RMI và Thread để xử lý đồng thời

- Yêu cầu: chương trình cần 2 máy để thực hành: 1 máy vừa Server 1 vừa client, máy còn lại làm Server 2
- Ta có 2 Server: trên mỗi Server cung cấp hàm cộng 2 ma trận
- Client tạo 2 ma trận ngẫu nhiên và yêu cầu đồng thời Server 1 cộng 1 nửa ma trận đầu, Server 2 cộng nửa còn lại, Client hiển thị kết quả sau khi cộng.



1. Trên Server tạo interface để khai báo hàm cộng 2 ma trận
2. Trên Server cài đặt giao diện để cộng 2 ma trận
3. Xây dựng lớp MyServer để quảng bá đối tượng để cộng 2 ma trận và đăng ký 1 địa chỉ với RMI.
4. Tạo file Server.bat bao gồm các lệnh để dịch chương trình Server
5. Xây dựng lớp Client để:
  - a. Tạo ngẫu nhiên 2 ma trận a và b
  - b. Tạo 2 tuyến đoạn chạy đồng thời
  - c. Tuyến đoạn 1 dò tìm địa chỉ mà Server 1 đã đăng ký và gọi hàm cộng ma trận trên Server để cộng nửa đầu tiên của 2 ma trận
  - d. Tuyến đoạn 2 dò tìm địa chỉ mà Server 2 đã đăng ký và gọi hàm cộng ma trận trên Server để cộng nửa cuối của 2 ma trận
6. Tạo file Client.bat bao gồm các lệnh để dịch chương trình Client

### Cài đặt các bước:

#### 1. Trên Server tạo interface: *ITinhToan* để khai báo hàm cộng 2 ma trận

```
import java.rmi.*;
```

```

public interface ITinhToan extends Remote {
    public SinhVien GetSv(SinhVien sv) throws RemoteException;
    public int[] cong(int[] a, int[] b, int vtbd, int vtk) throws RemoteException;
}

```

Trong đó:  $a, b$  là 2 ma trận cần tính tổng

Vtbd: vị trí bắt đầu tính tổng trên ma trận  $a$  và  $b$

Vtk: vị trí cuối cần tính tổng trên ma trận  $a$  và  $b$

Ví dụ: ma trận  $a$

0	1	2	3	4	5	6	7
Giá trị 1	...	...	..	...	...	Giá trị 6	Giá trị 7

Ví dụ: ma trận  $b$

0	1	2	3	4	5	6	7
Giá trị 1	...	...	..	...	...	Giá trị 6	Giá trị 7

$Cong(a, b, 0, 3)$ : Cộng  $\frac{1}{2}$  đầu

$Cong(a, b, 4, 7)$ : Cộng  $\frac{1}{2}$  còn lại

## 2. Trên Server tạo lớp TinhToan cài đặt giao diện để cộng 2 ma trận

```
import java.rmi.*;
```

```
public class TinhToan implements ITinhToan {
```

```
    public int[] cong(int[] a, int[] b, int vtbd, int vtk) throws RemoteException
```

```
{
```

```
    int[] c = new int[vtk - vtbd + 1]; // Tạo 2 ma trận ma trận c để chứa kết quả sau khi cộng
```

```
    int j = 0;
```

```
    for (int i = vtbd; i <= vtk; i++) // Cộng
```

```
    { c[j] = a[i] + b[i]; j = j + 1; }
```

```
    System.out.println("Ma tran ket qua");
```

```
    for (int t = 0; t < c.length; t++)
```

```
        System.out.print(c[t] + " ");
```

```
    System.out.println();
```

```
    return c;
```

```
}
```

```
}
```

## 3. Xây dựng lớp MyServer để quảng bá đối tượng để cộng 2 ma trận và đăng ký 1 địa chỉ với RMI.

```
import java.rmi.*;
```

```
import java.rmi.server.*;
```

```

import java.net.*;
public class MyServer {
    public static void main(String[] args) {
        try{
            TinhToan tt= new TinhToan();
            UnicastRemoteObject.exportObject(tt);
            InetAddress ip=InetAddress.getLocalHost();
            Naming.bind("rmi://" + ip.getHostName() + "/TinhToan",tt);
            System.out.print("Dang cho Client yeu cau: ");
        }catch(Exception tt) { System.out.print(tt); }
    }
}

```

#### 4. Tạo file Server.bat bao gồm các lệnh để dịch chương trình Server

```

javac *.java
rmic TinhToan
start rmiregistry
java MyServer

```

Sao chép lớp: TinhToan\_Sub.class sang client

#### 5. Xây dựng lớp Client để:

- Tạo ngẫu nhiên 2 ma trận a và b
- Tạo 2 tuyến đoạn chạy đồng thời
- Tuyến đoạn 1 dò tìm địa chỉ mà Server 1 đã đăng ký và gọi hàm cộng ma trận trên Server để cộng nửa đầu tiên của 2 ma trận
- Tuyến đoạn 2 dò tìm địa chỉ mà Server 2 đã đăng ký và gọi hàm cộng ma trận trên Server để cộng nửa cuối của 2 ma trận

```

import java.util.*;
import java.rmi.*;
public class Client {
    public static Thread[] td= new Thread[2] ;//Tạo 2 tuyến đoạn
    public static int somay=2;//Số Server tham gia tính toán
    //Địa chỉ ip của các Server, sinh viên tự thay đổi địa chỉ ip tương ứng khi thực hành
    public static String[] ip={"169.254.71.230","169.254.62.86"};
    public static int len=10000;//Chiều dài của 2 mảng a và b
    public static int[] a=new int[len];//Tạo ra mảng a
    public static int[] b=new int[len];//Tạo ra mảng b

```

*public static class Xuly extends Thread//Tạo ra 1 lớp con để xử lý đồng thời*

*{ int i;*

*public Xuly(int i) {*

*this.i = i;//i là tuyến đoạn I cần tính*

*}*

*public void run() {Tự động chạy khi tuyến đoạn bắt đầu*

*try {dò tìm Server ip[i]}*

*ITinhToan server = (ITinhToan) Naming.lookup("rmi://" + ip[i] + "/TinhToan");*

*System.out.println("Server:" + ip[i] + " đang tính toán: ");*

*if (i == 0) {Nếu tuyến đoạn I làm làm việc*

*xuat(server.cong(a, b, 0, len/2-1));//Xuất ma trận kết quả: ½ ma trận đầu*

*System.out.println("Server:" + ip[i] + " đã tính toán ");*

*td[0].stop() ;//Kết thúc tuyến đoạn 0*

*}*

*else//Tuyến đoạn 1 đang làm việc*

*{*

*xuat(server.cong(a, b, len/2, len-1));// Xuất ma trận kết quả: ½ ma trận cuối*

*System.out.println("Server:" + ip[i] + " đã tính toán ");*

*td[1].stop(); //Kết thúc tuyến đoạn 1*

*}*

*} catch (Exception tt) {   System.out.print("loi"); }*

*//Kết thúc run()*

*//Kết thúc lớp xử lý*

*//Xây dựng hàm xuất trong lớp Client để xuất ra giá trị của ma trận nào đó*

*public static void xuat(int[] a)*

*{*

*for(int i=0;i<a.length ;i++)*

*System.out.print(a[i]+" " ) ;*

*System.out.println();*

*}*

*//Hàm main của lớp Client*

*public static void main(String[] st)*

*{*

*try{*

*Random r=new Random();//Lấy ngẫu nhiên 2 ma trận a và b*

```

for(int i=0;i<len;i++)
{
    a[i]=r.nextInt(100);
    b[i]=r.nextInt(100);
}
xuat(a);
xuat(b);
for(int i=0;i<somay;i++)//Kích hoạt đồng thời 2 tuyến đoạn
{
    td[i]= new Xuly(i) ;
    //Kích hoạt tuyến đoạn I hoạt động, hàm run() của lớp Xuly sẽ thực hiện để
    // dò tìm máy chủ có ip lưu ở mảng ip[i] và gọi hàm cộng trên máy chủ này để
    //nếu i=0 sẽ tính tổng ½ mảng đầu, i=1 sẽ tính nửa còn lại
    td[i].start();//Kích hoạt tuyến đoạn i
}
}catch(Exception tt){ System.out.print(tt.getMessage()) ;}
}
}

```

6. Tạo file Client.bat bao gồm các lệnh để dịch chương trình Client

```
javac *.java
```

```
java Client
```

9. Sao chép các lớp *ItinhToan*, *TinhToan*, *Myserver* và file *Server.dat* sang máy chủ 2 và chạy file *Server.bat*.

*Trên máy chủ 1: chạy file Server.bat và chạy lớp Client để xem kết quả*

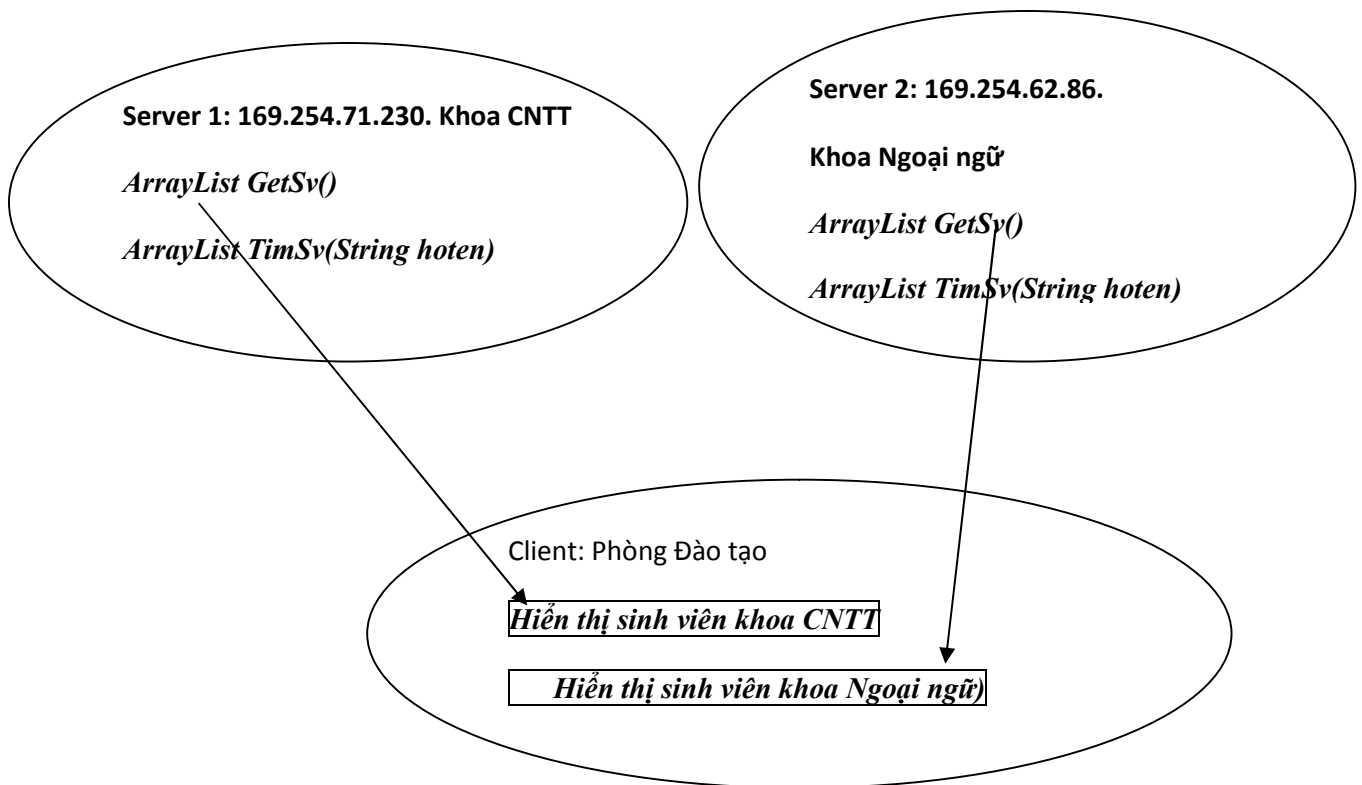
10. Yêu cầu bổ sung

- Trên server cung cấp thêm hàm nhân 2 ma trận vuông  $a, b$
- Trên Client: tại 1 thời điểm mở 2 tuyến đoạn, mỗi tuyến đoạn nhân ½ ma trận  $a$  và  $b$



### BÀI THỰC HÀNH 3: RMI, Thread và SQLServer

Giả sử khoa CNTT học có 1 máy chủ lưu csdl: QISv bao gồm bảng SinhVien(MaSv,HoTen,DTB) để lưu thông tin của khoa mình, Khoa Ngoại ngữ cũng có 1 máy chủ lưu csdl: QISv bao gồm bảng SinhVien(MaSv,HoTen,DTB) để lưu thông tin của sinh viên. Phòng đào tạo cần danh sách các sinh viên của 2 khoa.



Trong đó: ArrayList chứa thông tin của sinh viên, mỗi phần tử của ArrayList chứa 1 sinh viên bao gồm: MaSv, HoTen, DTB

#### Chuẩn bị:

Trên cả 2 Server: CNTT và Ngoại ngữ tạo CSDL: QISv bao gồm bảng SinhVien(MaSv,HoTen, DTB), nhập vào các sinh viên

#### Thuật toán:

Trên cả 2 Server: CNTT và Ngoại ngữ tạo:

1. Tạo lớp SinhVien để truyền theo tham trị
2. interface để khai báo 2 hàm để lấy về các sinh viên và tìm kiếm sinh viên theo họ tên
3. Cài đặt giao diện để cài 2 hàm trên giao diện
4. Xây dựng lớp MyServer để quảng bá đối tượng và đăng ký 1 địa chỉ với RMI.
5. Tạo file Server.bat bao gồm các lệnh để dịch chương trình Server

Trên Client:

1. Tạo lớp SinhVien giống như lớp SinhVien trên Server

## 2. Xây dựng lớp Frame: FrmClient để:

- Tạo 2 tuyến đoạn chạy đồng thời
- Tuyến đoạn 1 dò tìm địa chỉ mà Server 1 (Khoa CNTT) đã đăng ký và gọi hàm để lấy về các sinh viên và hiển thị
- Tuyến đoạn 2 dò tìm địa chỉ mà Server 2 (Khoa Ngoại Ngữ) đã đăng ký và gọi hàm để lấy về các sinh viên và hiển thị

### Cài đặt các bước:

#### 1. Trên cả 2 Server tạo lớp SinhVien để truyền bằng trị:

```
import java.io.*;

public class SinhVien implements Serializable {
    public String Masv;
    public String HoTen;
    public double DTB;
    public SinhVien(String Masv, String HoTen, double DTB) {
        this.Masv =Masv;
        this.HoTen =HoTen;
        this.DTB =DTB;
    }
}
```

#### 2. Trên cả 2 Server tạo lớp CoSo để kết nối đến CSDL và lấy dữ liệu của bảng SinhVien

```
import java.sql.*;
import java.net.*;

public class CoSo {
    public static Connection cn;
    public static void MoKetNoi() {
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            cn = DriverManager.getConnection("jdbc:sqlserver://tên máy:1433; databaseName=
            QlSv; user=...; password=...");
            System.out.println("Da ket noi CSDL ");
        }
        catch (Exception t1) {
            System.out.print(t1.getMessage());
        }
    }
}
```

```

public static ResultSet LayBang(String tb) {
    try {
        Statement cmd = cn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
        return cmd.executeQuery("select * from " + tb);
    }
    catch (Exception t2) {
        System.out.print(t2.getMessage());
        return null;
    }
}

```

**3. Trên 2 Server tạo interface: *ITinhToan* để lấy về các sinh viên và tìm kiếm họ tên**

```

import java.util.*;
import java.rmi.*;
public interface ITinhToan extends Remote {
    public ArrayList GetSv() throws RemoteException;
    public ArrayList TimSv(String hoten) throws RemoteException;
}

```

**4. Trên cả 2 Server tạo lớp TinhToan cài đặt giao diện**

```

import java.sql.*;
import java.util.*;
import java.rmi.*;
public class TinhToan implements ITinhToan {
    public ArrayList GetSv() throws RemoteException//Trả về thông tin của các sv
    {
        try{
            ArrayList ds= new ArrayList();//Tạo 1 ArrayList để chứa các sinh viên
            CoSo.MoKetNoi();//Mở kết nối đến CSDL
            ResultSet rs=CoSo.LayBang("SinhVien");//Lấy về toàn bộ bảng SinhVien
            while(rs.next())//Duyệt ResultSet để lưu sv vào ArrayList
            {
                SinhVien sv= new SinhVien(rs.getString("MaSv"),rs.getString("HoTen"),
                    Double.parseDouble(rs.getString("DTB")));
                ds.add(sv);
            }
        }
    }
}

```

```

    }
    CoSo.cn.close() ;
    return ds;
} catch(Exception tt)
{
    System.out.print("so bi loi"+tt);
    return null;
}
}
public ArrayList TimSv(String hoten) throws RemoteException//Tìm kiếm sv theo ht
{
    try{
        ArrayList ds= new ArrayList();
        CoSo.MoKetNoi();
        ResultSet rs=CoSo.LayBang("SinhVien");
        while(rs.next())
        {if(rs.getString("hoten").indexOf(hoten)>=0)//nếu thỏa mãn điều kiện
            {
                SinhVien sv= new SinhVien(rs.getString("MaSv"),rs.getString("HoTen"),
                                           Double.parseDouble("DTB"));

                ds.add(sv);
            }
        }
        CoSo.cn.close() ;
        return ds;

    } catch(Exception tt)
    {
        System.out.print(tt); return null;
    }
}

```

### **5. Xây dựng lớp MyServer để quảng bá đối tượng và đăng ký 1 địa chỉ với RMI.**

```

import java.rmi.*;
import java.rmi.server.*;
import java.net.*;
public class MyServer {

```

```

public static void main(String[] args) {
    try{
        TinhToan tt= new TinhToan();
        UnicastRemoteObject.exportObject(tt);
        InetAddress ip=InetAddress.getLocalHost();
        Naming.bind("rmi://" + ip.getHostName() + "/TinhToan",tt);
        System.out.print("Dang cho Client yeu cau: ");
    }catch(Exception tt) { System.out.print(tt); }
}
}

```

## 6. Tạo file Server.bat bao gồm các lệnh để dịch chương trình Server

```

javac *.java
rmic TinhToan
start rmiregistry
java -cp .;mysql-connector-java-5.1.8-bin.jar MyServer

```

**Chú ý:**

- Chép file mysql-connector-java-5.1.8-bin.jar vào thư mục chứa các lớp
- Sao chép lớp: TinhToan\_Sub.class và SinhVien.class sang client

## 7. Trên Client (Phòng đào tạo)

Xây dựng lớp Frame: FrmClient để:

- a. Tạo 2 tuyến đoạn chạy đồng thời
  - b. Tuyến đoạn 1 dò tìm địa chỉ mà Server 1 (Khoa CNTT) đã đăng ký và gọi hàm để lấy về các sinh viên và hiển thị
  - c. Tuyến đoạn 2 dò tìm địa chỉ mà Server 2 (Khoa Ngoại Ngữ) đã đăng ký và gọi hàm để lấy về các sinh viên và hiển thị
- **Tạo lớp NapBang để nạp 1 ArrayList vào DefaultTableModel**

```

import java.util.*;
import javax.swing.table.*;

public class NapBang {
    public static DefaultTableModel NapDuLieuVaoBang( ArrayList ds,int socot)
    {
        try{
            int SoCot=3;//masv, hoten va Dtb
            DefaultTableModel m= new DefaultTableModel(0,SoCot);

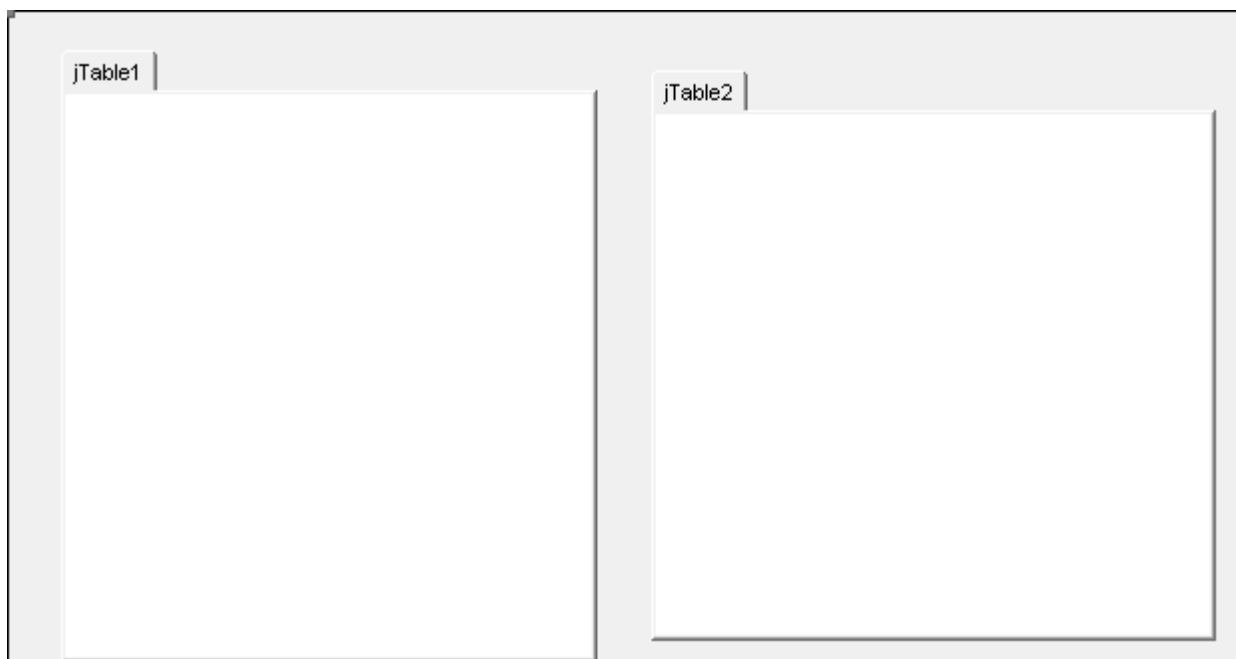
```

```

Object[] t= new Object[socot];
for(int i=0;i<ds.size() ;i++)
{
    SinhVien sv=(SinhVien)ds.get(i);
    t[0]=sv.Masv ;
    t[1]=sv.HoTen ;
    t[2]=String.valueOf(sv.DTB);
    m.addRow(t);
}
return m;
}catch(Exception tt)
{
    System.out.print("Loi roi"+tt) ;return null;
}
}
}

```

- **Tạo Frame: FrmClient có giao diện như sau:**



**Khai báo các biến và lớp Xuly để lấy dữ liệu về:**

```

public class FrmClient extends JFrame {
    public Thread[] td= new Thread[2] ;//Tạo 2 tuyến đoạn
    public int somay=2;
    public String[] ip={"169.254.71.230","169.254.62.86"};
    public class Xuly extends Thread

```

```

{ int i;
public Xuly(int i) {
    this.i = i;
}
public void run() {
    try {
        ITinhToan server = (ITinhToan) Naming.lookup("rmi://" + ip[i] + "/TinhToan");
        if (i == 0) { //Nếu tuyến đoạn td[0] thực hiện
            System.out.println("Server:" + ip[i] + " đang tính toán: ");
            ArrayList ds=new ArrayList();
            ds=server.GetSv();//Lấy về các sinh viên từ Server: 169.254.71.230
            jTable1.setModel(NapBang.NapDuLieuVaoBang(ds,4) );//Hiển thị dữ liệu
            td[0].stop() ;//dừng tuyến đoạn
        }
        else
        { System.out.println("Server:" + ip[i] + " đang tính toán: ");
            ArrayList ds=new ArrayList();
            ds=server.GetSv();//Lấy về các sinh viên từ Server: 169.254.62.86
            jTable2.setModel(NapBang.NapDuLieuVaoBang(ds,4) );
            td[1].stop() ;
        }
    } catch (Exception tt) {
        System.out.print(tt);
    }
} //kết thúc run()
} //Kết thúc lớp xử lý
...các mã thiết kế giao diện
} //kết thúc lớp FrmClient

- Lập trình trên sự kiện windowOpened để mở đồng thời 2 tuyến đoạn
void this_windowOpened(WindowEvent e) {
try{
    for(int i=0;i<somay;i++)
    {
        td[i]=new Xuly(i);
        td[i].start();
    }
}

```

```

    }
    }catch(Exception tt)
    {
        System.out.print(tt);
    }
}
}

```

**8. Sao chép các lớp *ItinhToan, TinhToan, Myserver* và file *Server.bat* sang máy chủ 2 và chạy file *Server.bat*.**

***Trên máy chủ 1: chạy file *Server.bat* và chạy lớp *Client* để xem kết quả***

**7. Yêu cầu bổ sung**

- Sinh viên tự thiết kế giao diện và lập trình để tìm kiếm sinh viên
- Mở rộng chương trình trên cho tất cả các khoa của trường