

LAB 05 – TCP



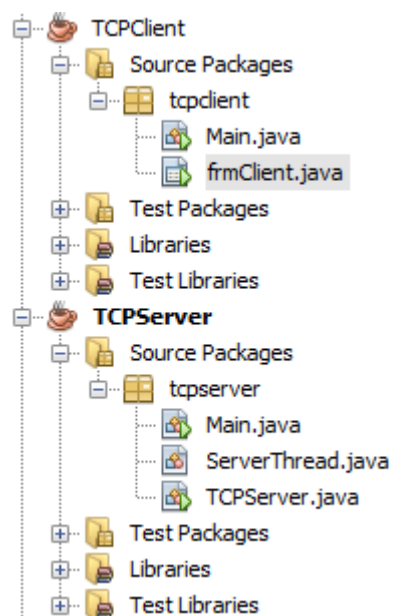
Bài 1: Viết chương trình giao tiếp giữa client và server sử dụng giao thức TCP, thực hiện các chức năng sau:

+ Client truyền 1 chuỗi lên server, server nhận chuỗi này và chuyển nó thành chữ in hoa sau đó gửi trả kết quả cho client.

+ Client nhận kết quả rồi sau đó xuất ra màn hình kết quả vừa nhận.

Yêu cầu: Sử dụng Multithread để server có thể giao tiếp được với nhiều client cùng lúc.

Bước 1: Tạo 2 project mới là TCPServer và TCPClient



Bước 2: Tạo Form frmClient có giao diện như sau:

Bước 3: Thêm khai báo vào đầu lớp frmClient

```
package tcpclient;

import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

/**
 *
 * @author Tuan
 */
public class frmClient extends javax.swing.JFrame {
    private Socket socket=null;
    private PrintWriter out=null;
    private Scanner in=null;
    /** Creates new form frmClient */
    public frmClient() {
        initComponents();
        txtchuoi.requestFocus();
    }
}
```

Bước 4: Xử lý sự kiện cho button Truyền chuỗi.

```
private void btntruyenchuoiActionPerformed(java.awt.event.ActionEvent evt) {
    String chuoi=txtchuoi.getText();//Lấy chuỗi
    String ketqua="";
    try{
        //Socket nhận tham số là địa chỉ Host và port
        socket=new Socket("127.0.0.1",1234);
        out=new PrintWriter(socket.getOutputStream(),true);
        in=new Scanner(socket.getInputStream());
        out.println(chuoi);//truyền chuỗi lên server để xử lý
        ketqua=in.nextLine().trim();//nhận chuỗi kết quả từ server
        txtketqua.setText(ketqua);//Hiển thị chuỗi nhận được từ server lên TextField
        socket.close();//Đóng Socket
    }catch(Exception e)
    {
        try{if(socket!=null) socket.close();}catch(Exception ex){e.printStackTrace();}
        e.printStackTrace();
    }
}
```

Bước 5: Xử lý sự kiện cho button Thoát

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

Bước 6: Tạo lớp TCPServer như sau:

```

public class TCPServer {
    static final int PORT = 1234;
    private ServerSocket server=null;
    public TCPServer() {
        try{
            server=new ServerSocket(PORT);
        }catch(Exception e){e.printStackTrace();}
    }
    public void action(){
        Socket socket=null;
        int i=0;
        System.out.println("Serverlistening...");
        try{
            while((socket=server.accept())!=null){
                new ServerThread(socket,"Client#"+i);
                System.out.printf("Thread for Client#%d generating...\n",i++);
            }
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public static void main(String[]args){
        new TCPServer().action();
    }
}

```

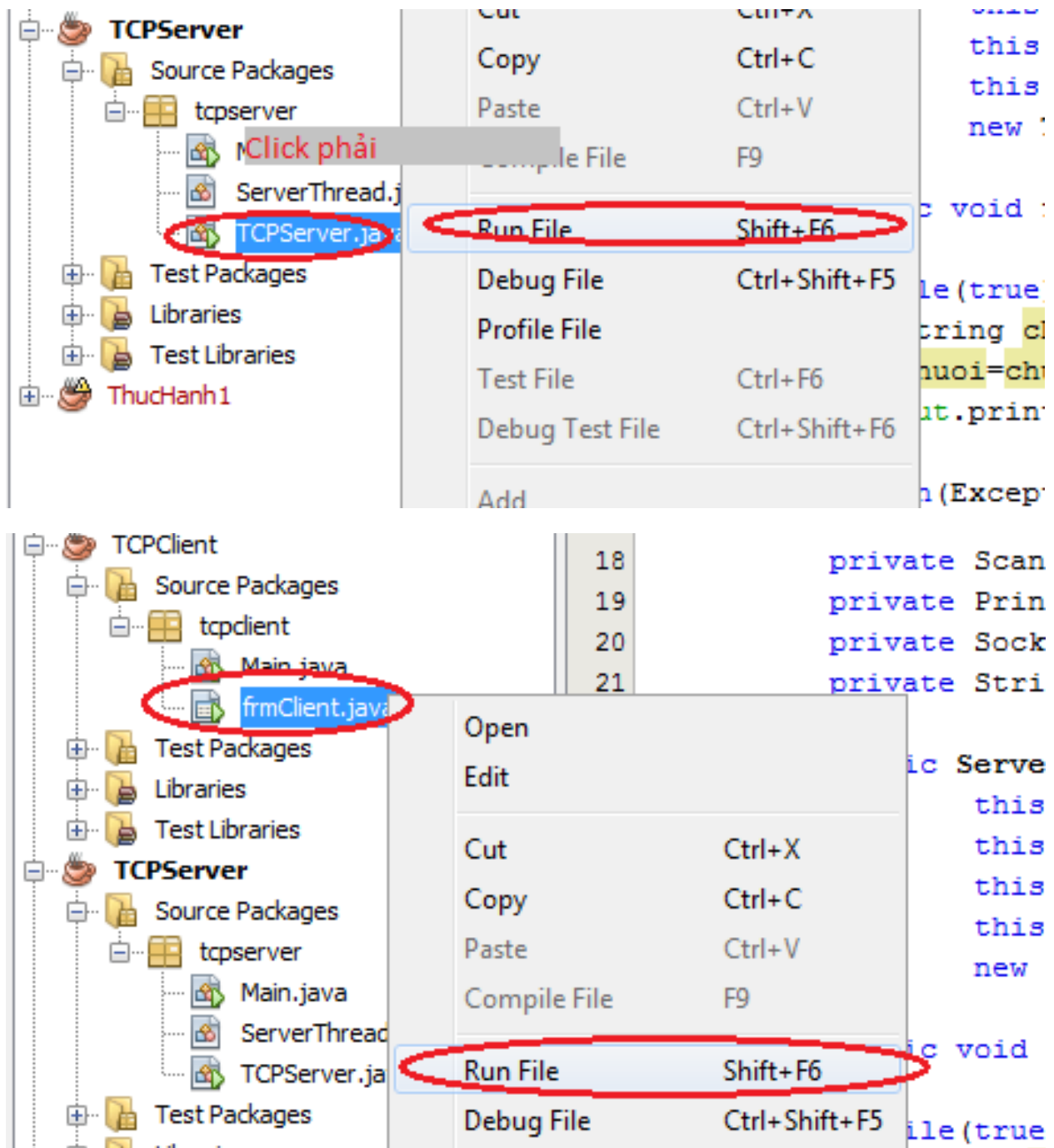
Bước 7: Tạo lớp ServerThread với nội dung như sau.

```

public class ServerThread implements Runnable {
    private Scanner in=null;
    private PrintWriter out=null;
    private Socket socket;
    private String name;

    public ServerThread(Socket socket,String name) throws IOException{
        this.socket=socket;
        this.name=name;
        this.in=new Scanner(this.socket.getInputStream());
        this.out=new PrintWriter(this.socket.getOutputStream(),true);
        new Thread(this).start();
    }
    public void run(){
        try{
            while(true){
                String chuoi=in.nextLine().trim();
                chuoi=chuoi.toUpperCase();
                out.println(chuoi);
            }
        }catch(Exception e){
            System.out.println(name+" has departed");
        }finally{
            try{socket.close();}catch(IOException e){}
        }
    }
}

```

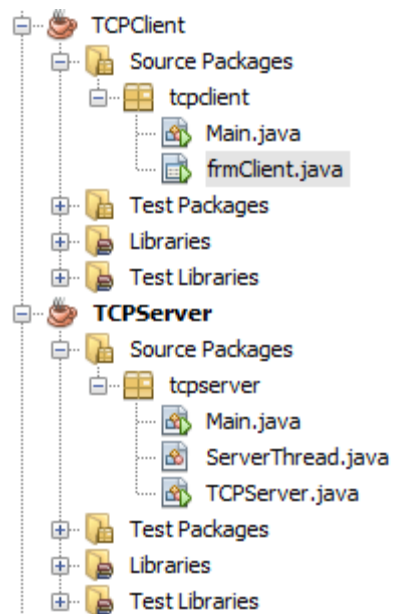
Bước 8: Chạy thử ứng dụng.

Bài 2: Viết chương trình giao tiếp giữa client và server sử dụng giao thức TCP, thực hiện các chức năng sau:

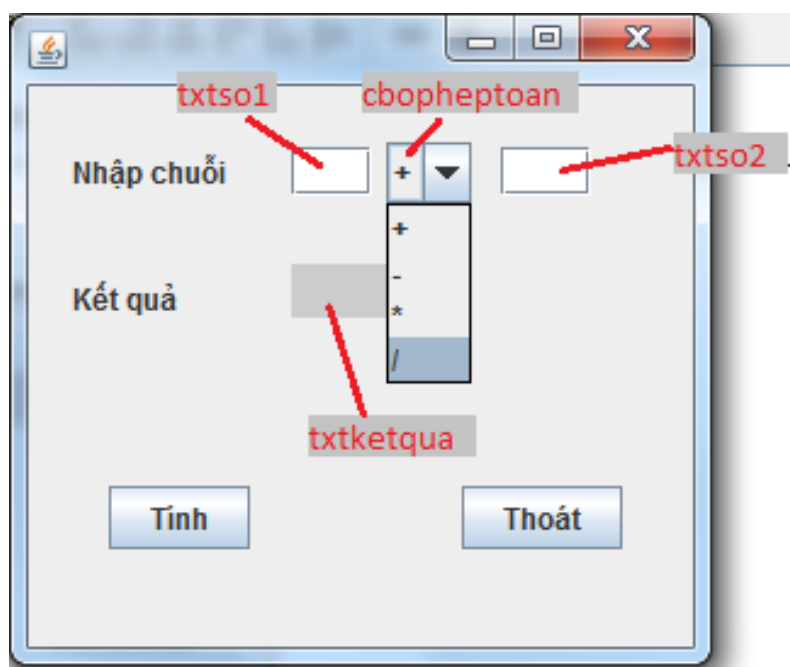
- Client truyền hai số nguyên và phép toán (cộng, trừ, nhân, chia) lên server, server sau khi nhận được thì thực hiện phép toán giữa hai số nguyên và trả kết quả về cho client.
- Client nhận lại kết quả và xuất ra màn hình.

Yêu cầu: Sử dụng Multithread để server có thể giao tiếp được với nhiều client cùng lúc.

Bước 1: Tạo 2 project mới là TCPServer và TCPClient



Bước 2: Tạo Form frmClient có giao diện như sau:



Bước 3: Thêm khai báo vào đầu lớp frmClient

```
package tcpclient;

import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

/**
 *
 * @author Tuan
 */
public class frmClient extends javax.swing.JFrame {
    private Socket socket=null;
    private PrintWriter out=null;
    private Scanner in=null;
    /** Creates new form frmClient */
    public frmClient() {
        initComponents();
        txtchuoi.requestFocus();
    }
}
```

Bước 4: Xử lý sự kiện cho button Tính.

```
private void btntinhActionPerformed(java.awt.event.ActionEvent evt) {
    int so1 =Integer.parseInt(txtso1.getText()); //Lấy số hạng thứ nhất
    int so2 =Integer.parseInt(txtso2.getText()); //Lấy số hạng thứ hai
    String pheptoan=cbopheptoan.getSelectedItem().toString(); //Lấy phép toán
    String chuoi=so1+"@"+pheptoan+"@"+so2; //Ghép thành chuỗi phân cách nhau bởi dấu @
    String ketqua="";
    try{
        //Socket nhận tham số là địa chỉ Host và port
        socket=new Socket("127.0.0.1",1234);
        out=new PrintWriter(socket.getOutputStream(),true);
        in=new Scanner(socket.getInputStream());
        out.println(chuoi); //truyền chuỗi lên server để xử lý
        ketqua=in.nextLine().trim(); //nhận chuỗi kết quả từ server
        txtketqua.setText(ketqua); //Hiển thị chuỗi nhận được từ server lên TextField
        socket.close(); //Đóng Socket
    }catch(Exception e)
    {
        try{if(socket!=null)socket.close();}catch(Exception ex){e.printStackTrace();}
        e.printStackTrace();
    }
}
```

Bước 5: Xử lý sự kiện cho button Thoát

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

Bước 6: Tạo lớp TCPServer như sau:

```
public class TCPServer {
    static final int PORT = 1234;
    private ServerSocket server=null;
    public TCPServer() {
        try{
            server=new ServerSocket(PORT);
        }catch(Exception e){e.printStackTrace();}
    }
    public void action(){
        Socket socket=null;
        int i=0;
        System.out.println("Serverlistening...");
        try{
            while((socket=server.accept())!=null){
                new ServerThread(socket,"Client#"+i);
                System.out.printf("Thread for Client#%d generating...\n",i++);
            }
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public static void main(String[]args){
        new TCPServer().action();
    }
}
```

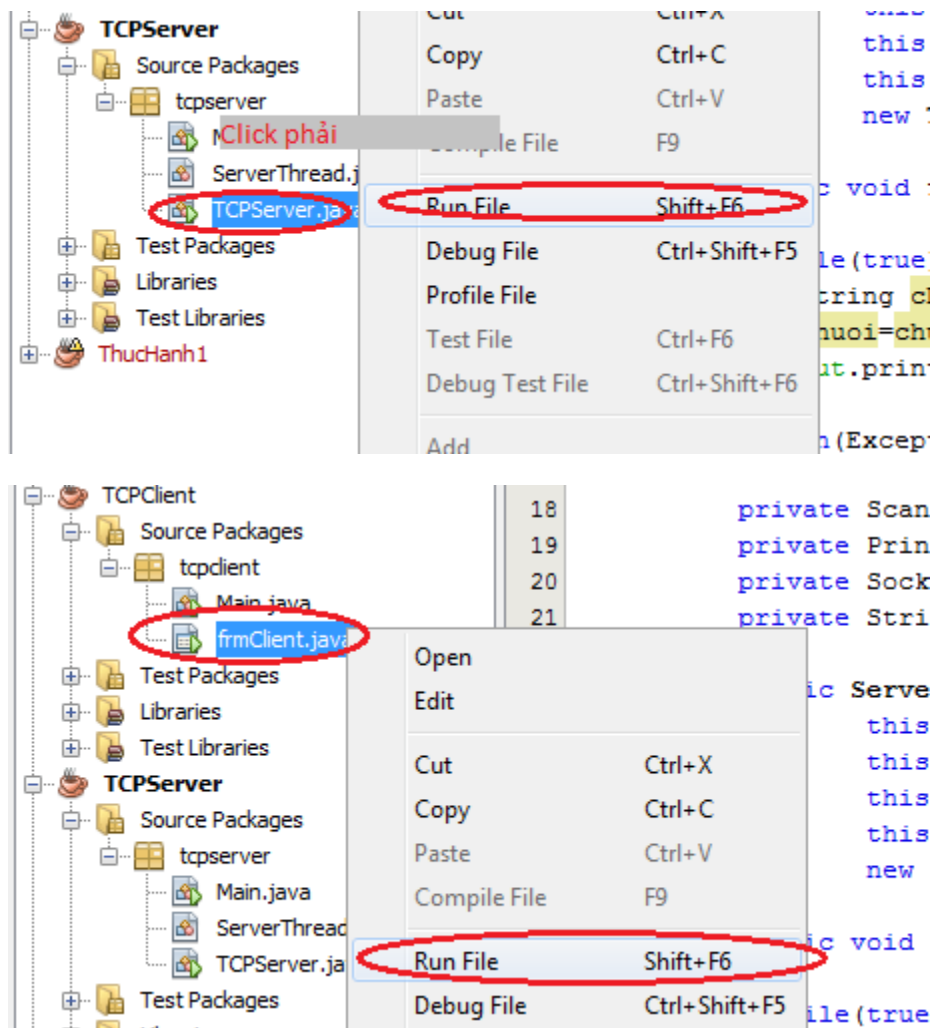
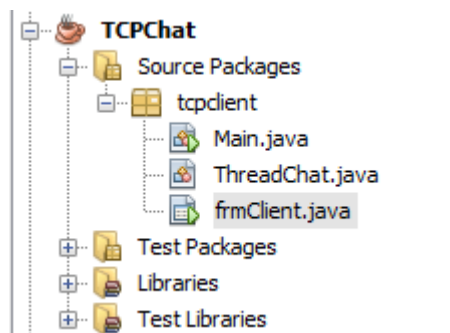
Bước 7: Tạo lớp ServerThread với nội dung như sau.

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

/**
 *
 * @author Tuan
 */
public class ServerThread implements Runnable {
    private Scanner in=null;
    private PrintWriter out=null;
    private Socket socket;
    private String name;

    public ServerThread(Socket socket,String name) throws IOException{
        this.socket=socket;
        this.name=name;
        this.in=new Scanner(this.socket.getInputStream());
        this.out=new PrintWriter(this.socket.getOutputStream(),true);
        new Thread(this).start();
    }
    public void run(){
        try{
            while(true){
                String chuoi=in.nextLine().trim();
                Scanner sc=new Scanner(chuoi);
                sc.useDelimiter("@");
                int so1=sc.nextInt();
                String pheptoan=sc.next();
                int so2=sc.nextInt();

                if(pheptoan.equals("+"))
                    out.println(so1+so2);
                else if(pheptoan.equals("-"))
                    out.println(so1-so2);
                else if(pheptoan.equals("*"))
                    out.println(so1*so2);
                else if(pheptoan.equals("/"))
                    out.println((float)so1/so2);
            }
        }catch(Exception e){
            System.out.println(name+" has departed");
        }finally{
            try{socket.close();}catch(IOException e){}
        }
    }
}
```


Bước 8: Chạy thử ứng dụng.**Bài 3: Viết chương trình cho phép hai máy chat với nhau.****Bước 1: Tạo Project mới TCPChat**

Bước 2: Tạo Form frmClient có giao diện như sau:

Bước 3: Thêm các khai báo cho frmClient

```
package tcpclient;

import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

/**
 *
 * @author Tuan
 */
public class frmClient extends javax.swing.JFrame {
    private Socket socket=null;
    private PrintWriter out=null;
    private Scanner in=null;
    /** Creates new form frmClient */
    public frmClient() {
        initComponents();
        txtHost.requestFocus();
        ThreadChat obj=new ThreadChat();
        obj.chat=this;
    }
}
```

Bước 4: Thêm sự kiện cho button Thoát, Viết thêm 1 hàm Hienthi

```

private void btnthoatActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}

public void Hienthi(String str){
    txtchat.append(str);
}

```

Thêm sự kiện cho button Thoát

Viết thêm hàm Hienthi để phục vụ cho việc nhận tin nhắn từ máy khác đến.

Bước 5: Thêm sự kiện cho button Send

```

private void btnsendActionPerformed(java.awt.event.ActionEvent evt) {
    String chuoi=txtsend.getText();
    String nick=txtNick.getText();
    String host=txtHost.getText();
    try{
        //Socket nhận tham số là địa chỉ Host và port
        socket=new Socket(host,1234);
        out=new PrintWriter(socket.getOutputStream(),true);
        in=new Scanner(socket.getInputStream());
        out.println(nick+": "+chuoi+"\n");//truyền chuỗi lên server để xử lý
        txtchat.append(nick+": "+chuoi+"\n");
        socket.close();//Đóng Socket
    }catch(Exception e)
    {
        try{if(socket!=null)socket.close();}catch(Exception ex){e.printStackTrace();}
        e.printStackTrace();
    }
}

```

Bước 6: Tạo thêm lớp ThreadChat.java

```

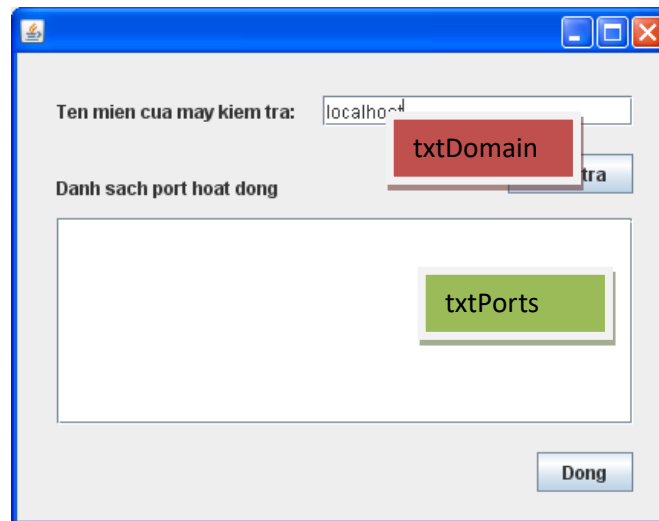
public class ThreadChat implements Runnable {
    private Scanner in=null;
    private Socket socket=null;
    public frmClient chat=null;
    ServerSocket server=null;
    public ThreadChat(){
        try{
            server=new ServerSocket(1234);//Tạo ra Server socket mới nhận port 1234 làm tham số.
        }catch(Exception e){e.printStackTrace();}
        new Thread(this).start();//Khởi động Thread
    }
    public void run(){
        try{
            while(true){
                while((socket=server.accept())!=null){//Nhận kết nối từ máy khác đến
                    this.in=new Scanner(this.socket.getInputStream());
                    String chuoi=in.nextLine().trim();//Nhận chuỗi
                    chat.Hienthi(chuoi+"\n");//Hiển thị chuỗi ra màn hình
                }
            }
        }catch(Exception e){}
        finally{
            try{socket.close();}catch(IOException e){}
        }
    }
}

```

Bước 7: Chạy thử chương trình trên nhiều máy.

- ✓ Mở chạy formClient.
- ✓ Nhập địa chỉ máy cần chat, nick của mình.
- ✓ Mở chạy frmClient trên máy khác và nhập địa chỉ máy cần chat.

Bài 4. Viết chương trình kiểm tra các cổng từ 1024 đến 65536 (hoặc từ 0 – 1024) của Server bất kỳ, có cổng nào hoạt động không?

Hướng dẫn:**Viết code chọn nút kiểm tra:**

```
Socket s;  
for(int port=1024;port<65536;port++)  
    try {  
        s = new Socket(this.txtDomain.getText(), port);  
        this.txtPorts.setText(this.txtPorts.getText()+ "\n" + port);  
    } catch (UnknownHostException ex) {  
        JOptionPane.showMessageDialog(this,ex);  
    } catch (IOException e) {  
        //JOptionPane.showMessageDialog(this,e);  
    }  
}
```

Bài 5. (Tham khảo thêm) Viết chương trình đọc thông tin Socket tại máy cục bộ và thông tin Socket của máy kết nối trên cổng 80.

Bài 6. (Tham khảo thêm) Viết chương trình thực hiện mô phỏng giao thức FTP như sau:

Chương trình client: thực hiện chức năng gửi yêu cầu đường dẫn tên tập tin tới Server sau đó đợi đáp trả từ server dữ liệu liên quan đến tập tin và hiển thị thông tin dữ liệu tập tin lên màn hình và lưu thành tập tin phía máy client nếu tìm thấy ngoài ra hiển thị chuỗi lỗi.

Chương trình server: có nhiệm vụ lắng nghe kết nối từ client, tìm và mở tập tin liên quan theo yêu cầu của client viết về cho client sau đó đóng luồng lại.

Hướng dẫn: *Viết chương trình FTP Server trước.*

Bước 1: Viết chương trình quản lý user như ở bài tập phần 1. Cho phép thêm user mới vào cơ sở dữ liệu user.

Bước 2: Viết code cho FTP lắng nghe giao tiếp với client, tuần tự tại mỗi thời điểm xử lý cho 1 client.

```
package ftpservervd;
import java.io.File;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;
import javax.swing.JOptionPane;
public class Main {
    //khai bao cac hang trong giao thuc giao tiep
    public static final int DANGNHAP=1;
    public static final int KHONGLALENH=0;
    public static final int DANGNHAPKHONGTHANHCONG=0;
    public static final int DANGNHAPTHANHCONG=1;
    public static final int THOAT=2;
    //ham doi chuoai giao tiep thanh hang cho de xu ly
    public static int laLenh(String cmd){
        if(cmd.equals("DANGNHAP"))
            return DANGNHAP;
        return KHONGLALENH;
    }
    //thiet lap port giao tiep cua ung dung, FTP co port la 20 va 21
    //vi du chon port 10000
    public static final int PORT = 10000;
    public static void main(String[] args) {
        //gia su co user, pass, path
        //sinh vien thay 1 user nay bang cach truy xuat co so du lieu
        // cac user da tao ra o buoc 1 do chuong trinh quan ly user
        String userA="tu";
        String passA="tu";
        String path="C:/";
        ServerSocket s;
        try {
            s = new ServerSocket(PORT);
            while (true) {
                Socket new_s = s.accept();
                //nhan lenh giao tiep tu client
                boolean lap=true;
                while(lap)
                {
                    String cmd;
                    Scanner sc=new Scanner(new_s.getInputStream());
                    cmd=sc.nextLine();
                    //dieu phoi su kien yeu cau o phai client
                    switch (laLenh(cmd)){
                        case DANGNHAP:
                            String user=sc.nextLine();
                            String pass=sc.nextLine();
                            PrintWriter pw;
                            pw=new PrintWriter(new_s.getOutputStream());
                            if(user.equals(userA) && pass.equals(passA)){
                                pw.println(DANGNHAPTHANHCONG);
                                //mo thu muc len goi ve cho client
                                File dir=new File(path);
                                File dsFile[]=dir.listFiles();
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

        if(dsFile==null)
        {
            JOptionPane.showMessageDialog(null, "Đường dẫn không đúng hay
không phải thư mục!");
        }else{
            pw.println(dsFile.length);
            for(int i=0;i<dsFile.length;i++)
            pw.println(dsFile[i].getName());
        }
    }else
    {
        //goi ve khong mo duoc
        pw.println(DANGNHAPKHONGTHANHCONG);
        pw.println("Dang nhap ko thanh cong");
    }
    pw.flush();
    break;
case THOAT:
    lap=false;
    break;
}
}
new_s.close();
}
} catch (Exception ex) {
    System.out.println(ex);
}
}
}

```

Bước 3: Tạo giao diện chương trình client truy cập đến server

Bước 4: Viết code cho nút login

```
Socket s;  
public static final int PORT = 10000;  
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    //tao socket  
    String domain=this.txtDomain.getText();  
    try {  
        InetAddress ia = InetAddress.getByName(domain);  
        try {  
            s = new Socket(ia, PORT);  
            //goi user, pass len server  
            PrintWriter pw=new PrintWriter(s.getOutputStream());  
            //lay du lieu tu form do su dung go vao  
            String user=this.txtUser.getText();  
            String pass=this.txtPass.getText();  
            String cmd="DANGNHAP";  
            pw.println(cmd);  
            pw.println(user);  
            pw.println(pass);  
            System.out.println(cmd);  
            System.out.println(user);  
            System.out.println(pass);  
            pw.flush();  
            //client doi phan hoi tu server  
            Scanner sc=new Scanner(s.getInputStream());  
            int cmdR=sc.nextInt();  
            if(cmdR==1){  
                JOptionPane.showMessageDialog(this, "Dang nhap thanh cong");  
                //luu duong dan hien tai ma list hien thi danh tap tin trong folder do.  
                DefaultListModel dm=new DefaultListModel();  
                int n=sc.nextInt();  
                for(int i=0;i<n;i++){  
                    dm.addElement(sc.nextLine());  
                }  
                this.lstPath.setModel(dm);  
            }else  
                JOptionPane.showMessageDialog(this, "Dang nhap khong thanh cong");  
        } catch (IOException ex) {  
            JOptionPane.showMessageDialog(this, ex.toString());  
        }  
    } catch (UnknownHostException ex) {  
        JOptionPane.showMessageDialog(this, ex.toString());  
    }  
}
```

Bước 5: viết code cho nút Browse

```
String path;  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    fchPath.setVisible(true);  
    //thiet lap  
    fchPath.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);  
    //kiem tra form nay duoc chon nut gi?  
    if(fchPath.showOpenDialog(this)==JFileChooser.APPROVE_OPTION)  
        try{  
            path=fchPath.getSelectedFile().getCanonicalPath();  
            File dir=new File(path);  
            File dsFile[]=dir.listFiles();  
            if(dsFile==null)  
            {  
                JOptionPane.showMessageDialog(null, "sai duong dan!");  
            }else{  
                try{  
                    //luu duong dan hien tai ma list hien thi danh tap  
                    // tin trong folder do.  
                }  
            }  
        }  
    }
```

```

        DefaultListModel dm=new DefaultListModel();
        for(int i=0;i<dsFile.length;i++){
            String filename=dsFile[i].getName();
            dm.addElement(filename);
        }
        this.lstClientPath.setModel(dm);
    }catch(Exception e)
    {JOptionPane.showMessageDialog(null,e.toString());}
}
}catch(Exception e)
{ }
}

```

Bước 6: Viết code cho nút upload

```

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:
    String fileName=(String) this.lstClientPath.getSelectedValue();
    String cpath=path+"\\\" + fileName;
    System.out.println(cpath);
    try{
        PrintWriter pw=new PrintWriter(s.getOutputStream());
        //gọi tin hiệu lệnh
        pw.println("UPLOAD");
        pw.flush();
        System.out.println("Đã gọi lệnh upload lên server");
        pw.println(fileName);
        pw.flush();
        System.out.println("Đã gọi tên tập tin lên server");
        BufferedOutputStream bos=new BufferedOutputStream(s.getOutputStream());
        DataOutputStream dos=new DataOutputStream(bos);
        //mở tập tin ra
        BufferedInputStream bis=new BufferedInputStream (new FileInputStream(cpath));
        //lấy nội dung tập tin và gửi lên server
        byte buf[]=new byte[bis.available()];
        //tạo bộ đệm đọc hết dữ liệu từ tập tin vào bộ đệm rồi đây
        //vào luồng lên server.
        bos.write(bis.read(buf));
        System.out.println("Đã gửi dữ liệu tập tin lên server");
        bos.flush();
        //bis.close();
        //đổi danh sách tập tin của folder ở server với tình trạng mới
        Scanner sc=new Scanner(s.getInputStream());
        String cmd=sc.nextLine();
        System.out.println("Đã nhận đáp trả từ server");
        if(cmd.equals("DANHSA"))
            JOptionPane.showMessageDialog(null, "Đã gửi tập tin thành công");
        else
            JOptionPane.showMessageDialog(null, "Gửi tập tin thất bại");
        //nhận update
        updateFolderServer();
    }catch(Exception e){
        System.out.println(e);
    }
}

```

Bước 7: Hàm cập nhật danh sách tập tin trong Folder của server sau khi upload

```

public void updateFolderServer(){
    try{
        BufferedInputStream bi=new BufferedInputStream(s.getInputStream());
        Scanner sc=new Scanner(bi);
        DefaultListModel dm=new DefaultListModel();
    }
}

```



```
//server gọi về số lượng tập tin thu được sau khi upload
int n=sc.nextInt();
System.out.println("Đã nhận được số lượng tập tin gọi từ server");
//nhận tên từng tập tin thu được
for(int i=0;i<n;i++)
{
    String filename=sc.nextLine();
    dm.addElement(filename);
}
System.out.println("Đã hiển thị xong danh sách tập tin");
//hiển thị lên form
this.lstPath.setModel(dm);
//vẽ lại giao diện
this.validate();
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Lỗi", e.toString(), JOptionPane.ERROR_MESSAGE);
}
}
```

Bước 8: Viết code server đáp trả lại client sau khi nhận tập tin upload thành công

Định nghĩa thêm hằng phía server.

```
public static final int UPLOAD=3;
```

Cập nhật lại hàm ánh xạ lệnh dạng chuỗi sang số.

```
public static int laLenh(String cmd) {
    if(cmd.equals("DANGNHAP"))
        return DANGNHAP;
    if(cmd.equals("UPLOAD"))
        return UPLOAD;
    return KHONGLALENH;
}
```

Bổ sung vào lệnh xử lý switch của server.

```
case UPLOAD:
System.out.println("Đã vào lệnh upload");
    String fileName=sc.nextLine();
    System.out.println("Đã lấy tên tập tin");
    try{
        String path2;
        //kiểm tra chuỗi đường dẫn có dấu / cuối cùng hay không?
        //và gán tên tập tin từ client vào tương ứng
        if (path.lastIndexOf("/")>=path.length()-1)
            path2=path + fileName;
        else
            path2=path + "/" + fileName;
        System.out.println(path2);
        FileOutputStream fos=new FileOutputStream(new File(path2));
        BufferedOutputStream bos=new BufferedOutputStream(fos);
        BufferedInputStream bis;
        bis=new BufferedInputStream(new_s.getInputStream());
        byte buf[]=new byte[bis.available()];
```

```
bos.write(bis.read(buf));
bos.flush();
bos.close();

pw=new PrintWriter(new_s.getOutputStream());
pw.println("DANHANH");
pw.flush();

//yeu cau update lai listbox o server
//mo thu muc ra va tra ve noi dung thu muc o phia server
Main.traThuMucClient(path,pw);
}
catch(Exception e)
{
    System.out.println(e);
}

break;
```

Đoạn code gửi danh sách tập tin của thư mục server sau khi upload.

```
static void traThuMucClient(String path, PrintWriter out) {
    try{
        File dir=new File(path);
        File dsFile[];
        System.out.println("Dang doc tap tin");
        try{
            dsFile = dir.listFiles();
            System.out.println("da la ds tap tin");
            out.println(dsFile.length);
            for(int i=0;i<dsFile.length;i++){
                String filename=dsFile[i].getName();
                out.println(filename);
            }
            out.flush();
            System.out.println("da goi client");
        }catch(Exception e)
        {JOptionPane.showMessageDialog(null,e.toString());}

    } catch(Exception e){
        System.out.println(e.toString());
    }
}
```

Bước 9: Viết code cho nút download

```
private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    //lay tap tin can download
```

```
String fileName=(String) this.lstPath.getSelectedValue();
System.out.println(fileName);
try{
    PrintWriter pw=new PrintWriter(s.getOutputStream());
    //goi tin hieu lenh
    pw.println("DOWNLOAD");
    pw.flush();
    System.out.println("Da goi lenh download len server");
    pw.println(fileName);
    pw.flush();
    //doi server goi noi dung tap tin ve
    System.out.println("Doi server goi noi dung tap tin ve");
    String cpath=path+"\\\" + fileName;
    FileOutputStream fos;
    fos = new FileOutputStream(new File(cpath));
    BufferedOutputStream bos=new BufferedOutputStream(fos);
    BufferedInputStream bis=new BufferedInputStream(s.getInputStream());
    byte buf[]=new byte[bis.available()];
    bos.write(bis.read(buf));
    bos.flush();
    bos.close();
    pw=new PrintWriter(s.getOutputStream());
    pw.println("DANHAN");
    pw.flush();
    //cap nhat lai thu muc client vua download
    this.capNhatClientFolder(cpath);
} catch (Exception ex) {
    System.out.println(ex);
}

}

private void capNhatClientFolder(String cpath) {
    //mo thu muc voi duong dan path ra
    File dir=new File(path);
    File dsFile[]=dir.listFiles();
    if(dsFile==null)
    {
        JOptionPane.showMessageDialog(null, " Duong dan sai!");
    }else{
        try{
            //luu duong dan hien tai ma list hien thi danh tap tin trong folder do.
            //path=txtPath.getText();

            DefaultListModel dm=new DefaultListModel();
            for(int i=0;i<dsFile.length;i++){
                String filename=dsFile[i].getName();
                dm.addElement(filename);
            }
        }
    }
}
```

```
        }  
        this.lstClientPath.setModel(dm);  
        this.validate();  
    }catch(Exception e){  
        JOptionPane.showMessageDialog(this, e);  
    }  
}  
}
```

Bổ sung hằng ở server:

```
public static final int DOWNLOAD=4;
```

Cập nhật hàm ánh xạ lệnh dạng chuỗi sang số:

```
public static int laLenh(String cmd){  
    if(cmd.equals("DANGNHAP"))  
        return DANGNHAP;  
    if(cmd.equals("UPLOAD"))  
        return UPLOAD;  
    if(cmd.equals("DOWNLOAD"))  
        return DOWNLOAD;  
    return KHONGLALENH;  
}
```

Bổ sung trường hợp download cho lệnh switch:

```
case DOWNLOAD:  
    //lay ten tap tin do client goi len  
    System.out.println("Da vao lenh download");  
    String fileNameD=sc.nextLine();  
    System.out.println("Da lay ten tap tin");  
    try{  
        String cpath;  
        //kiem tra chuoi duong dan co dau / cuoi cung hay ko?  
        //va gan ten tap tin tu client vao tuong ung  
        if (path.lastIndexOf("/")>=path.length()-1)  
            cpath=path + fileNameD;  
        else  
            cpath=path + "/" + fileNameD;  
        System.out.println(cpath);  
        //mo tap tin ra  
        BufferedInputStream bis;  
        bis=new BufferedInputStream (new FileInputStream(cpath));  
        //lap doc noi dung tap tin va goi lieu len server  
        byte buf[]=new byte[bis.available()];
```

```
//tao bo dem doc het du lieu tu tap tin vao bo dem roi day
//vao luong len server.
BufferedOutputStream bos;
bos=new BufferedOutputStream(new_s.getOutputStream());
bos.write(bis.read(buf));
System.out.println("da goi du lieu tap tin ve cho client");
bos.flush();

//doi nhan danh sach tap thu cua folder o server voi tinh trang moi
Scanner scRequest=new Scanner(new_s.getInputStream());
String cmdRequest=scRequest.nextLine();
System.out.println("da nhan dap tra tu server");
if(cmdRequest.equals("DANHANH"))
    System.out.println("Đã gửi tập tin thành công");
else
    System.out.println("Gửi tập tin thất bại");
}catch(Exception e){
    System.out.println(e);
}
break;
```

Bài 7. Viết chương trình cho phép hai máy chat với nhau.

Bài 8. Viết chương trình client/server cho phép các máy chat với nhau trên mạng. Trong đó, một máy đóng vai trò server quản lý thông tin người dùng đăng nhập và cho phép hai user ở hai máy khác nhau chat với nhau.

Bài 9. Viết chương trình chơi game Caro trực tuyến trên mạng (Yêu cầu viết chỉ cho 2 người chơi với nhau - Một phần mềm client để gọi yêu cầu chơi tới Server và một phần mềm Server đợi kết nối từ client và thực hiện kênh trao đổi chơi với một client).

Game gồm 2 người chơi, một trắng và một đen. Mỗi người chơi ở hai máy khác nhau. Mỗi người chơi lần lượt chọn con cờ đặt lên bàn cờ của mình và bàn cờ cũng của người chơi thứ 2 cũng được cập nhật theo.

Người chơi trắng nếu anh ta có 5 quân cờ thẳng hàng ngang, dọc hoặc cắt chéo.

OCo: JButton
<ul style="list-style-type: none">- int d;- int c;- int trangThai;- FrameCaroClient fcc;
<ul style="list-style-type: none">+ OCo(int, int, int);+ capNhatTrangThai(int);+ xuLyNutDuocNhap();

Xử lý nút nhật xảy ra khi người chơi nhấp vào button này. Khi đó, hàm này sẽ thực hiện các chức năng như: cập nhật trạng thái bàn cờ vị trí có tọa độ d, c. Sau đó, gọi hàm cập nhật trạng thái của đối tượng bàn cờ. Cập nhật giao diện phía client. Tạo kết nối tới Server. Gửi ID cặp người chơi và đối tượng bàn cờ tới Server.

NguoiChoi
<ul style="list-style-type: none">- String ten;- String pass;- int trangThai; //chơi, hoặc không chơi- boolean choiCo; //chưa chơi cờ hoặc đã chơi
<ul style="list-style-type: none">+ NguoiChoi(String, String);+ capNhatTrangThai(int);+ capNhatChoiCo(boolean);

Lớp Frame sẽ hiển thị giao diện cho người sử dụng chơi và có nhiệm vụ kết nối tới server để đăng ký chơi. Mỗi lần có một quân cờ được đặt, thì lớp này sẽ phản ứng như, cập nhật trạng thái bàn cờ, kết nối tới server, gửi thông tin bàn cờ tới server, cho server cập nhật bàn cờ của cặp người chơi có số hiệu ID. Đồng thời, phía client cũng cập nhật lại trạng thái bàn cờ cho người dùng xem.

BanCo`//3: ô trống, 0: ô trắng, 1: ô đen;``- int oCo[][];``- int n; //kích thước``- NgnoiChoi nc1, nc2;``- boolean trangThai; //thắng true,
thua false``+ BanCo(int);``+ OCo(int d, int c):int;``+ NC1(): NgnoiChui;``+ NC2(): NgnoiChui;``+ TrangThai():boolean``+ capNhatTrangThai();`


```
FrameCaroClient: JFrame
```

```
//phan du lieu xu ly
```

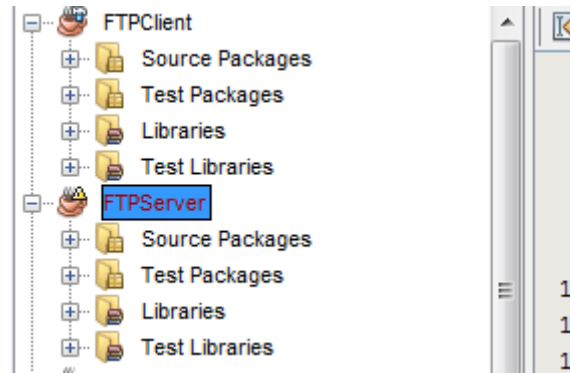
```
- BanCo bc;  
- OCo oCo[][];  
- JPanel PBanThongTin; //hiển thị thông tin  
  người chơi  
- JPanel PBanCo; //hiển thị thông tin bàn cờ  
- JBarMenu bm;  
- JMenu mDangky;  
- JMenuItem miDangKy;  
- Socket s;      //kết nối tới Server  
- String domain;  
- int port;
```

```
+ FrameCaroClient(BanCo) ;  
+ dangKyChoi() ;  
+ chapNhanChoi() ;  
+ napBanCo(BanCo) ;  
+ luuServerNgungChoi() ;
```

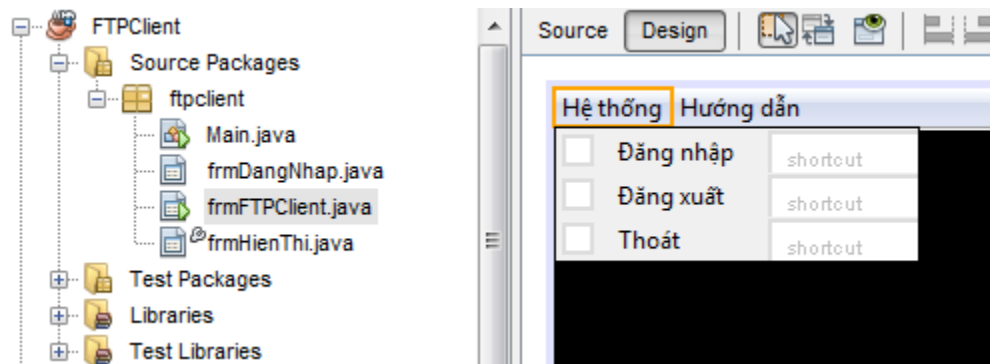
Bài 10. Viết chương trình FTP xử lý đa tiến trình.

Hướng dẫn: Tương tự bài 6, nhưng bổ sung thêm việc xử lý đa tiến trình ở Server và xử lý truy xuất hệ quản trị cơ sở dữ liệu MySQL.

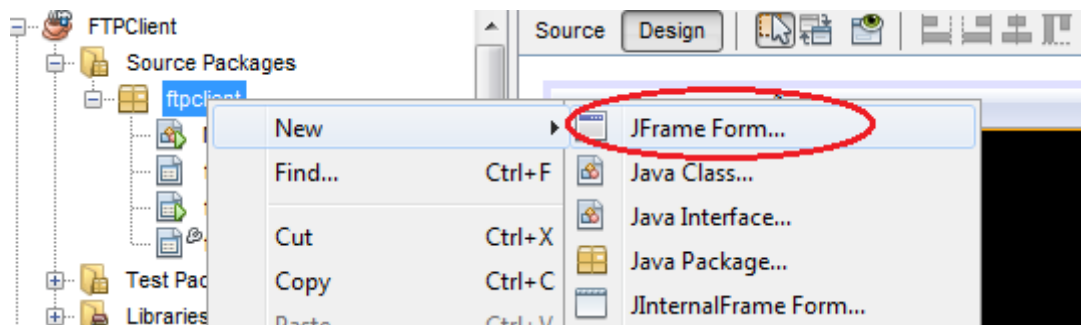
Bước 1: Tạo ra 2 project FTPClient và FTPServer



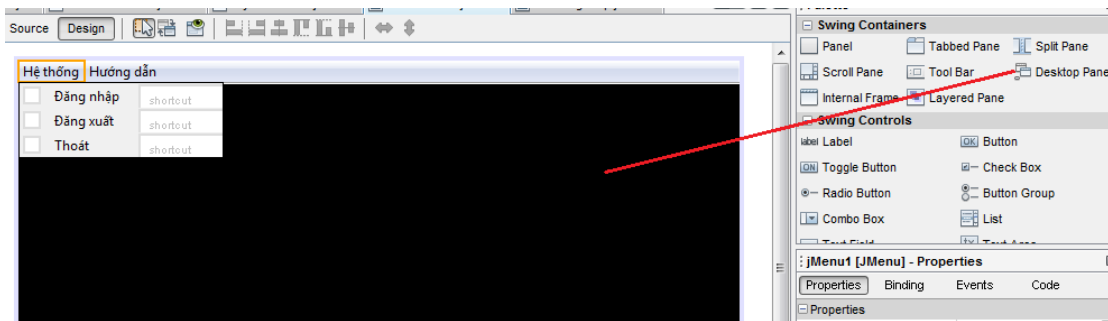
Bước 2: Phía Client tạo ra 3 form: frmDangNhap, frmFTPClient, frmHienThi



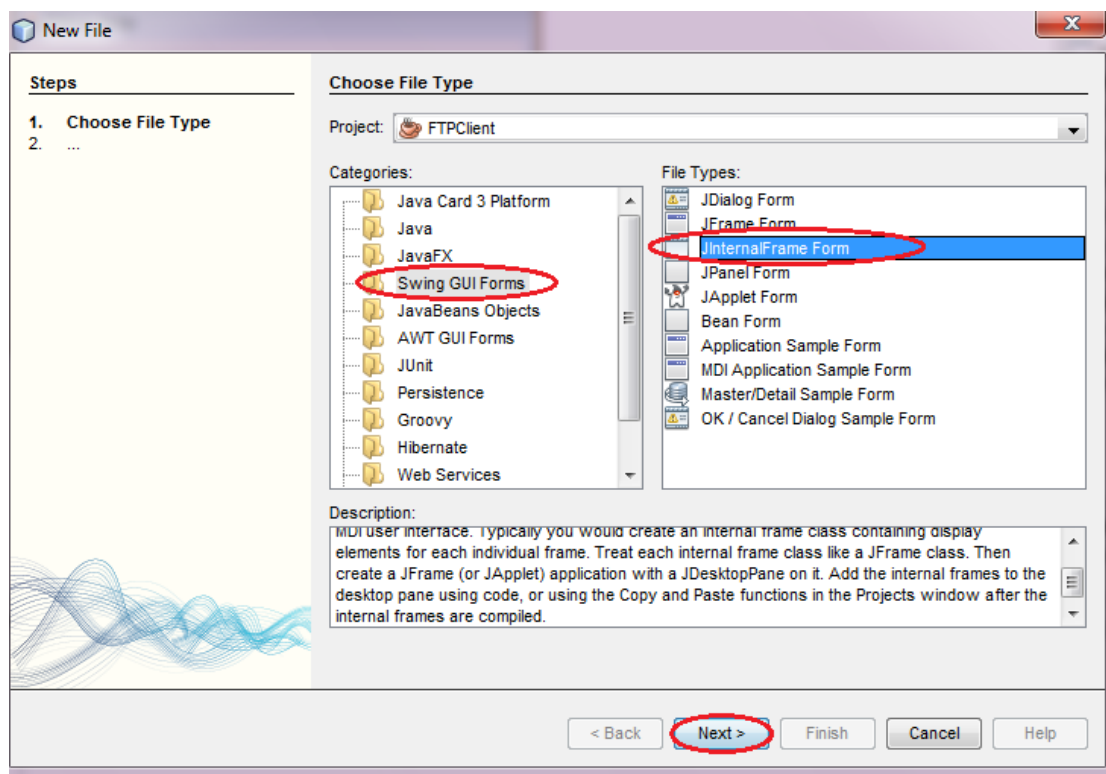
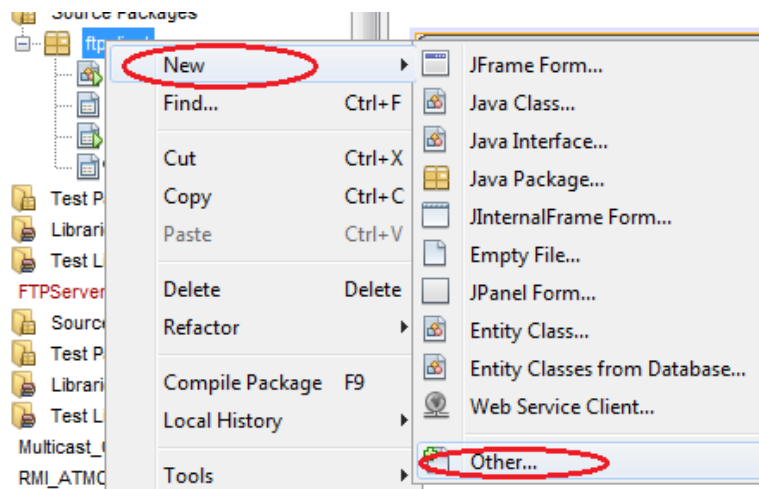
Bước 2.1: Tạo frmFTPClient: kế thừa lớp JFrame Form

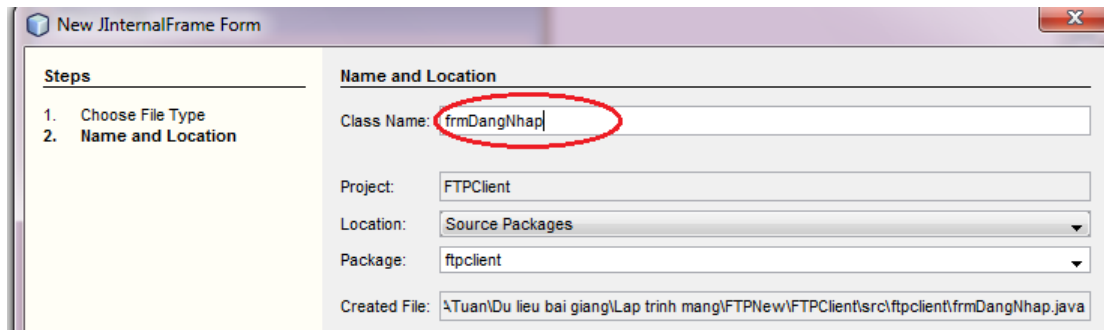


Thêm đối tượng Desktop Pane và đặt tên biến là: dspane

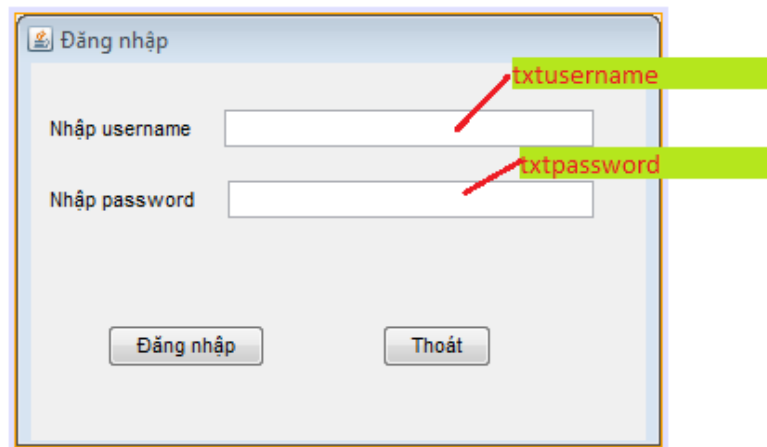


Bước 2.2: Tạo frmDangNhap: kế thừa lớp JInternalFrame

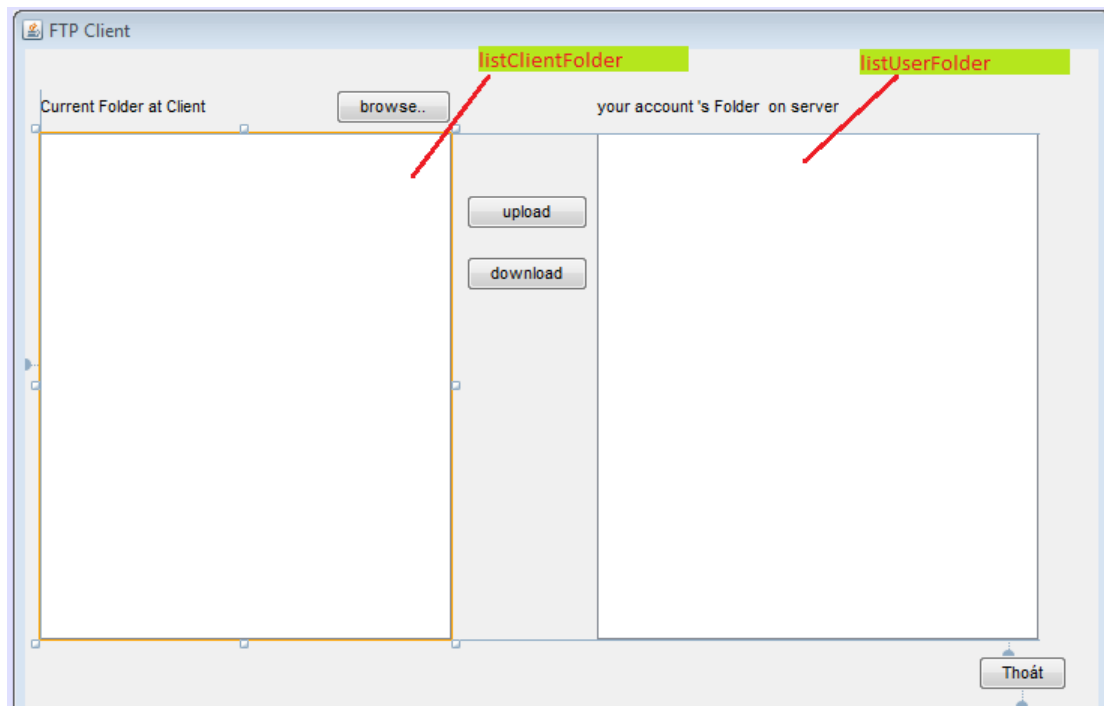




Bước 2.3: Tạo giao diện cho Form đăng nhập như sau



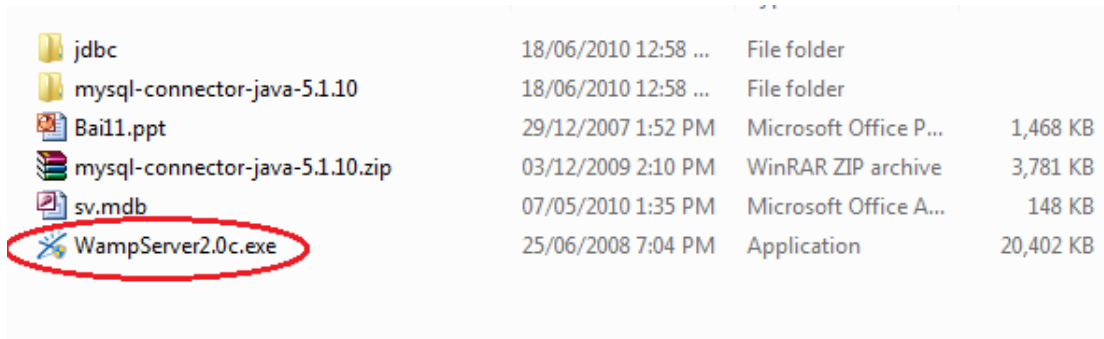
Bước 2.4: Tạo frmHienThi: làm như frmDangNhap, kế thừa lớp JInternalFrame



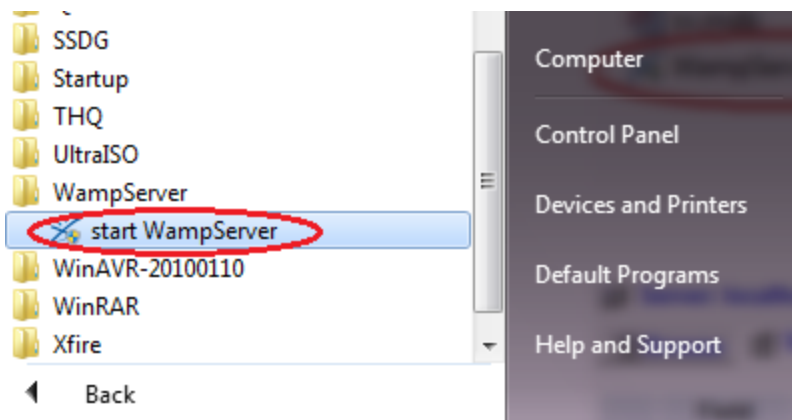
Bước 3: Phía Server tạo ra các lớp: Các lớp giao diện và các lớp thao tác trên CSDL và các đối tượng xử lý dữ liệu của server.

Bước 3.1: Tạo CSDL MySQL

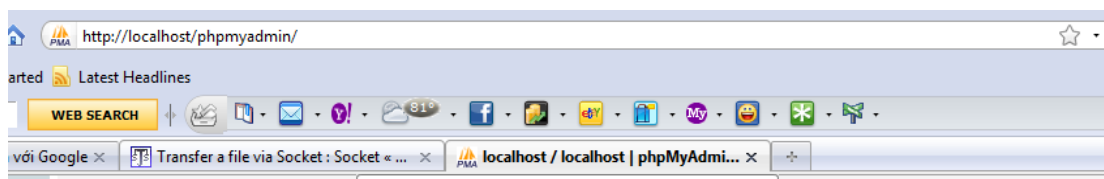
Cài wampServer



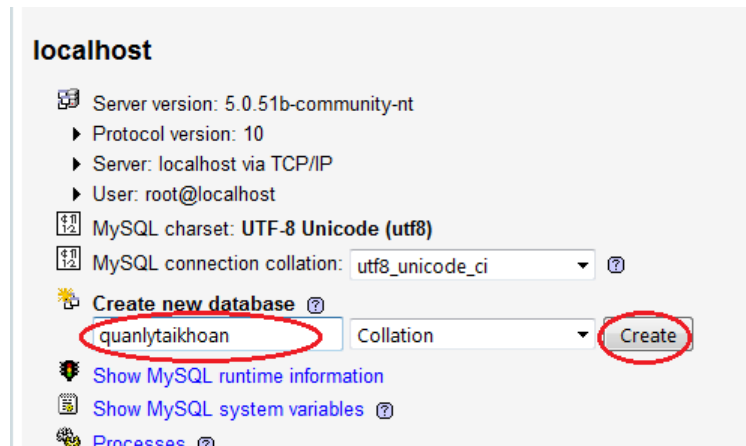
Sau khi cài WampServer ta vào Start→Program file để Start Server



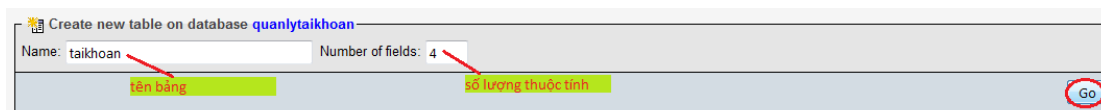
Vào trình duyệt (Firefox) gõ địa chỉ để vào trang phpmyadmin



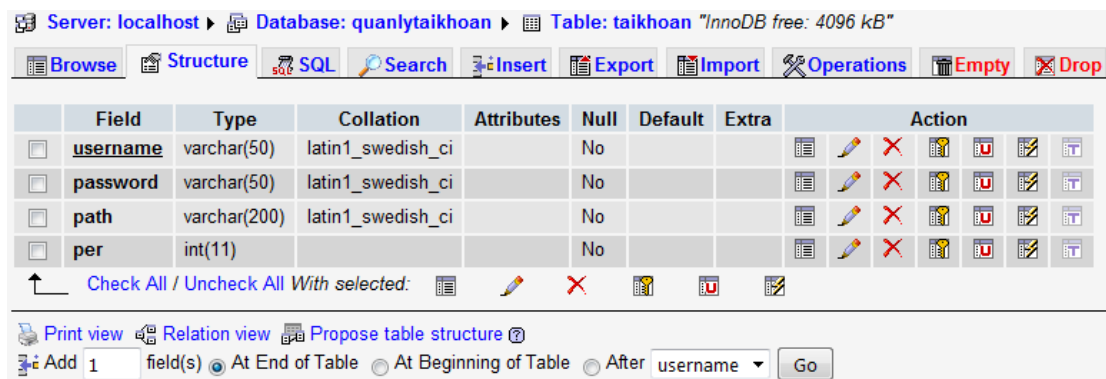
Tạo database quanlytaikhoan



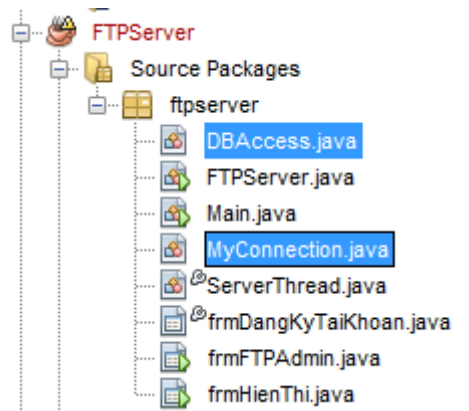
Vào database quanlytaikhoan, Tạo bảng taikhoan



Tạo các thuộc tính cho bảng dữ liệu:

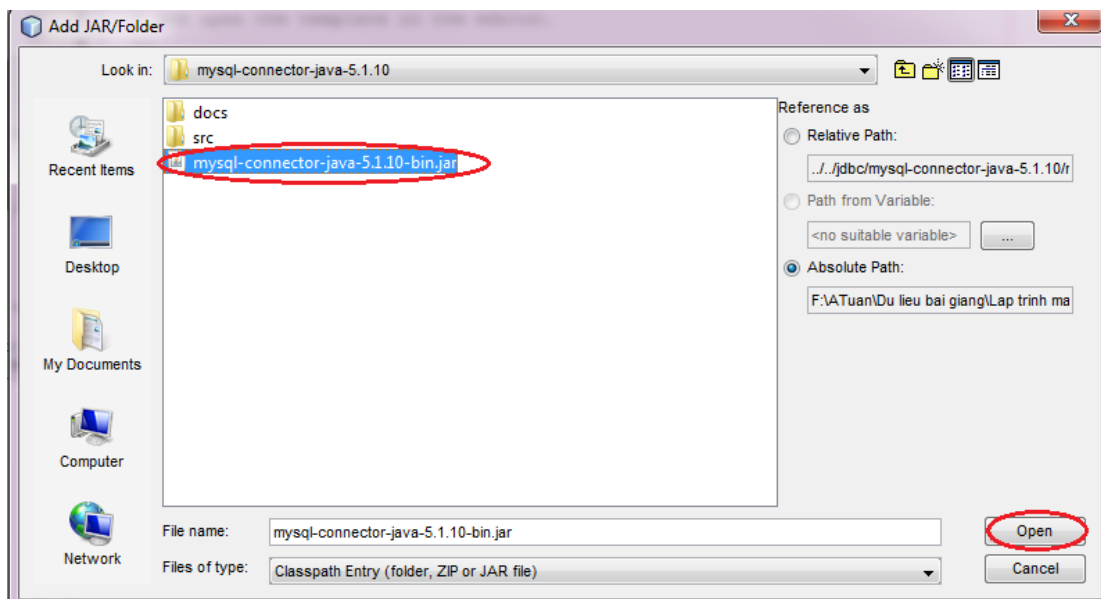
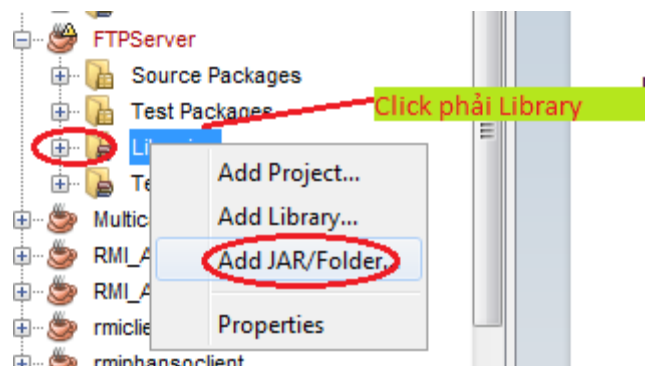


Bước 3.2: Tạo ra các lớp đối tượng thao tác truy xuất CSDL. Chúng ta sẽ thao tác truy xuất trên CSDL thông qua các lớp này. Gồm có 2 lớp là MySqlConnection.java để tạo kết nối và lớp DBAccess.java để truy xuất CSDL (Ta có thể thêm 1 lớp User để việc truy xuất CSDL rõ ràng và chuyên biệt hơn).

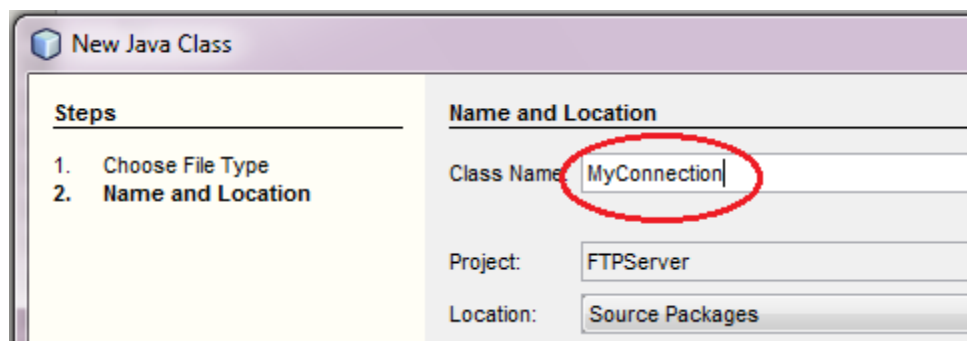
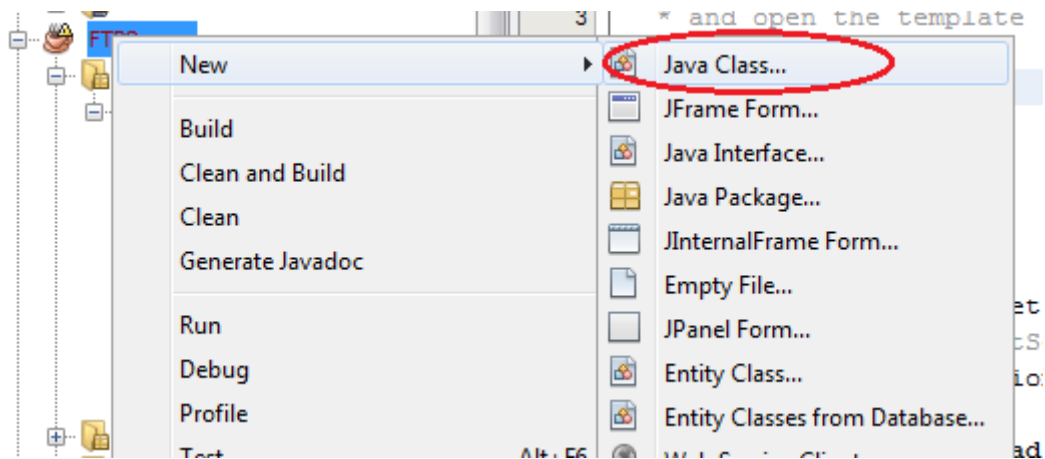


Lưu ý ta phải add thêm driver nếu muốn kết nối CSDL. Ta làm như sau:

Click phải vào Library → Add JAR/Folder.



Tạo lớp MyConnection có nội dung như sau:



```

import java.sql.*;
import javax.swing.*;

public class MyConnection {

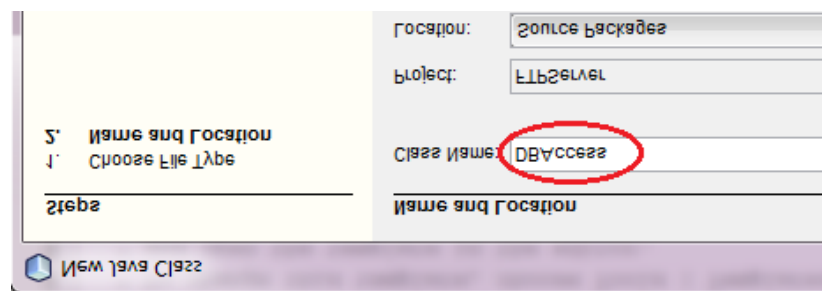
    public Connection getConnection() {
        try {
            Class.forName("com.mysql.jdbc.Driver");//.newInstance();
            String URL = "jdbc:mysql://localhost/quanlytaikhoan?user=root&password=";
            Connection con = DriverManager.getConnection(URL);
            return con;
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, ex.toString(), "Lỗi", JOptionPane.ERROR_MESSAGE);
            return null;
        }
    }
}

```

Annotations in the code:

- ten database** points to the database name in the URL.
- password** points to the password in the URL.
- địa chỉ server** points to the host in the URL.
- username** points to the username in the URL.

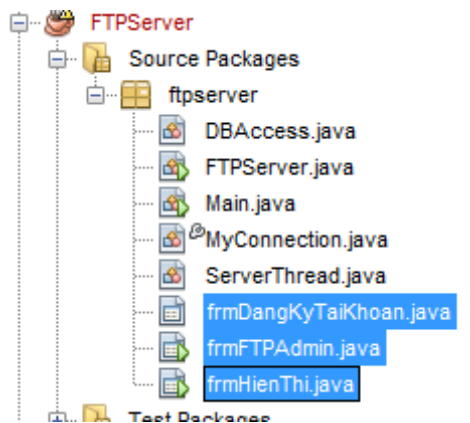
Tạo lớp DBAccess có nội dung như sau:



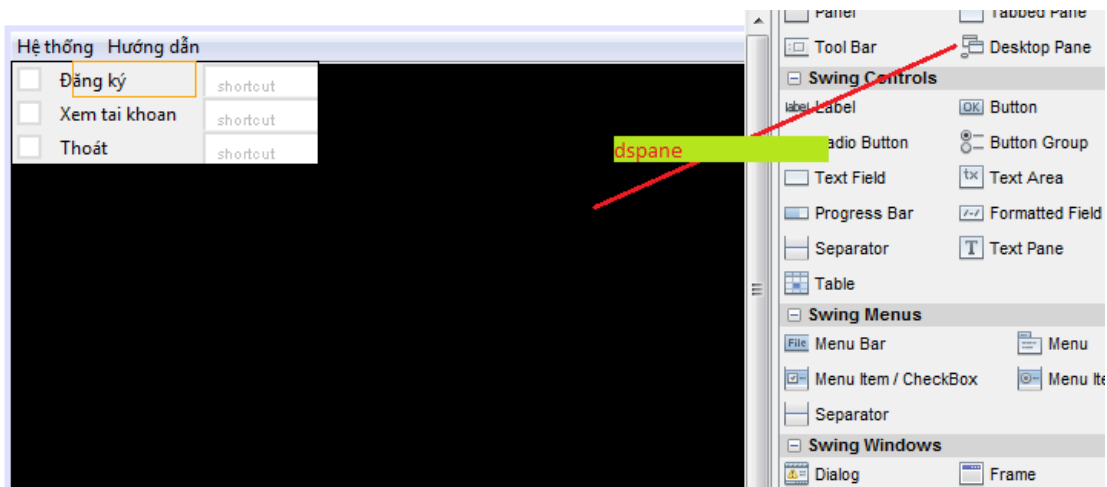

```
package ftpserver;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
public class DBAccess {
    private Connection con;
    private Statement stmt;
    public DBAccess() {
        try {
            MyConnection mycon=new MyConnection();
            con = mycon.getConnection();
            stmt=con.createStatement();
        } catch (Exception e) {
        }
    }
    public int Update(String str){
        try{
            int i=stmt.executeUpdate(str);
            return i;
        }catch(Exception e){
            return -1;
        }
    }
    public ResultSet Query(String srt){
        try {
            ResultSet rs=stmt.executeQuery(srt);
            return rs;
        } catch (Exception e) {
            return null;
        }
    }
}
```

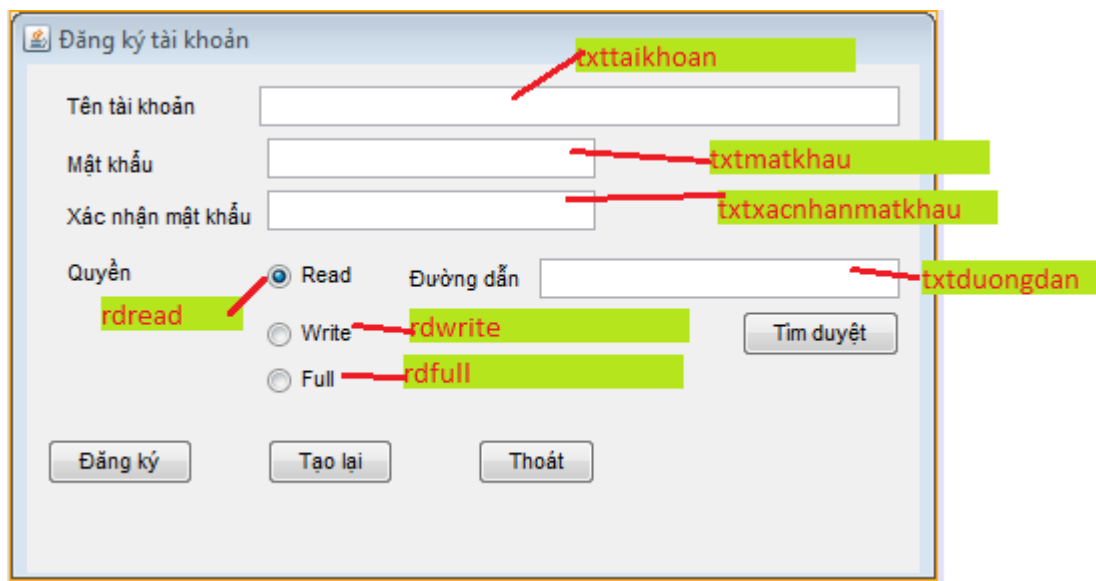
Bước 3.3: Tiếp đến, ta tạo các lớp đối tượng quản lý tài khoản bên phía server

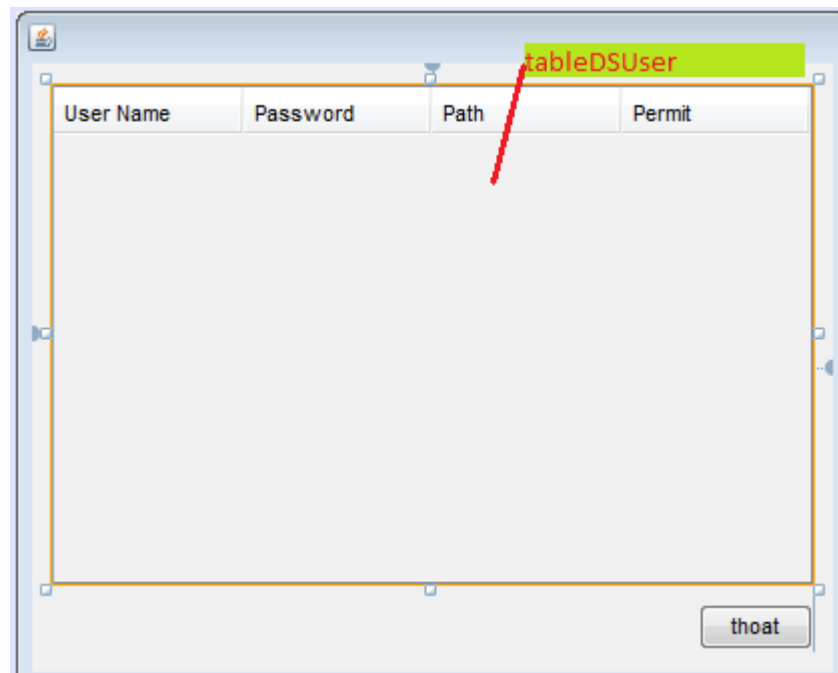


Bước 3.3.1: Tạo form frmFTPAdmin.java



Bước 3.3.2: Tạo frmDangKyTaiKhoan.java



Bước 3.3.3: Tạo frmHienThi.java**Bước 3.4: Bắt đầu viết sự kiện cho các form****Bước 3.4.1: frmFTPAdmin.java****Sự kiện click vào itemmenu Đăng ký**

```
private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {  
    frmDangKyTaiKhoan obj=new frmDangKyTaiKhoan();  
    obj.setVisible(true);  
    dspane.add(obj);  
}
```

Sự kiện click vào itemmenu Xem tài khoản

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    frmHienThi obj=new frmHienThi();  
    obj.setVisible(true);  
    dspane.add(obj);  
}
```

Sự kiện click vào itemmenu Thoát

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

Bước 3.4.2: frmDangKyTaiKhoan.java

Sự kiện click trên button Tìm duyệt

```
private void bttimduyetActionPerformed(java.awt.event.ActionEvent evt) {
    this.FileChooserServer.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    if(this.FileChooserServer.showOpenDialog(this)==JFileChooser.APPROVE_OPTION){
        try{
            //lay duong dang
            String path=this.FileChooserServer.getSelectedFile().getCanonicalPath();
            txtduongdan.setText(path);
            this.validate();
        }catch(Exception e)
        {
        }
    }
}
```

Xử lý sự kiện click trên button Thoát

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
```

Xử lý sự kiện click trên button tạo lại

```

    }
}

public void xoa() {
    txtmatkhau.setText("");
    txttaikhoan.setText("");
    txtxacnhanmatkhau.setText("");
    txttaikhoan.requestFocus();
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    xoa();
}
```

Xử lý sự kiện click trên button Đăng ký

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String username=txttaikhoan.getText();
    String password=new String(txtmatkhau.getPassword());
    String xnPassword=new String(txtxacnhanmatkhau.getPassword());
    int per=0;
    if(this.rdtype.isSelected())
        per=0;
    if(this.rdtype.isSelected())
        per=1;
    if(this.rdtype.isSelected())
        per=2;
    if(!password.equals(xnPassword)){
        JOptionPane.showMessageDialog(null, "loi mat khau ");
    }else{

```

```
try{
    //Lấy đường dẫn của project hiện hành
    File directory = new File ("");
    String path = txtduongdan.getText();
    if(path.equals("")){
        //Nếu không nhập đường dẫn thì sẽ tạo ra 1 thư mục
        // trùng tên với username và để cho user quản
        path=directory.getCanonicalPath ()+"\\ "+txttaikhoan.getText();
        (new File(path)).mkdir();
        path=path.replace("\\", "/");
    } else
        path=path.replace("\\", "/");
    System.out.print(path);
    DBAccess acc=new DBAccess();
    String query="insert into taikhoan (username,password,path,per) values('"+username+"','"+password+"','"+ path + "','"+
per + "')";
    System.out.println(query);
    if(acc.Update(query)>0){
        JOptionPane.showMessageDialog(null,"Đăng ký thành công!");
        xoa();
    } else{//Đăng nhập thất bại
        JOptionPane.showMessageDialog(null, "Đăng ký thất bại!");
    }
} catch(Exception e){
    JOptionPane.showMessageDialog(null, e.toString());
}
}
```

Bước 3.4.3 frmHienThi

Sự kiện click trên button Thoát

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.dispose();
}
```

Viết thêm code vào hàm dựng của lớp

```
public class frmHienThi extends javax.swing.JInternalFrame {  
  
    /** Creates new form frmHienThi */  
    public frmHienThi () {  
        initComponents();  
        //ket noi csdl hien thi ds tai
```

Thêm code vào phần này

```
public frmHienThi() {  
    initComponents();  
    //ket noi csdl hien thi ds tai khoan  
    try{  
        DBAccess acc=new DBAccess();  
        String query="select * from taikhoan";  
        ResultSet rs;  
        rs=acc.Query(query);  
        if(rs.isNull()){  
            System.out.println("rong");  
            return;  
        }  
        Vector row;  
        DefaultTableModel dm;  
        dm=(DefaultTableModel) this.tableDSUser.getModel();  
        while (rs.next()){  
            String user=rs.getString("username");  
            String pass=rs.getString("password");  
            String path=rs.getString("path");  
            int per=rs.getInt("per");  
            row=new Vector();  
            row.add(user);  
            row.add(pass);  
            row.add(path);  
            row.add(per);  
            dm.addRow(row);  
        }  
  
        this.validate();  
        this.setVisible(true);  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, e.toString(), "Lỗi", JOptionPane.ERROR_MESSAGE);  
    }  
}
```

Bước 3.5: Ta tiếp tục tạo các lớp xử lý dữ liệu từ client chuyển tới**Bước 3.5.1: FTPServer.java**

```
public class FTPServer {
    static final int PORT = 1234;
    private ServerSocket server=null;
    public FTPServer() {
        try{
            server=new ServerSocket(PORT);
        }catch(Exception e){e.printStackTrace();}
    }
    public void action(){
        Socket socket=null;
        int i=0;
        System.out.println("Serverlistening...");
        try{
            while ((socket=server.accept())!=null){
                new ServerThread(socket,"Client#"+i);
                System.out.printf("Thread for Client#%d generating...\n",i++);
            }
        }catch(Exception e){
            e.printStackTrace();
        }
    }
    public static void main(String[]args){
        new FTPServer().action();
    }
}
```

Bước 3.5.2: ServerThread.java

```
package ftpserver;

import java.io.*;
import java.util.*;
import java.net.*;
import java.sql.ResultSet;
//import java.sql.ResultSetMetaData;
import javax.swing.JOptionPane;

public class ServerThread implements Runnable{
    private Scanner in=null;
    private PrintWriter out=null;
    private Socket socket;
    private String name;

    public ServerThread(Socket socket,String name) throws IOException{
        this.socket=socket;
        this.name=name;
        this.in=new Scanner(this.socket.getInputStream());
    }
}
```

```
this.out=new PrintWriter(this.socket.getOutputStream(),true);

    new Thread(this).start();
}

//cac cau lenh giao tiep voi client
public static final int DANGNHAP=1;
public static final int DOWNLOAD=2;
public static final int NOCMD=-1;
public static final int UPLOAD=3;

public int lenh(String strcmd){
    if(strcmd.equals("dangnhap"))
        return DANGNHAP;//Client yeu cau dang nhap
    if(strcmd.equals("upload"))
        return UPLOAD;//client upload file len sever
    if(strcmd.equals("download"))
        return DOWNLOAD;//client yeu cau download file
    return NOCMD;//client khong co yeu cau gi
}

private void traThuMucClient(String path, PrintWriter out) {
    try{
        File dir=new File(path);
        File dsFile[];
        try{
            dsFile = dir.listFiles();
            if(dsFile==null)
                out.println(0);
            else{
                out.println(dsFile.length);
                for(int i=0;i<dsFile.length;i++){
                    String filename=dsFile[i].getName();
                    out.println(filename);
                }
            }
            out.flush();
        } catch(Exception e)
        {JOptionPane.showMessageDialog(null,e.toString());}

    } catch(Exception e){
        System.out.println(e.toString());
    }
}
```



```
}  
}  
  
    public void run(){  
        Scanner sc=null;  
        DBAccess acc=null;  
        String filename;  
        String path;  
        File f;  
        byte[] mybytearray;  
        try{  
            while(true){  
                String s=in.nextLine().trim();  
                sc=null;  
                String cmd="";  
                String data="";  
                try{  
                    sc=new Scanner(s);  
                    sc.useDelimiter("@");  
                    cmd=sc.next();  
                    data=sc.next();  
                } catch(Exception e){ }  
                //dieu phoi su kien tu client  
                switch(lenh(cmd))  
                {  
                    case DANGNHAP:  
                        acc=new DBAccess();  
                        ResultSet rs=acc.Query(data);  
                        if(rs.next()){  
                            out.println("OK");  
                            //out.println("\n");  
                            path=rs.getString("path");  
                            //mo thu muc ra va tra ve noi dung thu muc o phia server  
                            this.traThuMucClient(path,out);  
                        } else  
                            out.println("NOTOK");  
                        break;  
                    case UPLOAD:  
                        int bytesRead;  
                        int current=0;  
                        int doDaiFile=0;
```

```
rs=acc.Query(data);
filename=sc.next();
doDaiFile=Integer.parseInt(sc.next());
rs.next();
path=rs.getString("path");
f=new File(path+"/"+filename);
if(!f.exists())
    f.createNewFile();
mybytearray = new byte[doDaiFile];
InputStream is = socket.getInputStream();
FileOutputStream fos = new FileOutputStream(f);
BufferedOutputStream bos;
    bos = new BufferedOutputStream(fos);
bytesRead = is.read(mybytearray,0,mybytearray.length);
current = bytesRead;
while(current!=doDaiFile){
    bytesRead = is.read(mybytearray, current, mybytearray.length-current);
    if(bytesRead>=0) current+=bytesRead;
}
bos.write(mybytearray, 0, current);
bos.flush();
bos.close();
this.traThuMucClient(path,out);
break;
case DOWNLOAD:
    rs=acc.Query(data);
    filename=sc.next();
    rs.next();
    path=rs.getString("path");
    f=new File(path+"/"+filename);

    out.println(f.length()); //truyen do dai file

    mybytearray = new byte[(int) f.length()];
    BufferedInputStream bis = new BufferedInputStream(new FileInputStream(f));
    bis.read(mybytearray, 0, mybytearray.length);
    OutputStream os = socket.getOutputStream();
    os.write(mybytearray, 0, mybytearray.length);
    os.flush();
    bis.close();
}
```

```
    }  
    }catch(Exception e){  
        System.out.println(name+" has departed");  
    }finally{  
        try{socket.close();}catch(IOException e){}  
        if(sc!=null)sc.close();  
    }  
}  
}
```

Bước 3.6: Thêm các sự kiện cho các form bên phía client

Bước 3.6.1: frmFTPClient.java

Sự kiện cho itemmenu Đăng nhập

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    frmDangNhap obj=new frmDangNhap(this);  
    obj.setVisible(true);  
    dspane.add(obj);  
}
```

Sự kiện cho itemmenu Thoát

```
private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

Thêm các hằng số

```
package ftpclient;  
  
import java.net.Socket;  
  
public class frmFTPClient extends javax.swing.JFrame {  
    static final int PORT=1234;  
    static final String HOST="127.0.0.1";  
    /** Creates new form frmFTPClient */  
    public frmFTPClient() {  
        initComponents();  
        this.setExtendedState(javax.swing.JFrame.MAXIMIZED_BOTH);  
        this.validate();  
    }  
}
```

Viết thêm 1 hàm chuẩn bị cho việc tạo form Hiển thị

```
public void themFrmHienThi(Socket s,String username, String password){
    frmHienThi obj=new frmHienThi(s,username,password);
    obj.setVisible(true);
    dspane.add(obj);
    this.validate();
}
```

Bước 3.6.2: frmDangNhap.java

Khai báo thêm các thư viện và các biến

```
public class frmDangNhap extends javax.swing.JInternalFrame {
    private Socket socket;
    private PrintWriter out=null;
    private Scanner in=null;
    private frmFTPClient cha;
    /** Creates new form frmDangNhap */
    public frmDangNhap() {
        initComponents();
    }
    public frmDangNhap(frmFTPClient cha) {
        initComponents();
        this.cha=cha;
        txtusername.requestFocus();
    }
}
```

Khai báo sự kiện click cho button Thoát

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}
```

Khai báo sự kiện cho button Đăng nhập

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String username=txtusername.getText();
    String password=new String(txtpassword.getPassword());
    String str="select * from taikhoan where username='"+username+"' and password='"+password+"'";
    try{
        String data="";
        socket=new Socket(frmFTPClient.HOST,frmFTPClient.PORT);
        out=new PrintWriter(socket.getOutputStream(),true);
        in=new Scanner(socket.getInputStream());
        out.println("dangnhap@"+str);
        data=in.nextLine().trim();
        if(data.equals("OK")){// Đăng nhập thành công
            this.dispose();
            cha.themFrmHienThi(socket,username,password);
        }else{//Đăng nhập thất bại
            JOptionPane.showMessageDialog(null, "Đăng nhập thất bại! Hãy kiểm tra lại username và password.",
                "Lỗi",JOptionPane.ERROR_MESSAGE);
        }
        // socket.close();
    }catch(Exception e)
    {
        e.printStackTrace();
    }
}

```

Bước 3.6.3: frmHienThi.java

Viết thêm hàm loadlist

```

public void loadlist(){
    try{
        BufferedInputStream bi=new BufferedInputStream(socket.getInputStream());
        in=new Scanner(bi);
        DefaultListModel dm=new DefaultListModel();
        int n=in.nextInt();
        for(int i=0;i<=n;i++)
        {
            String filename=in.nextLine();
            dm.addElement(filename);
        }
        this.listUserFolder.setModel(dm);
        listUserFolder.setSelectedIndex(0);
        this.validate();

    }catch(Exception e){
        JOptionPane.showMessageDialog(null,e.toString(),"Lỗi",JOptionPane.ERROR_MESSAGE);
    }
}

```

Viết thêm constructor và các thuộc tính

```

public class frmHienThi extends javax.swing.JInternalFrame {
    private Socket socket=null;
    private PrintWriter out=null;
    private Scanner in=null;
    private String username="";
    private String password="";
    private String path="";
    /** Creates new form frmHienThi */
    public frmHienThi () {
        initComponents();
    }
    public frmHienThi(Socket s,String username,String password) {
        initComponents();
        this.socket=s;

        this.setTitle("Xin chào "+username);
        this.username=username;
        this.password=password;
        loadlist();
    }
}

```

Xử lý sự kiện click trên button Thoát

```

private void btnThoatActionPerformed(java.awt.event.ActionEvent evt) {
    this.dispose();
}

```

Xử lý sự kiện click trên button browser

```

private void btnBrowserActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    this.fileChooserClient.setSelectionMode(JFileChooser.DIRECTORIES_ONLY);
    if(this.fileChooserClient.showOpenDialog(this)==JFileChooser.APPROVE_OPTION){
        try{
            //lay duong dang
            path=this.fileChooserClient.getSelectedFile().getCanonicalPath();
            File dir=new File(path);
            File dsFile[]=dir.listFiles();
            if(dsFile==null){
                JOptionPane.showMessageDialog(null, "Đường dẫn không đúng hay không phải thư mục!");
            }else{
                DefaultListModel dm=new DefaultListModel();
                for(int i=0;i<dsFile.length;i++){
                    String filename=dsFile[i].getName();
                    dm.addElement(filename);
                }
                this.listClientFolder.setModel(dm);
                listClientFolder.setSelectedIndex(0);
            }
        }catch(Exception e)
        {
        }
    }
}

```

Xử lý sự kiện click trên button upload

```
private void btnUploadActionPerformed(java.awt.event.ActionEvent evt) {  
    String str="upload@"+"select * from taikhoan where username='"+username+"'@";  
    String filename;  
  
    if(!path.equals("")){  
        try{  
            filename=listClientFolder.getSelectedValue().toString();  
            File f=new File(path+"\\"+filename);  
            str=str+filename;  
            out=new PrintWriter(socket.getOutputStream(),true);  
            out.println(str+"@"+(int)f.length());  
  
            byte[] mybytearray = new byte[(int) f.length()];  
            BufferedInputStream bis = new BufferedInputStream(new FileInputStream(f));  
            bis.read(mybytearray, 0, mybytearray.length);  
            OutputStream os = socket.getOutputStream();  
            os.write(mybytearray, 0, mybytearray.length);  
            os.flush();  
            bis.close();  
            loadlist();  
        }catch(Exception e)  
        {  
            try{if(socket!=null)socket.close();}catch(Exception ex){e.printStackTrace();}  
            e.printStackTrace();  
        }  
    }else  
        JOptionPane.showMessageDialog(null, "Bạn chưa chọn file","Lỗi",JOptionPane.ERROR_MESSAGE);  
}
```

Xử lý sự kiện click trên button download

```
private void btnDownloadActionPerformed(java.awt.event.ActionEvent evt) {  
    String str="download@"+"select * from taikhoan where username='"+username+"'@";  
    String filename;  
  
    try{  
        //Truyền lệnh lên server yêu cầu file  
        filename=listUserFolder.getSelectedValue().toString();  
        str=str+filename;  
        out=new PrintWriter(socket.getOutputStream(),true);  
        out.println(str);  
        //Nhận file từ server về lưu trong path  
        BufferedInputStream bi=new BufferedInputStream(socket.getInputStream());  
        in=new Scanner(bi);  
        int doDaiFile=in.nextInt();//lay do dai file  
  
        if(path.equals("")){  
            this.fileChooserClient.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);  
            if(this.fileChooserClient.showOpenDialog(this)==JFileChooser.APPROVE_OPTION)  
                path=this.fileChooserClient.getSelectedFile().getCanonicalPath();  
        }  
    }
```

```
int bytesRead;
int current=0;
File f=new File(path+"\"+filename);
byte[] mybytearray = new byte[doDaiFile];
InputStream is = socket.getInputStream();
FileOutputStream fos = new FileOutputStream(f);
BufferedOutputStream bos = new BufferedOutputStream(fos);
bytesRead = is.read(mybytearray,0,mybytearray.length);
current = bytesRead;
while(current!=doDaiFile){
    bytesRead = is.read(mybytearray, current, mybytearray.length-current);
    if(bytesRead>=0) current+=bytesRead;
}
bos.write(mybytearray, 0, current);
bos.flush();
bos.close();
File dir=new File(path);
File dsFile[]=dir.listFiles();

DefaultListModel dm=new DefaultListModel();
for(int i=0;i<dsFile.length;i++){
    String name=dsFile[i].getName();
    dm.addElement(name);
}
this.listClientFolder.setModel(dm);
listClientFolder.setSelectedIndex(0);
} catch (Exception e)
{
    try{if(socket!=null)socket.close();} catch (Exception ex){e.printStackTrace();}
    e.printStackTrace();
}
}
```

Bước 3.7: Tiến hành chạy chương trình

- ✓ Start Wamp server
- ✓ Chạy frmFTPAdmin.java để tạo tài khoản mới.
- ✓ Chạy FTPServer.java
- ✓ Chạy file FTPClient.java để kiểm tra các chức năng download và upload .