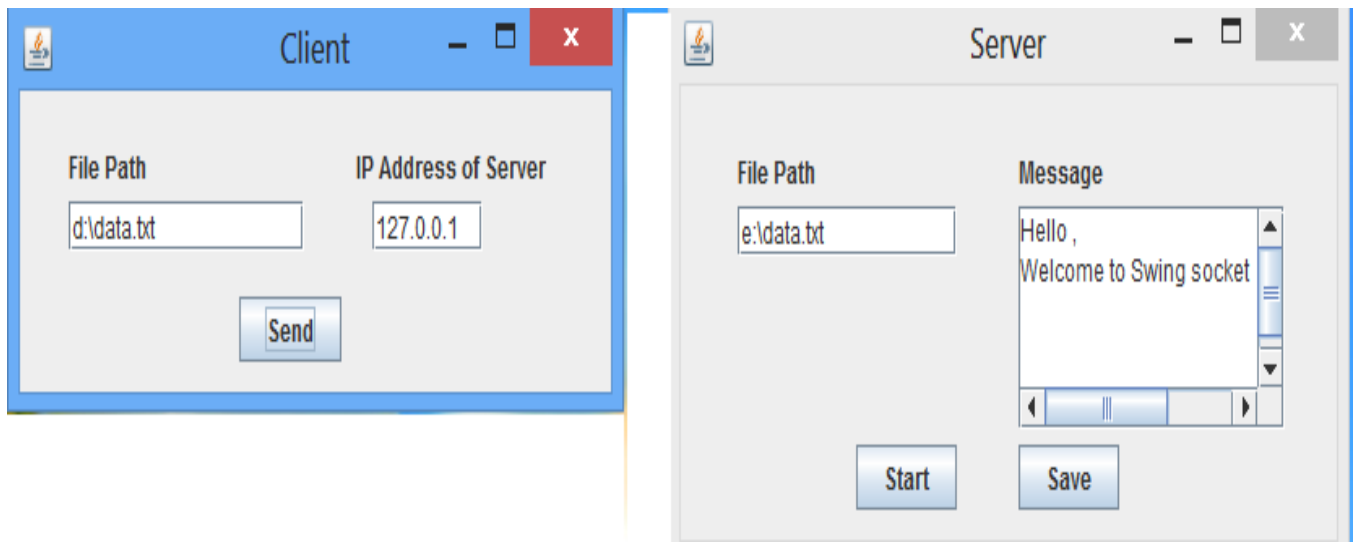


**Sau khi thực hành xong bài này, học viên có khả năng :**

- Xây dựng được ứng dụng Swing truyền dữ liệu Client – Server thông qua Socket
- Trình bày và sử dụng được các lớp : `ServerSocket`, `Socket`, `InputStreamReader`,...

Xây dựng hai ứng dụng JavaApplication sử dụng Swing có giao diện sau đây truyền dữ liệu từ tập tin .txt trên Client qua Server . Phía Server , user nhấn nút Save để lưu dữ liệu trong JTextArea vào trong tập tin .txt với đường dẫn nhập từ textfield .



❖ **Code cho phía Client**

```
public class Client extends javax.swing.JFrame {
    Socket sk;
    public Client() {
        initComponents();
    }
}
```

Generated Code

```
private void btnSendActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        //Khai bao socket ket noi den server
        sk = new Socket(txtServerIP.getText(),8888);
        //Tao doi tuong file de doc du lieu
        File f = new File(txtPathFile.getText());
        FileReader fr = new FileReader(f);
        //Doc du lieu tu file vao buffer
        BufferedReader br = new BufferedReader(fr);
        PrintStream ps = new PrintStream(sk.getOutputStream());
        String str;
        //gui du lieu qua Server
        while ((str = br.readLine())!=null) {
            ps.println(str);
        }
        ps.println("Finish");
        //nhan thong diep tra ve tu Server
        br = new BufferedReader(new InputStreamReader(sk.getInputStream()));
        if ((str = br.readLine()).equalsIgnoreCase("Send finish")) {
            JOptionPane.showMessageDialog(this,"Send file completed.");
        }
    }
}
```

```

        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                Client frm = new Client();
                frm.setLocation(300, 300);
                frm.setVisible(true);
            }
        });
    }
}

```

.....

}//end class

#### ❖ Code cho phía Server

```

public class Server extends javax.swing.JFrame {
    ServerSocket skServer;
    Socket sk;

    public Server() {
        initComponents();
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                Server frm = new Server();
                frm.setLocation(800, 300);
                frm.setVisible(true);
            }
        });
    }
}

```

```

private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    String data = txtMessage.getText();
    //Kiem tra ten tap tin khong bo trong
    if (txtPathFile.getText().isEmpty()){
        JOptionPane.showMessageDialog(this, "Please enter the file path");
        return;
    }
    try {
        //Ghi du lieu tu textbox ra file
        PrintWriter pw = new PrintWriter(txtPathFile.getText());
        pw.print(data);
        JOptionPane.showMessageDialog(this, "Save completed !");
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}

```

```

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

    try {
        skServer = new ServerSocket(8888);
        sk = skServer.accept();
        String data;
        BufferedReader br = new BufferedReader(new InputStreamReader(sk.getInputStream()));
        //nhan du lieu tu client cho den khi nhan duoc chuoi "Finish"
        while (!(data = br.readLine()).equalsIgnoreCase("Finish")){
            txtMessage.append(data);
            txtMessage.append("\n");
        }
        //Gui thong diep sau khi nhan cho Client
        PrintStream ps = new PrintStream(sk.getOutputStream());
        ps.println("Send finish");

    } catch (Exception e) {
        e.printStackTrace();
    }
}

```