Báo cáo thực hành 8 lập trình mạng

**Server.java**

Chương trình **Server** là một ứng dụng Java đơn giản thực hiện các chức năng cơ bản của một server. Dưới đây là mô tả các chức năng chính:

1. **Kết nối và Chờ Đợi:**
   * Server sử dụng **ServerSocket** để mở một cổng chờ kết nối từ client (ở đây là cổng 12345).
   * Sau khi mở cổng, server bắt đầu chờ các kết nối từ client bằng cách sử dụng **serverSocket.accept()**. Khi một client kết nối, server sẽ tạo một socket mới cho client và tiếp tục xử lý kết nối đó.
2. **Xử Lý Yêu Cầu từ Client:**
   * Server đọc yêu cầu từ client, chẳng hạn như tên file muốn download.
   * Nếu yêu cầu hợp lệ (**FILENAME**), server kiểm tra xem file tồn tại hay không và gửi thông tin về file (tên và kích thước) cho client.
3. **Truyền File cho Client:**
   * Nếu file tồn tại và client đã sẵn sàng (**READY**), server bắt đầu truyền nội dung của file cho client qua socket.
   * Server sử dụng **FileInputStream** để đọc từ file và **DataOutputStream** để gửi dữ liệu đến client.
4. **Đóng Kết Nối:**
   * Sau khi truyền xong, server đóng các luồng và kết nối socket.

**Client.java**

Chương trình **Client** là một ứng dụng Java đơn giản thực hiện các chức năng cơ bản của một client. Dưới đây là mô tả các chức năng chính:

1. **Nhập Địa Chỉ IP và Tên File:**
   * Client yêu cầu người dùng nhập địa chỉ IP của server và tên file mà client muốn download.
2. **Kết Nối đến Server:**
   * Client sử dụng **Socket** để kết nối đến server theo địa chỉ IP và cổng cụ thể.
3. **Gửi Yêu Cầu đến Server:**
   * Client gửi yêu cầu đến server, chẳng hạn như **FILENAME sample.txt**.
4. **Nhận Thông Tin về File từ Server:**
   * Client nhận thông tin về file từ server, chẳng hạn như tên file và kích thước file.
5. **Xác Nhận Sẵn Sàng và Nhận File:**
   * Client gửi xác nhận (**READY**) cho server nếu muốn tiếp tục nhận file.
   * Sau đó, client sử dụng **FileOutputStream** để lưu nội dung file được truyền từ server.
6. **Đóng Kết Nối:**
   * Sau khi nhận xong, client đóng kết nối socket.

