**Báo cáo thực hành 12**

Bài 1:

**Server (Server.java):**

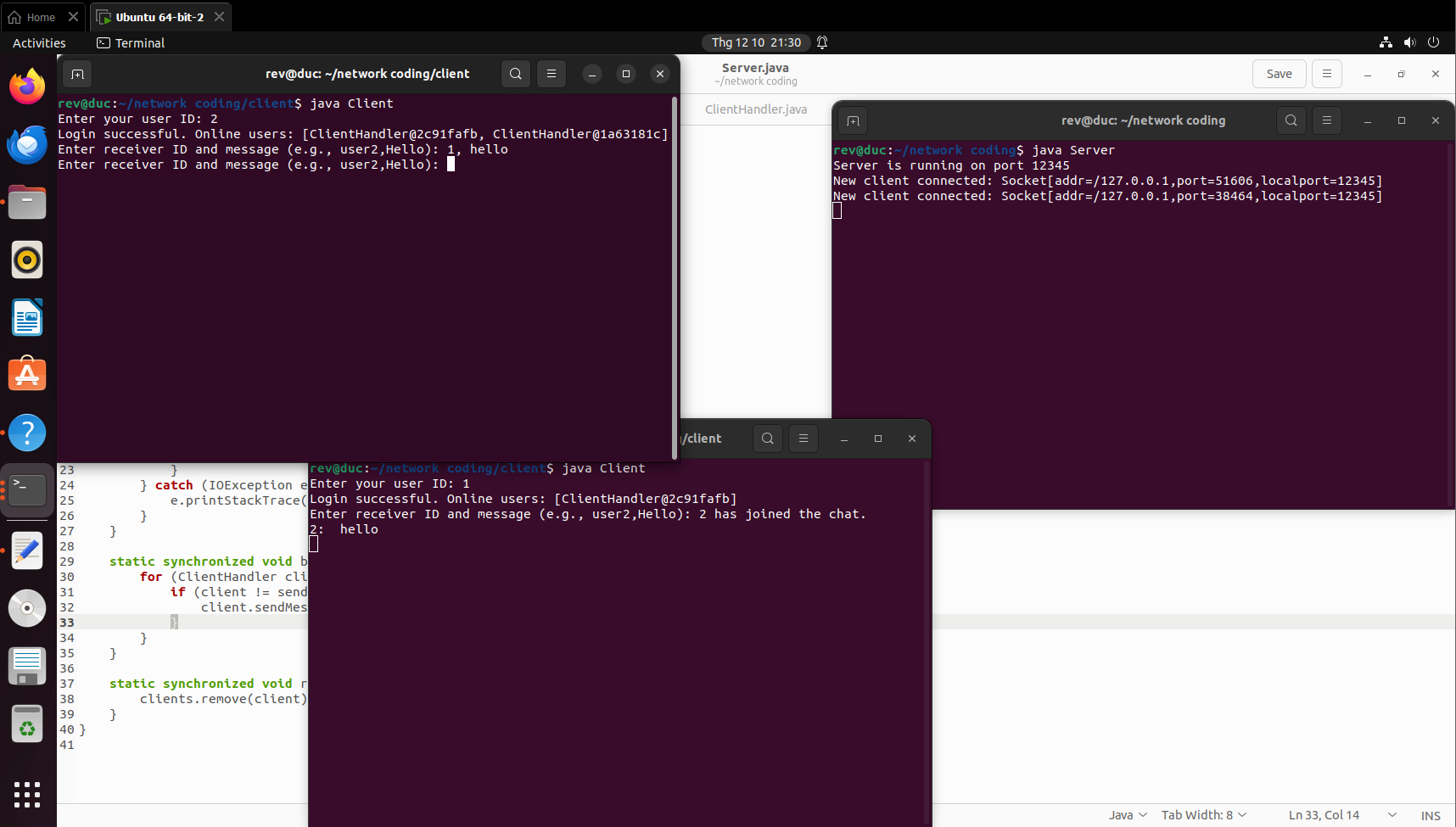
1. **Khởi tạo Server:**
   * Mở một **ServerSocket** trên một cổng đã chọn (ở đây là cổng 12345).
   * In ra console thông báo bắt đầu chạy server.
2. **Chấp nhận kết nối từ Clients:**
   * Trong vòng lặp vô hạn, chấp nhận kết nối từ clients bằng cách sử dụng **serverSocket.accept()**.
   * Mỗi khi có một client kết nối mới, tạo một **ClientHandler** mới và khởi động một thread mới để xử lý client đó.
3. **Broadcast Message:**
   * Cung cấp phương thức **broadcastMessage** để gửi một tin nhắn đến tất cả các clients, trừ client gửi tin nhắn (người gửi).
   * Đồng bộ hóa việc gửi tin nhắn để tránh xung đột.
4. **Remove Client:**
   * Cung cấp phương thức **removeClient** để xóa client khỏi danh sách khi client ngắt kết nối.

**Client (Client.java):**

1. **Khởi tạo Client:**
   * Mở một **Socket** để kết nối đến server với địa chỉ IP là "localhost" và cổng là 12345 (hoặc cổng mà server đang lắng nghe).
2. **Nhập User ID:**
   * Người dùng được yêu cầu nhập User ID.
3. **Gửi User ID đến Server:**
   * Gửi User ID đến server để đăng nhập.
4. **Gửi và Nhận Tin Nhắn:**
   * Người dùng có thể nhập tin nhắn và gửi đến server.
   * Client nhận tin nhắn từ server và hiển thị trên console.
5. **Kết thúc phiên làm việc:**
   * Khi người dùng nhập "exit", client gửi thông báo kết thúc phiên làm việc và đóng kết nối.

**ClientHandler (ClientHandler.java):**

1. **Xử lý Đăng nhập:**
   * Nhận User ID từ client và thông báo cho các client khác về sự xuất hiện của client mới.
2. **Xử lý Tin Nhắn:**
   * Trong vòng lặp vô hạn, nhận tin nhắn từ client và xử lý chúng.
   * Nếu tin nhắn là để chat (ví dụ: "user2,Hello user2!"), xác định người nhận và gửi tin nhắn cho người đó.
   * Nếu tin nhắn là để kết thúc phiên làm việc, thông báo cho các client khác về sự rời đi và đóng kết nối.
3. **Xử lý Đăng xuất:**
   * Thông báo cho các client khác về sự rời đi và đóng kết nối.
4. **Gửi Tin Nhắn:**
   * Cung cấp phương thức **sendMessage** để gửi tin nhắn đến client.
5. **Đồng bộ Hóa:**
   * Đồng bộ hóa việc gửi tin nhắn để tránh xung đột.



Bài 2:

**1. Server (Server.java):**

* **Server** là máy chủ chấp nhận kết nối từ nhiều clients và quản lý thông điệp giữa chúng.
* **Cổng kết nối:** Chương trình sử dụng cổng **12345** để lắng nghe kết nối.
* **ClientHandler:** Mỗi khi một client kết nối, một **ClientHandler** mới được tạo và quản lý trao đổi thông điệp với client đó.
* **Broadcast Messages:** Server có khả năng broadcast tin nhắn từ một client đến tất cả các client khác.

**2. Client (Client.java):**

* **Client** là phần mềm cho phép người dùng nhập địa chỉ IP của server, user ID của họ, và sau đó gửi và nhận thông điệp.
* **Gửi User ID**: User ID được gửi đến server khi client kết nối.
* **Broadcast Messages**: Client có thể gửi tin nhắn đến tất cả các client khác

**3. ClientHandler (ClientHandler.java):**

**ClientHandler** là một luồng riêng biệt được tạo ra cho mỗi client để xử lý truyền thông với server và các clients khác.

**Gửi và Nhận Tin Nhắn**: ClientHandler có thể gửi và nhận tin nhắn từ client và gửi chúng đến tất cả các clients khác.

**4. ServerListener (ServerListener.java):**

* **ServerListener** là một luồng riêng biệt chịu trách nhiệm đọc tin nhắn từ server và in chúng ra màn hình.

