CISC 3120

Exercise 5

Chatter

Overview

In this exercise:

- You will gain experience writing socket code.
- You will continue to gain experience using the Java API documentation to understand how to use these libraries.

Requirements

This is a pretty simple application, in concept. You're going to write another simple chat application. The first version of the chat room in the reading was "simple" in the sense that the client could send to the server, but it couldn't receive any messages sent by other clients. (Also… there isn't really a server running in that example!)

In this exercise, you'll write a simple chat client/server application that is "simple" in the sense that the two parties are *forced* to take turns with their chats. Any network communication requires a *protocol* that governs the behavior of the participants. Here's ours:

1. Once a connection is made, the server acts first, by asking the client what it wants.

2. The client responds.

3. The client and server alternate; they cannot send consecutive messages.

4. This repeats until one side terminates the connection by stopping their program.

Write two applications (called `ChatterClient` and `ChatterServer`) that implement this protocol. Do input with the keyboard/console (no GUI code). The client and server will have slightly different "setup" logic, but once the chat commences, their logic should be essentially identical.

Next Steps

If you get that working quickly/easily, then extend the protocol in a way that slightly relaxes the forced alternation. The new protocol has an extra rule: if the "current chatter" types a chat which ends in three periods ("…"), then instead of the other chatter getting their turn, the current chatter gets to type another chat. This can go on as long as the current chatter wants to keep talking (technically speaking, this allows the other chatter to *bogart the mic*). When the current chatter types something that doesn't end in three dots, then the other chatter gets their turn (or turns). Note that these three dots should have no spaces between or after them.