Issued to
Professor Solanki
CS 4390.003
Computer Networks
Spring 2023

Network Application Project Design Document

Due April 27, 2023

IMPLEMENTATION OF THE NETWORK APPLICATION

Server

The server is implemented with the TCPServer class and the ClientHandler class. The TCPServer class creates the server socket on the port number (given as a command line argument, our example output uses port 1234). It also sets the server socket up to listen for a client connection. Once the server socket hears a client, it is accepted and passed to the ClientHandler class. The thread created by the ClientHandler with the passed client can then be started to handle the rest of the interaction. As previously mentioned, the ClientHandler class extends Thread and takes in the clientSocket as a parameter. This class is responsible for setting up the input and output stream for the data to travel through. It then prompts the client for its name and logs the connection. Once the connection has been made and logged, it prompts the client to enter a math expression and listens until it reads in “e”, closes the input/output streams, disconnects from the client, and logs the interaction. If the ClientHandler thread receives a math expression, it sends it on to the Calculator class, prints the results, and keeps listening for the next expression.

Client

The client is implemented with the TCPClient class. This class creates the client socket that connects to the server on the given port (in this case, port 1234). Once the connection is made, the input/output streams are set up as well as a scanner so receive input from the user. The client reads in prompts from the server and sends back input from the user i.e. the client’s name, math expressions, and the exit character “e”. Once the client sends back the exit character, it closes the input/output streams and terminates the connection.

Calculator

The calculator is implemented with the Calculator class which has four methods:

```
public static double calculate(String expression);
public static boolean isOperator(char c);
public static int precedence(char c);
public static double performOperator(double num1, double num2, char op).
```

The calculate method takes the math expression string that the client passed to the server. It creates 2 stacks, one of numbers and one of operators. Every character is sorted into one of these stacks. If the character is a digit or is part of a digit, it is appended to a StringBuilder until the entire number is formed. The string created by the StringBuilder is then parsed into a double and pushed onto the number's stack. Alternatively, if the character is a parentheses or an operator (found by passing the character to the isOperator method), it is pushed onto the operators stack. To solve the equation, the performOperator method is called when an end parenthesis is read in, when the precedence of the top operator on the stack (found by passing operator.peek() to the precedence method) is greater than the precedence of the current character being examined, or while the operators stack isn't empty after all the characters are pushed onto one of the two stacks. (Note: when the performOperator is called due to a greater precedence of the top operator on the stack, the character being actively examined is not pushed onto the stack until the performOperator method has already been called). To get the parameters for the performOperator method, 2 numbers are popped from the numbers stack and 1 operator is popped from the operators stack. The performOperator method then compares the given operator with '+', '-', '*', and "/" until it finds a match and performs the appropriate operation. If a match the operator is not found, an exception is thrown. Assuming a match is found, the result of the operation is then pushed onto the number's stack. This process continues until there is only one number left on the stack which the calculate method returns to the server.