One way sever client communication

//serversite

import java.io.DataInputStream;

import java.io.IOException;

import java.net.\*;

public class ServerOneWay {

public static void main(String[] args) throws IOException {

ServerSocket ss = new ServerSocket(5306);

System.out.println("Server is connecting\n");

System.out.println("Waiting for client\n");

Socket s = ss.accept();

System.out.println("accepted the client request ");

DataInputStream input = new DataInputStream(s.getInputStream());

String str = "";

while (!str.equals("stop")) {

str = input.readUTF();

System.out.println("Client Says: " + str);

}

s.close();

input.close();

}

}

//clinet site

import java.io.BufferedReader;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.\*;

public class ClientOneWay {

public static void main(String[] args) throws IOException {

Socket s = new Socket("localhost", 5306);

System.out.println("Client is connecting" );

DataOutputStream output = new DataOutputStream(s.getOutputStream());

BufferedReader read = new BufferedReader(new InputStreamReader(System.in));

String str = "";

while (!str.equals("stop")) {

str = read.readLine();

output.writeUTF(str);

}

output.close();

read.close();

s.close();

}

}

//exam a jevabe asbe

Check the palindrome or lower to capital

\\two way

\server

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

public class twoWayServersite {

public static void main(String[] args) throws IOException {

ServerSocket ss = new ServerSocket(5306);

System.out.println("Server is connecting\n");

System.out.println("Waiting for client\n");

Socket s = ss.accept();

System.out.println("Accepted the client request");

DataInputStream input = new DataInputStream(s.getInputStream());

DataOutputStream output = new DataOutputStream(s.getOutputStream());

String str = "";

while (!str.equals("stop")) {

str = input.readUTF();

System.out.println("Client Says: " + str);

// Send a response back to the client

String response = "Server received: " + str;

output.writeUTF(response);

}

s.close();

input.close();

output.close();

}

}

\client

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class twoWayClinetside {

public static void main(String[] args) throws IOException {

Socket s = new Socket("localhost", 5306);

System.out.println("Client is connecting");

DataOutputStream output = new DataOutputStream(s.getOutputStream());

DataInputStream input = new DataInputStream(s.getInputStream());

BufferedReader read = new BufferedReader(new InputStreamReader(System.in));

String str = "";

while (!str.equals("stop")) {

// Read input from the console

str = read.readLine();

output.writeUTF(str);

// Read response from the server

String serverResponse = input.readUTF();

System.out.println(serverResponse);

}

output.close();

read.close();

input.close();

s.close();

}

}

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.ServerSocket;

import java.net.Socket;

public class ServerTwoWay {

public static void main(String[] args) throws IOException {

ServerSocket serverSocket = new ServerSocket(5306);

System.out.println("Server is connecting...\nWaiting for client...\n");

Socket socket = serverSocket.accept();

System.out.println("Accepted the client request");

DataInputStream input = new DataInputStream(socket.getInputStream());

DataOutputStream output = new DataOutputStream(socket.getOutputStream());

BufferedReader consoleInput = new BufferedReader(new InputStreamReader(System.in));

String clientMessage = "";

String serverMessage = "";

while (!clientMessage.equals("stop") && !serverMessage.equals("stop")) {

// Read message from the client

clientMessage = input.readUTF();

System.out.println("Client says: " + clientMessage);

// Check if the message is a palindrome

String palindromeResponse = isPalindrome(clientMessage) ? "is a palindrome." : "is not a palindrome.";

output.writeUTF("The message \"" + clientMessage + "\" " + palindromeResponse);

System.out.println("Server sent: " + "The message \"" + clientMessage + "\" " + palindromeResponse);

}

socket.close();

input.close();

output.close();

serverSocket.close();

}

// Function to check if a string is a palindrome

private static boolean isPalindrome(String str) {

String cleaned = str.replaceAll("[^a-zA-Z0-9]", "").toLowerCase();

return cleaned.equals(new StringBuilder(cleaned).reverse().toString());

}

}

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class ClientTwoWay {

public static void main(String[] args) throws IOException {

Socket socket = new Socket("localhost", 5306);

System.out.println("Client is connecting...");

DataOutputStream output = new DataOutputStream(socket.getOutputStream());

DataInputStream input = new DataInputStream(socket.getInputStream());

BufferedReader consoleInput = new BufferedReader(new InputStreamReader(System.in));

String clientMessage = "";

String serverMessage = "";

while (!clientMessage.equals("stop") && !serverMessage.equals("stop")) {

// Get input from the client console

System.out.print("Client: ");

clientMessage = consoleInput.readLine();

output.writeUTF(clientMessage);

System.out.println("Client sent: " + clientMessage);

// Read message from the server

serverMessage = input.readUTF();

System.out.println("Server says: " + serverMessage);

}

output.close();

input.close();

socket.close();

}

}

private static boolean isPrime(String str) {

int number;

try {

number = Integer.parseInt(checked);

} catch (NumberFormatException e) {

return false;

}

if (number <= 1) {

return false; // 0 and 1 are not prime

}

for (int i = 2; i <= Math.sqrt(number); i++) {

if (number % i == 0) {

return false;

}

}

return true;

}

\\prime chaeck

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.ServerSocket;

import java.net.Socket;

public class ServerTwoWay {

public static void main(String[] args) throws IOException {

ServerSocket serverSocket = new ServerSocket(5306);

System.out.println("Server is connecting...\n");

System.out.println("The connection is builted successfully ");

Socket socket = serverSocket.accept();

System.out.println("Accepted request from client request");

DataInputStream input = new DataInputStream(socket.getInputStream());

DataOutputStream output = new DataOutputStream(socket.getOutputStream());

BufferedReader consoleInput = new BufferedReader(new InputStreamReader(System.in));

String clientMessage = "";

String serverMessage = "";

while (!clientMessage.equals("stop") && !serverMessage.equals("stop")) {

clientMessage = input.readUTF();

System.out.println("Client says: " + clientMessage);

// check the palindrome

String primeCheck = isPrime(clientMessage) ? "is a prime Number." : "NO! not a prime number";

output.writeUTF("The message \"" + clientMessage + "\" " + primeCheck);

System.out.println("Server sent: " + "The message \"" + clientMessage + "\" " + primeCheck);

}

socket.close();

input.close();

output.close();

serverSocket.close();

}

private static boolean isPrime(String checked) {

int number;

try {

number = Integer.parseInt(checked);

} catch (NumberFormatException e) {

return false;

}

if (number <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(number); i++) {

if (number % i == 0) {

return false;

}

}

return true;

}

}

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.Socket;

public class ClientTwoWay {

public static void main(String[] args) throws IOException {

Socket socket = new Socket("localhost", 5306);

System.out.println("Client is connecting...");

DataOutputStream output = new DataOutputStream(socket.getOutputStream());

DataInputStream input = new DataInputStream(socket.getInputStream());

BufferedReader consoleInput = new BufferedReader(new InputStreamReader(System.in));

String clientMessage = "";

String serverMessage = "";

while (!clientMessage.equals("stop") && !serverMessage.equals("stop")) {

// Get input from the client console

System.out.print("Client: ");

clientMessage = consoleInput.readLine();

output.writeUTF(clientMessage);

System.out.println("Client sent: " + clientMessage);

// Read message from the server

serverMessage = input.readUTF();

System.out.println("Server says: " + serverMessage);

}

output.close();

input.close();

socket.close();

}

}