import java.io.\*;

import java.net.\*;

public class PalindromeServer {

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

ServerSocket serverSocket = new ServerSocket(5000);

System.out.println("Server is running on port 5000...");

while (true) {

Socket clientSocket = serverSocket.accept();

System.out.println("Client connected: " + clientSocket);

DataInputStream dis = new DataInputStream(clientSocket.getInputStream());

DataOutputStream dos = new DataOutputStream(clientSocket.getOutputStream());

String input;

while (true) {

dos.writeUTF("Enter a word or number to check (type 'Exit' to quit): ");

input = dis.readUTF();

if (input.equalsIgnoreCase("Exit")) {

dos.writeUTF("Connection closed.");

break;

}

boolean result = isPalindrome(input);

if (result) {

dos.writeUTF("The input \"" + input + "\" is a Palindrome.");

} else {

dos.writeUTF("The input \"" + input + "\" is Not a Palindrome.");

}

}

clientSocket.close();

dis.close();

dos.close();

}

}

public static boolean isPalindrome(String input) {

String cleanInput = input.replaceAll("\\s+", "").toLowerCase();

int left = 0, right = cleanInput.length() - 1;

while (left < right) {

if (cleanInput.charAt(left) != cleanInput.charAt(right)) {

return false;

}

left++;

right--;

}

return true;

}

}

…………………………..

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class PalindromeClient {

public static void main(String[] args) {

try {

Socket clientSocket = new Socket("localhost", 5000);

DataOutputStream dos = new DataOutputStream(clientSocket.getOutputStream());

DataInputStream dis = new DataInputStream(clientSocket.getInputStream());

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println(dis.readUTF());

String input = scanner.nextLine();

dos.writeUTF(input);

if (input.equalsIgnoreCase("Exit")) {

System.out.println("Connection closed.");

break;

}

String response = dis.readUTF();

System.out.println(response);

}

clientSocket.close();

dis.close();

dos.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

import java.io.\*;

import java.net.\*;

public class ServerThread {

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

ServerSocket serverSocket = new ServerSocket(5000);

System.out.println("Server is running on port 5000...");

int client = 0;

while (client < 4) {

Socket clientSocket = serverSocket.accept();

client++;

System.out.println("Client " + client + " connected: " + clientSocket);

DataInputStream dis = new DataInputStream(clientSocket.getInputStream());

DataOutputStream dos = new DataOutputStream(clientSocket.getOutputStream());

Thread clientThread = new ClientHandler(clientSocket, dis, dos, client);

clientThread.start();

}

System.out.println("Server lifetime ended. No more clients accepted.");

serverSocket.close();

}

}

class ClientHandler extends Thread {

private final Socket socket;

private final DataInputStream dis;

private final DataOutputStream dos;

private final int client;

public ClientHandler(Socket socket, DataInputStream dis, DataOutputStream dos, int client) {

this.socket = socket;

this.dis = dis;

this.dos = dos;

this.client = client;

}

@Override

public void run() {

try {

int sentenceCount = 0;

while (sentenceCount < 3) {

dos.writeUTF("Client " + client + ": Send a sentence (or type 'Exit' to quit): ");

String received = dis.readUTF();

if (received.equalsIgnoreCase("Exit")) {

System.out.println("Client " + client + " disconnected.");

break;

}

String convertedSen = capitalizeMiddleLetters(received);

dos.writeUTF("Processed: " + convertedSen);

System.out.println("Processed for Client " + client + ": " + convertedSen);

sentenceCount++;

}

System.out.println("Client " + client + " session ended.");

socket.close();

dis.close();

dos.close();

} catch (IOException e) {

System.err.println("Error handling client " + client + ": " + e.getMessage());

}

}

private String capitalizeMiddleLetters(String sentence) {

String[] words = sentence.split(" ");

StringBuilder result = new StringBuilder();

for (String word : words) {

if (word.length() == 0) continue;

int mid = word.length() / 2;

char middleChar = Character.toUpperCase(word.charAt(mid));

String transformedWord = word.substring(0, mid) + middleChar + word.substring(mid + 1);

result.append(transformedWord).append(" ");

}

return result.toString().trim();

}

}

………………….

import java.io.\*;

import java.net.\*;

import java.util.Scanner;

public class ClientThread {

public static void main(String[] args) {

try {

Socket clientSocket = new Socket("localhost", 5000);

System.out.println("Connected to server.");

DataOutputStream dos = new DataOutputStream(clientSocket.getOutputStream());

DataInputStream dis = new DataInputStream(clientSocket.getInputStream());

Scanner scanner = new Scanner(System.in);

while (true) {

String serverMessage = dis.readUTF();

System.out.println(serverMessage);

System.out.print("Please Enter any sentences: ");

String input = scanner.nextLine();

dos.writeUTF(input);

if (input.equalsIgnoreCase("Exit")) {

System.out.println("Exiting session.");

break;

}

String response = dis.readUTF();

System.out.println("Server Response: " + response);

}

clientSocket.close();

dis.close();

dos.close();

} catch (IOException e) {

System.err.println("Error: " + e.getMessage());

}

}

}