For: Melissa Bakke

Assignment: Exercise 31-9 Chat

|  |
| --- |
| **Screenshot(s)** |
|  |
|  |

|  |
| --- |
| **Code** |
| *import java.io.DataInputStream;*  *import java.io.DataOutputStream;*  *import java.io.IOException;*  *import java.net.ServerSocket;*  *import java.net.Socket;*  *import javafx.application.Application;*  *import javafx.geometry.Insets;*  *import javafx.scene.Scene;*  *import javafx.scene.control.Label;*  *import javafx.scene.control.TextArea;*  *import javafx.scene.input.KeyCode;*  *import javafx.scene.layout.VBox;*  *import javafx.stage.Stage;*  */\*\**  *\* Class: Chat Server*  *\* Developer: Melissa Bakke*  *\* Date: 04/18/2017*  *\* Purpose: Program that enables two users to chat.*  *\*/*  *public class Server extends Application {*  *// IO streams*  *DataOutputStream toClient = null;*  *DataInputStream fromClient = null;*  *@Override*  *public void start(Stage primaryStage) {*  *//Text area for displaying contents*  *TextArea clientTA = new TextArea();*  *TextArea serverTA = new TextArea();*  *clientTA.setEditable(false);*    *VBox vBox = new VBox(10);*  *Label lblClient = new Label("Client");*  *Label lblServer = new Label("Server");*  *vBox.getChildren().addAll(lblClient, clientTA, lblServer, serverTA);*  *vBox.setPadding(new Insets(10));*    *// Create a scene and place it in the stage*  *Scene scene = new Scene(vBox, 450, 450);*  *primaryStage.setTitle("Server");*  *primaryStage.setScene(scene);*  *primaryStage.show();*    *// Gives the text area focus when the application starts*  *serverTA.requestFocus();*    *// Send text back to the client*  *serverTA.setOnKeyPressed(event -> {*  *if(event.getCode() == KeyCode.ENTER){*  *try {*  *String[] sa = serverTA.getText().split("\n");*  *toClient.writeUTF(sa[sa.length - 1]);*  *toClient.flush();*  *} catch (IOException ex) {*  *ex.printStackTrace();*  *}*  *}*  *});*    *new Thread(() -> {*  *try {*  *// Create a server socket*  *ServerSocket serverSocket = new ServerSocket(8000);*    *// Listen for a connection request*  *Socket socket = serverSocket.accept();*    *// Create data input and output streams*  *fromClient = new DataInputStream(socket.getInputStream());*  *toClient = new DataOutputStream(socket.getOutputStream());*    *while (true) {*  *// Receive text from the client*  *String message = fromClient.readUTF();*    *// Display message in text area*  *clientTA.appendText(message + "\n");*  *toClient.flush();*  *}*  *}*  *catch (IOException ex) {*  *ex.printStackTrace();*  *}*  *}).start();*  *}*  */\*\**  *\* @param args the command line arguments*  *\*/*  *public static void main(String[] args) {*  *Application.launch(args);*  *}*    *}*  *import java.io.DataInputStream;*  *import java.io.DataOutputStream;*  *import java.io.IOException;*  *import java.net.Socket;*  *import javafx.application.Application;*  *import javafx.geometry.Insets;*  *import javafx.scene.Scene;*  *import javafx.scene.control.Label;*  *import javafx.scene.control.TextArea;*  *import javafx.scene.input.KeyCode;*  *import javafx.scene.layout.VBox;*  *import javafx.stage.Stage;*  */\*\**  *\* Class: Chat Client*  *\* Developer: Melissa Bakke*  *\* Date: 04/18/2017*  *\* Purpose: Program that enables two users to chat.*  *\*/*  *public class Client extends Application {*  *// IO streams*  *DataOutputStream toServer = null;*  *DataInputStream fromServer = null;*    *@Override*  *public void start(Stage primaryStage) {*  *//Text area for displaying contents*  *TextArea clientTA = new TextArea();*  *TextArea serverTA = new TextArea();*  *serverTA.setEditable(false);*    *VBox vBox = new VBox(10);*  *Label lblClient = new Label("Client");*  *Label lblServer = new Label("Server");*  *vBox.getChildren().addAll(lblServer, serverTA, lblClient, clientTA);*  *vBox.setPadding(new Insets(10));*    *// Create a scene and place it in the stage*  *Scene scene = new Scene(vBox, 450, 450);*  *primaryStage.setTitle("Client");*  *primaryStage.setScene(scene);*  *primaryStage.show();*    *// Gives the text area focus when the application starts*  *clientTA.requestFocus();*    *// Send text back to the server*  *clientTA.setOnKeyPressed(event -> {*  *if(event.getCode() == KeyCode.ENTER){*  *try {*  *String[] sa = clientTA.getText().split("\n");*  *toServer.writeUTF(sa[sa.length - 1]);*  *toServer.flush();*  *} catch (IOException ex) {*  *ex.printStackTrace();*  *}*  *}*  *});*    *try {*  *// Create a socket to connect to the server*  *Socket socket = new Socket("localhost", 8000);*    *// Create an input stream to receive data from the server*  *fromServer = new DataInputStream(socket.getInputStream());*    *// Create an output stream to send data to the server*  *toServer = new DataOutputStream(socket.getOutputStream());*  *} catch (IOException ex) {*  *clientTA.appendText(ex.toString() + "\n");*  *}*    *new Thread(() -> {*  *try {*    *while (true) {*  *// Receive text from the server*  *String message = fromServer.readUTF();*    *// Display message in text area*  *serverTA.appendText(message + "\n");*  *toServer.flush();*  *}*  *}*  *catch (IOException ex) {*  *ex.printStackTrace();*  *}*  *}).start();*  *}*  */\*\**  *\* @param args the command line arguments*  *\*/*  *public static void main(String[] args) {*  *Application.launch(args);*  *}*  *}* |
|  |