import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

import java.util.UUID;

/\*--------------------------------------------------------

1. Name: Athina Verroiopoulou / Date: 1/22/2017

2. Java version used: 1.8

3. Precise command-line compilation examples / instructions: e.g.:

> javac JokeClient.java or javac \*.java

4. Precise examples / instructions to run this program: e.g.:

In separate shell windows:

> java JokeServer

> java JokeClient

> java JokeClientAdmin

All acceptable commands are displayed on the various consoles.

This runs across machines, in which case you have to pass the IP address of

the server to the clients. For example, if the server is running at 140.192.1.22 then you would type:

> java JokeClient 192.168.0.34 (my IP)

> java JokeClientAdmin 192.168.0.34 (my IP)

5. List of files needed for running the program. e.g.:

a. checklist.html

b. JokeServer.java

c. JokeClient.java

d. JokeClientAdmin.java

5. Notes: e.g.: works as it is required

----------------------------------------------------------\*/

//JokeClient ask for a username and then a user have to press enter in order to request for a joke/proverb

public class JokeClient {

static final String cookie = UUID.randomUUID().toString(); //generate unique sequences of bytes , want to create a cookie and pass it to server

public static void main(String args[]) {

String serverName;

if (args.length < 1) //in case we don't give another IP,input

serverName = "localhost"; //the default choice is the local host

else

serverName = args[0]; //if we give an ip it connects to that one, for my machine the IP is : "192.168.0.34"

System.out.println("!!! Athina's JokeClient !!!\n"); //print the message

System.out.println("Server that we try to connect is : " + " " +serverName + ", listening to Port: 4545\n");//print server name

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

try {

String idName;//a string that is used to save the user name provided by the user

String connect=null; //depending the command will connect to the server, have to be enter

System.out.println("Please enter your username for identification");

idName = in.readLine(); //read the user name

do { //run until quit message or a word with "quit " inside

System.out.print("The user : " + " " + idName +" asks for a joke or proverb ," + "Please press enter to confirm or quit to exit: ");

System.out.flush();//is used to empty the memory from every byte that is written in the buffer

connect = in.readLine();//reads enter or quit

if (connect.isEmpty()&& (connect.indexOf("quit") < 0)){ //no quit and enter from user

getConnection(idName,serverName); //user name and server name as parameter

}

else if (connect.equals("quit")){System.out.println("user wants to exit \n");}

else {System.out.println("Wrong input,Press enter or w");} //in case of invalid output, not enter

} while (connect.indexOf("quit") < 0); // this will run until the user give an exit message , a word with "quit " inside

System.out.println("Exit from JokeClient \n"); //print exit message

} catch (IOException x) {

x.printStackTrace();

}

}

//this method is called from the main to connect with the server

static void getConnection(String idName,String serverName) {

String serverResponce; //a string to save server answer

try {

Socket sock = new Socket(serverName, 4545); //connection with server and port 4545

//The BufferedReader and PrintStream are used to send and receive messages from sockets

BufferedReader receiveFromServer = new BufferedReader(new InputStreamReader(sock.getInputStream())); //read

PrintStream sendToServer = new PrintStream(sock.getOutputStream()); //write

//The server is waiting for client to ask for a joke or proverb

sendToServer.println(idName); //send a request to the server with the user name

sendToServer.flush(); //clean the memory

sendToServer.println(cookie); //send cookie

sendToServer.flush(); //clean the memory

//server response should be a joke or proverb

serverResponce = receiveFromServer.readLine(); //the client reads the server message

if (serverResponce != null)//if the server answer is not empty

System.out.println(serverResponce);//print server message

sock.close(); //we call the close method of Java.net.Socket class to terminate the connection

} catch (IOException x) {

System.out.println("Socket error."); //the try catch is used to identify connection errors

x.printStackTrace();

}

}

}

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

/\*--------------------------------------------------------

1. Name: Athina Verroiopoulou / Date: 1/22/2017

2. Java version used: 1.8

3. Precise command-line compilation examples / instructions: e.g.:

> javac JokeClientAdmin.java or javac \*.java

4. Precise examples / instructions to run this program: e.g.:

In separate shell windows:

> java JokeServer

> java JokeClient

> java JokeClientAdmin

All acceptable commands are displayed on the various consoles.

This runs across machines, in which case you have to pass the IP address of

the server to the clients. For example, if the server is running at 140.192.1.22 then you would type:

> java JokeClient 192.168.0.34 (my IP)

> java JokeClientAdmin 192.168.0.34 (my IP)

5. List of files needed for running the program. e.g.:

a. checklist.html

b. JokeServer.java

c. JokeClient.java

d. JokeClientAdmin.java

5. Notes: e.g.:

Runs according to the requirements. Have to press pro for proverb or j for joke or word with "quit " inside to exit

anything else is consider invalid input.

Consider to send one more selection for exit to terminate the server,

so the server could have one more switch case choice, but did not work out as expected

----------------------------------------------------------\*/

// JokeClientAdmin sets the server in different modes (joke/proverb)

public class JokeClientAdmin {

public static void main(String args[]) {

String serverName;

if (args.length < 1) //in case we don't give another IP,input

serverName = "localhost"; //the default choice is the local host

else

serverName = args[0];//if we give an ip it connects to that one

System.out.println("!!! Athina's JokeClientAdmin !!!\n");

System.out.println("Server that we try to connect is : " + " " +serverName + ", listening to Port: 5050 \n");//print the given server name we try to connect

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));//read from Socket

try {

String mode;//a string that is used to save the user command mode

do {//run until quit message or a word with "quit " inside

//give the choices for different mode,the user have to press "pro" or "j" or "quit"

System.out.println("Please enter the mode u want: "+

"Select pro: for proverb mode or j: for joke mode or quit to exit the adminClient \n");

System.out.flush();//is used to empty the memory from every byte that is written in the buffer

mode = in.readLine(); //read the mode choice from the buffer

if ((mode.indexOf("quit") < 0)&&(((mode.equals("pro"))||(mode.equals("j"))))) {

//if the mode is proverb or joke we call the method

getModeToServer(mode,serverName); //take mode and server name as parameter

}

else if (mode.equals("quit")){System.out.println("admin wants to exit \n");}

else {System.out.println("Wrong input,Press enter or w");} //check for invalid selection

} while (mode.indexOf("quit") < 0); // this will run until the user give an exit message or a word with "quit " inside

System.out.println("Exit From JokeClientAdmin \n");//print exit message

} catch (IOException x) {

x.printStackTrace();

}

}

// this method is called from the main to connect with the server and pass the selected mode

static void getModeToServer(String mode, String serverName) {

String serverResponce; // a string to save server answer

try {

Socket sock = new Socket(serverName, 5050); // connection with server in different port than JokeClient

// The BufferedReader and PrintStream are used to send and receive messages from sockets

BufferedReader receiveFromServer = new BufferedReader(new InputStreamReader(sock.getInputStream())); //read

PrintStream sendToServer = new PrintStream(sock.getOutputStream()); //write

// The server is waiting for admin client to set him in a mode

sendToServer.println(mode); // send the selected mode to the server

sendToServer.flush();// clean the memory

serverResponce = receiveFromServer.readLine(); // read server response

if (serverResponce != null) // if the server answer is not empty

//the message should be: The selected mode is : proverb/joke

//as default is the joke mode

System.out.println(serverResponce);// print server message

sock.close(); // we call the close method of Java.net.Socket class to terminate the connection

} catch (IOException x) {

System.out.println("Socket error."); //the try catch is used to identify connection errors

x.printStackTrace();

}

}

}

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Hashtable;

import java.util.Iterator;

/\*--------------------------------------------------------

1. Name: Athina Verroiopoulou / Date: 1/22/2017

2. Java version used: 1.8

3. Precise command-line compilation examples / instructions: e.g.:

> javac JokeServer.java or javac \*.java

4. Precise examples / instructions to run this program: e.g.:

In separate shell windows:

> java JokeServer

> java JokeClient

> java JokeClientAdmin

All acceptable commands are displayed on the various consoles.

This runs across machines, in which case you have to pass the IP address of

the server to the clients. For example, if the server is running at 140.192.1.22 then you would type:

> java JokeClient 192.168.0.34 (my IP)

> java JokeClientAdmin 192.168.0.34 (my IP)

5. List of files needed for running the program. e.g.:

a. checklist.html

b. JokeServer.java

c. JokeClient.java

d. JokeClientAdmin.java

5. Notes: e.g.:

Not implement the second server

----------------------------------------------------------\*/

class Cookie { //helper class that will allow the connection of the joke and proverb arraylists with the client hashtable

//volatile is a keyword that is used in multithreading, allow null values, when we access volatile synchronization is happening

private volatile ArrayList <String> j;

private volatile ArrayList <String> p;

public Cookie() {

j = new ArrayList<String>();

p = new ArrayList<String>();

}

public ArrayList <String> getJoke(){

return j;

}

public ArrayList <String> getProverb(){

return p;

}

}

class Worker extends Thread {

private static ArrayList <String> proverb = new ArrayList <String> (); //a list for the proverb

private static ArrayList <String> jokes = new ArrayList <String> (); //a list for the jokes

//we select hashtable and not hashmap as hashtable is synchronized

static volatile Hashtable<String,Cookie> clients = new Hashtable<String,Cookie>(); //store client cookies into the hashtable with type of String and Cookie class

public static int mode=0; // the default mode from client admin is for jokes (0 mode)

Cookie newCookie; //class cookie

protected Socket sock; //Create a local Socket

//create the socket for communication, act as a constructor

Worker(Socket s) {

sock = s; //pass s as a socket parameter and assign it to sock

}

public void run() {

PrintStream sendClientResponse = null; //initialize the value

BufferedReader readClientMessage = null; //initialize the value

try {

//connect with a socket and read and send messages to a client

readClientMessage = new BufferedReader(new InputStreamReader(sock.getInputStream())); //read

sendClientResponse = new PrintStream(sock.getOutputStream()); //write

try {

String userName;//save the client response - the given user name

userName = readClientMessage.readLine(); //read what the client message, the user name

System.out.println("connection to the client with Username" +" " + userName); //message with the user name

String cookie = readClientMessage.readLine(); //read from JokeClient the cookie

//If this is a new client, add it to the client hashtable

if (!clients.containsKey(cookie)){ //if the given cookie not exist in the hashtable

newCookie = new Cookie(); //call the helper class

//in the hashtable we store the cookie and that for that client the arraylists should be unique

clients.put(cookie, newCookie); //map a key (cookie) to a value

}

jokes=clients.get(cookie).getJoke(); //retrieve the cookie from the hashtable and the joke for that client

proverb=clients.get(cookie).getProverb();

System.out.println("My cookie is" +" " + cookie); //print the cookie , for helping reasons - no need

getJokesProverb(userName,sendClientResponse); //call the method, username and output is parameters

} catch (IOException x) {

System.out.println("Read error"); //in case of error in reading

x.printStackTrace();

}

sock.close(); //we call the close method of Java.net.Socket class to terminate the connection with the specific client, not the server

System.out.println("Worker thread closed");

} catch (IOException ioe) {

System.out.println(ioe);

}

}

//inside the method we have to call the admin to set the proper mode

//then we add the joke/proverbs into the arraylist if its empty, we shuffle the joke/proverbs and send them one by one per user request

static void getJokesProverb(String userName, PrintStream sendClientResponse) {

try {

//the choice of mode is read by AdminWorker and depending on that AdminWorker notify Worker for that mode

if(mode==0){//joke mode

if (jokes.isEmpty()){ //arraylist is empty

putJokes(userName); // call the method to add the jokes in the arraylist and we pass the username as parameter

}

Collections.shuffle(jokes); //randomize the jokes as we shuffle them

for (String j : jokes) {

Iterator<String> it = jokes.iterator();

while(it.hasNext()) { //while it has a joke in the arraylist

if (it.next().contains(j)) { //get the joke

sendClientResponse.println(j);//send each joke to Jokeclient

sendClientResponse.flush();//clean the memory

it.remove(); //remove the joke from the arraylist

//use that to see the arraylist with the removed jokes

//see the list getting empty

System.out.println("..........\n"+jokes+"..........\n");

System.out.println("Already Send to Client the joke: \n"+j); //the removed element from the arraylist

}

}

}

}//end of mode 0

if(mode==1){//proverb mode

if (proverb.isEmpty()){ //arraylist is empty

putProverbs(userName); // call the method to add the proverbs in the arraylist and we pass the username as parameter

}

Collections.shuffle(proverb); //randomize the proverbs as we shuffle them

for (String p : proverb) {

Iterator<String> it = proverb.iterator();

while(it.hasNext()) { //while it has a proverb in the arraylist

if (it.next().contains(p)) { //get the proverb

sendClientResponse.println(p);//send each proverb to Jokeclient

sendClientResponse.flush();//clean the memory

it.remove(); //remove the proverb from the arraylist

//use that to see the arraylist with the removed jokes

//see the list getting empty

System.out.println("..........\n"+proverb+"..........\n");

System.out.println("Already Send to Client the proverb: \n"+p); //the removed element from the arraylist

}

}

}

}//end of mode 1

} catch (Exception ex) {

sendClientResponse.println("Error during I/O");

}

}//end getJokesProverb method

//jokeResponce and proverbResponce are used to add jokes and prover in the array list

//user name is passed as parameter in order to used as the required output

private static void putJokes(String userName) {

jokes.add("JA "+userName+ " : "+" Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.");

jokes.add("JB "+userName+" : "+" Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;) ");

jokes.add("JC "+userName+" : " +" What kind of shoes do ninjas wear??? Sneakers!!!");

jokes.add("JD "+userName+" : " +" How does NASA organize their company parties??? They planet!!");

}

private static void putProverbs(String userName) {

proverb.add("PA "+userName+ " : "+" The beginning is the half of every action");

proverb.add("PB "+userName+" : "+" Before you can score, u must have a goal");

proverb.add("PC "+userName+" : " +" Act quickly , think slowly ");

proverb.add("PD "+userName+" : " +" Whatever is good to know is difficult to learn");

}

}//end of worker thread

class AdminWorker extends Thread{

Socket Adminsock; //Create a local Socket value "sock"

//create the socket for communication, act as a constructor

String mode;

AdminWorker(Socket sock) {

Adminsock = sock; //pass s as a socket parameter and assign it to sock

}

public void run() {

mode="joke"; //default mode

PrintStream sendAdminClientResponse = null; //initialize the value

BufferedReader readAdminClientMessage = null; //initialize the value

try {

//connect with a socket and read and send messages to the admin client

readAdminClientMessage = new BufferedReader(new InputStreamReader(Adminsock.getInputStream()));

sendAdminClientResponse = new PrintStream(Adminsock.getOutputStream());

try {

String AdminCommand;//save the admin client response

AdminCommand = readAdminClientMessage.readLine(); //read what the admin client send, the mode

switch(AdminCommand){

case"pro": mode="proverb";

Worker.mode=1; //communicate with worker thread and notify for mode 1

sendAdminClientResponse.println("The selected mode is : " + mode); //message to JokeClientAdmin

break;

case"j": mode="joke";

Worker.mode=0; //communicate with worker thread and notify for mode 0

sendAdminClientResponse.println("The selected mode is : " + mode);

break;

default:mode="joke";

Worker.mode=0; //communicate with worker thread and notify for mode 0

sendAdminClientResponse.println("The selected mode is : " + mode);

}

} catch (IOException x) {

System.out.println("Read error");//in case of error in reading

x.printStackTrace();

}

Adminsock.close(); //we call the close method of Java.net.Socket class to terminate the connection with the specific client, not the server

System.out.println("thread admin is closed");

} catch (IOException ioe) {

System.out.println(ioe);

}

}

}

class AdminLooper implements Runnable { //make it run , not require its own thread

public static boolean adminControlSwitch = true;

public void run(){ //make it run

System.out.println("Admin thread");

int requests = 6;

int port = 5050; // different port

Socket sock;

try{

ServerSocket servsock = new ServerSocket(port, requests);

System.out.println("JokeServer connection to adminClient, listening at port 5050.\n");//print message

while (adminControlSwitch) {

sock = servsock.accept(); //start the connection

new AdminWorker (sock).start(); //start the new thread

}

servsock.close();

System.out.println("closed\n");//print exit message

}catch (IOException ioe) {System.out.println(ioe);}

}

}

public class JokeServer { //the server

public static boolean controlSwitch= true;

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

int requests = 8; //number of client requests

int port = 4545;

ServerSocket server=null;

try{

AdminLooper AL = new AdminLooper(); //when multiple threads share the same object

Thread thread = new Thread(AL); //create thread with the admin object

thread.start(); //starts the new thread

server = new ServerSocket(port, requests);//the socket take as parameter the port and request number and we create an object

System.out.println("JokeServer connection to Jokeclient, listening at port 4545.\n");//print message

} catch (Exception e) {}

while (controlSwitch) { //work for unlimited connections

Socket connection = server.accept(); //start the connection

new Worker(connection).start(); ////start the new thread

}

server.close();

System.out.println("closed\n");//print exit message

}

}

/\* Joke JokeClient Admin output \*/

/\* Check the default mode with one one client, then add 2 more clients, we can observe that the mode changed successfully and in case of wrong input we are notified \*/

!!! Athina's JokeClientAdmin !!!

Server that we try to connect is : 192.168.0.34, listening to Port: 5050

Please enter the mode u want: Select pro: for proverb mode or j: for joke mode or quit to exit the adminClient

The selected mode is : proverb

Please enter the mode u want: Select pro: for proverb mode or j: for joke mode or quit to exit the adminClient

The selected mode is : joke

Please enter the mode u want: Select pro: for proverb mode or j: for joke mode or quit to exit the adminClient

Wrong input,Press enter or w

Please enter the mode u want: Select pro: for proverb mode or j: for joke mode or quit to exit the adminClient

admin wants to exit

Exit From JokeClientAdmin

/\* First JokeClient output \*/

/\* Test with my IP and we can observe that the jokes are random but not repeated, the user name is passed one time and then we can see it in the jokes/proverb\*/

/\* notify for wrong input (not enter or quit)\*/

!!! Athina's JokeClient !!!

Server that we try to connect is : 192.168.0.34, listening to Port: 4545

Please enter your username for identification

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: JD Athina : How does NASA organize their company parties??? They planet!!

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: JC Athina : What kind of shoes do ninjas wear??? Sneakers!!!

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: JA Athina : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;)

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: JA Athina : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: PD Athina : Whatever is good to know is difficult to learn

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: PC Athina : Act quickly , think slowly

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: PA Athina : The beginning is the half of every action

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: PB Athina : Before you can score, u must have a goal

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: PC Athina : Act quickly , think slowly

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;)

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: JD Athina : How does NASA organize their company parties??? They planet!!

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: Wrong input,Press enter or w

The user : Athina asks for a joke or proverb ,Please press enter to confirm or quit to exit: Wrong input,Press enter or w

Exit from JokeClient

/\* Second JokeClient output \*/

/\* Try for default connection, observe that it doent interfere with other client communication (can see more on server side)\*/

!!! Athina's JokeClient !!!

Server that we try to connect is : localhost, listening to Port: 4545

Please enter your username for identification

The user : Maria asks for a joke or proverb ,Please press enter to confirm or quit to exit: PA Maria : The beginning is the half of every action

The user : Maria asks for a joke or proverb ,Please press enter to confirm or quit to exit: PB Maria : Before you can score, u must have a goal

The user : Maria asks for a joke or proverb ,Please press enter to confirm or quit to exit: PD Maria : Whatever is good to know is difficult to learn

The user : Maria asks for a joke or proverb ,Please press enter to confirm or quit to exit: JC Maria : What kind of shoes do ninjas wear??? Sneakers!!!

The user : Maria asks for a joke or proverb ,Please press enter to confirm or quit to exit: user wants to exit

Exit from JokeClient

/\* Third JokeClient output \*/

/\* Try for default connection, observe that it doent interfere with other client communication (can see more on server side)\*/

!!! Athina's JokeClient !!!

Server that we try to connect is : localhost, listening to Port: 4545

Please enter your username for identification

The user : Liza asks for a joke or proverb ,Please press enter to confirm or quit to exit: PD Liza : Whatever is good to know is difficult to learn

The user : Liza asks for a joke or proverb ,Please press enter to confirm or quit to exit: PB Liza : Before you can score, u must have a goal

The user : Liza asks for a joke or proverb ,Please press enter to confirm or quit to exit: PA Liza : The beginning is the half of every action

The user : Liza asks for a joke or proverb ,Please press enter to confirm or quit to exit: JC Liza : What kind of shoes do ninjas wear??? Sneakers!!!

The user : Liza asks for a joke or proverb ,Please press enter to confirm or quit to exit: JB Liza : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;)

The user : Liza asks for a joke or proverb ,Please press enter to confirm or quit to exit: JA Liza : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.

The user : Liza asks for a joke or proverb ,Please press enter to confirm or quit to exit: user wants to exit

Exit from JokeClient

/\* JokeServer output \*/

/\* Try for connection to the first client and connect with admin, observe that it reads the username and save a cookie for the first client\*/

/\* Can observer the list changes and what is removed every time (it means this goes to client), see the change to proverb, run as it should\*/

Admin thread

JokeServer connection to adminClient, listening at port 5050.

JokeServer connection to Jokeclient, listening at port 4545.

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;) , JC Athina : What kind of shoes do ninjas wear??? Sneakers!!!, JA Athina : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.]..........

Already Send to Client the joke:

JD Athina : How does NASA organize their company parties??? They planet!!

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;) , JA Athina : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.]..........

Already Send to Client the joke:

JC Athina : What kind of shoes do ninjas wear??? Sneakers!!!

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;) ]..........

Already Send to Client the joke:

JA Athina : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[]..........

Already Send to Client the joke:

JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;)

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[JC Athina : What kind of shoes do ninjas wear??? Sneakers!!!, JD Athina : How does NASA organize their company parties??? They planet!!, JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;) ]..........

Already Send to Client the joke:

JA Athina : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.

Worker thread closed

thread admin is closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[PB Athina : Before you can score, u must have a goal, PA Athina : The beginning is the half of every action, PC Athina : Act quickly , think slowly ]..........

Already Send to Client the proverb:

PD Athina : Whatever is good to know is difficult to learn

Worker thread closed

/\* Try for connection to the secont client and observe that it reads the different username and save a different cookie for the second client\*/

/\* Can observer no interference, run as it should\*/

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[PA Athina : The beginning is the half of every action, PB Athina : Before you can score, u must have a goal]..........

Already Send to Client the proverb:

PC Athina : Act quickly , think slowly

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[PB Athina : Before you can score, u must have a goal]..........

Already Send to Client the proverb:

PA Athina : The beginning is the half of every action

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[]..........

Already Send to Client the proverb:

PB Athina : Before you can score, u must have a goal

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[PD Athina : Whatever is good to know is difficult to learn, PB Athina : Before you can score, u must have a goal, PA Athina : The beginning is the half of every action]..........

Already Send to Client the proverb:

PC Athina : Act quickly , think slowly

Worker thread closed

connection to the client with Username Maria

My cookie is 8773593c-b53c-41f9-ae5a-13650dac8c0a

..........

[PC Maria : Act quickly , think slowly , PD Maria : Whatever is good to know is difficult to learn, PB Maria : Before you can score, u must have a goal]..........

Already Send to Client the proverb:

PA Maria : The beginning is the half of every action

Worker thread closed

connection to the client with Username Maria

My cookie is 8773593c-b53c-41f9-ae5a-13650dac8c0a

..........

[PD Maria : Whatever is good to know is difficult to learn, PC Maria : Act quickly , think slowly ]..........

Already Send to Client the proverb:

PB Maria : Before you can score, u must have a goal

Worker thread closed

/\* Try for connection to the third client and observe no interference, run as it should\*/

connection to the client with Username Maria

My cookie is 8773593c-b53c-41f9-ae5a-13650dac8c0a

..........

[PC Maria : Act quickly , think slowly , PD Maria : Whatever is good to know is difficult to learn, PB Maria : Before you can score, u must have a goal]..........

Already Send to Client the proverb:

PA Maria : The beginning is the half of every action

Worker thread closed

connection to the client with Username Maria

My cookie is 8773593c-b53c-41f9-ae5a-13650dac8c0a

..........

[PD Maria : Whatever is good to know is difficult to learn, PC Maria : Act quickly , think slowly ]..........

Already Send to Client the proverb:

PB Maria : Before you can score, u must have a goal

Worker thread closed

connection to the client with Username Liza

My cookie is 6570931f-ffa6-4146-936f-3353a27ba5fb

..........

[PA Liza : The beginning is the half of every action, PC Liza : Act quickly , think slowly , PB Liza : Before you can score, u must have a goal]..........

Already Send to Client the proverb:

PD Liza : Whatever is good to know is difficult to learn

Worker thread closed

connection to the client with Username Liza

My cookie is 6570931f-ffa6-4146-936f-3353a27ba5fb

..........

[PC Liza : Act quickly , think slowly , PA Liza : The beginning is the half of every action]..........

Already Send to Client the proverb:

PB Liza : Before you can score, u must have a goal

Worker thread closed

connection to the client with Username Liza

My cookie is 6570931f-ffa6-4146-936f-3353a27ba5fb

..........

[PC Liza : Act quickly , think slowly ]..........

Already Send to Client the proverb:

PA Liza : The beginning is the half of every action

Worker thread closed

connection to the client with Username Maria

My cookie is 8773593c-b53c-41f9-ae5a-13650dac8c0a

..........

[PC Maria : Act quickly , think slowly ]..........

Already Send to Client the proverb:

PD Maria : Whatever is good to know is difficult to learn

Worker thread closed

thread admin is closed

connection to the client with Username Liza

My cookie is 6570931f-ffa6-4146-936f-3353a27ba5fb

..........

[JA Liza : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all., JD Liza : How does NASA organize their company parties??? They planet!!, JB Liza : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;) ]..........

Already Send to Client the joke:

JC Liza : What kind of shoes do ninjas wear??? Sneakers!!!

Worker thread closed

connection to the client with Username Liza

My cookie is 6570931f-ffa6-4146-936f-3353a27ba5fb

..........

[JA Liza : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all., JD Liza : How does NASA organize their company parties??? They planet!!]..........

Already Send to Client the joke:

JB Liza : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;)

Worker thread closed

connection to the client with Username Liza

My cookie is 6570931f-ffa6-4146-936f-3353a27ba5fb

..........

[JD Liza : How does NASA organize their company parties??? They planet!!]..........

Already Send to Client the joke:

JA Liza : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[JC Athina : What kind of shoes do ninjas wear??? Sneakers!!!, JD Athina : How does NASA organize their company parties??? They planet!!]..........

Already Send to Client the joke:

JB Athina : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;)

Worker thread closed

connection to the client with Username Athina

My cookie is bfad6afe-dbb8-4949-bde6-44c51953d895

..........

[JC Athina : What kind of shoes do ninjas wear??? Sneakers!!!]..........

Already Send to Client the joke:

JD Athina : How does NASA organize their company parties??? They planet!!

Worker thread closed

connection to the client with Username Maria

My cookie is 8773593c-b53c-41f9-ae5a-13650dac8c0a

..........

[JD Maria : How does NASA organize their company parties??? They planet!!, JB Maria : Why does Snoop Dogg carry an umbrella??? Fo’ drizzle!! ;) , JA Maria : Can a kangaroo jump higher than a house???? Of course, a house does not jump at all.]..........

Already Send to Client the joke:

JC Maria : What kind of shoes do ninjas wear??? Sneakers!!!

Worker thread closed