import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

import java.net.Socket;

/\*\*

\*

\* @author Administrator

\*/

public class TCPclient {

public static void main(String[] args) {

try{

//create client socket listening to port 1212

Socket client = new Socket("localhost", 1212);

System.out.println("Enter message...(decode/encode message, disconnected with .or empty string)\n");

// create bufferedReader stream from keyboard

BufferedReader sysin = new BufferedReader(new InputStreamReader(

System.in));

// create outputstream for client socket

PrintWriter out = new PrintWriter(client.getOutputStream(), true);

// create inputstream from server socket

BufferedReader in = new BufferedReader(new InputStreamReader(client

.getInputStream()));

while(true){

String message = sysin.readLine();

//if user type ’ . ’ or empty input, end loop and close connection

if(message.indexOf('.')!=-1||message== null ||message.isEmpty())break;

out.println(message);

// print out the original message

System.out.println("[Client]: " + message);

// print out the decoded message from server

System.out.println("[Server]: " + in.readLine());

System.out.println("\n"+"Enter message...");

}

client.close();

} catch (IOException ex) {

System.out.println("Error:"+ex);

}

}

}

import java.io.BufferedReader;

import java.net.ServerSocket;

import java.net.Socket;

import java.io.InputStreamReader;

import java.io.PrintWriter;

/\*\*

\*

\* @author Administrator

\*/

public class TCPserver {

public static void main(String[] args){

try {

System.out.println("Connected!" );

// creat TCPserver listening to port 1212

ServerSocket server = new ServerSocket(1212);

Socket ServSocket= server.accept();

//create inputstream from client socket

BufferedReader Servin = new BufferedReader(new InputStreamReader(ServSocket

.getInputStream()));

//create outputstream to client socket

PrintWriter servout = new PrintWriter(ServSocket.getOutputStream(),true);

while (true) {

String servmessage=Servin.readLine();

// no entry is interpreted as terminate translation

if (servmessage== null ||servmessage.isEmpty()) {

servout.println("you are disconnected!");

break;

}

// server accept flag as decode/encode indicator

if (servmessage.startsWith("decode")) {

String servmessage1= servmessage.substring(6);

servmessage1=servmessage1.trim();

StringBuilder result = new StringBuilder(servmessage1);

for(int i = 0; i < servmessage1.length(); i++) {

char ch=servmessage1.charAt(i);

int ascii=(int) ch+1;

result.setCharAt(i,(char)ascii);

}

servout.println(result);

}

if (servmessage.startsWith("encode")) {

String servmessage1= servmessage.substring(6);

servmessage1=servmessage1.trim();

StringBuilder result = new StringBuilder(servmessage1);

for(int i = 0; i < servmessage1.length(); i++) {

char ch=servmessage1.charAt(i);

int ascii=(int) ch-1;

result.setCharAt(i,(char)ascii);

}

servout.println(result);

}

// server disconnected when receive empty message from client

else if(!servmessage.startsWith("encode")&&!servmessage.startsWith("decode"))

servout.println("Please start with encode/decode!");

}

server.close();

} catch (Exception e) {

System.out.println("Error: " + e);

}

}

}



