Assignment 4: write three programs in Java programming language to achieve the following functions separately.

1. Given a web page address, download the web page with URLConnection and save it in the computer as a file.
2. Send the file in question 1 to the server with Socket. The server side supports multi-threading, that is, the server can receive files from multiple clients at the same time.
3. Realize the chat between client and server based on UDP with Socket.

1.

import java.io.\*;

import java.net.\*;

public class Main {

public static void downloadWebPage(String urlString, String fileName) {

try {

URL url = new URL(urlString);

URLConnection conn = url.openConnection();

InputStream in = conn.getInputStream();

BufferedReader reader = new BufferedReader(new InputStreamReader(in));

PrintWriter writer = new PrintWriter(fileName, "UTF-8");

String line;

while ((line = reader.readLine()) != null) {

writer.println(line);

}

in.close();

reader.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

// Test the program

public static void main(String[] args) {

downloadWebPage("https://yle.fi/", "yle.html");

}

}

2.

import java.io.\*;

import java.net.\*;

public class Main {

private static final String SERVER\_IP = "127.0.0.1";

private static final int SERVER\_PORT = 8888;

public static void sendFile(String fileName) {

// Create a new thread to send the file

new Thread(() -> {

try {

Socket socket = new Socket(SERVER\_IP, SERVER\_PORT);

OutputStream out = socket.getOutputStream();

InputStream in = new FileInputStream(fileName);

byte[] buffer = new byte[1024];

int len;

while ((len = in.read(buffer)) > 0) {

out.write(buffer, 0, len);

}

in.close();

out.close();

socket.close();

} catch (IOException e) {

e.printStackTrace();

}

}).start();

}

// Test the program

public static void main(String[] args) {

sendFile("yle.html");

}

}

3.

import java.io.\*;

import java.net.\*;

import java.util.\*;

public class UDPClient {

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

DatagramSocket clientSocket = new DatagramSocket();

InetAddress IPAddress = InetAddress.getByName("localhost");

byte[] sendData = new byte[1024];

byte[] receiveData = new byte[1024];

Scanner inFromUser = new Scanner(System.in);

String sentence = inFromUser.nextLine();

sendData = sentence.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

clientSocket.send(sendPacket);

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

clientSocket.receive(receivePacket);

String modifiedSentence = new String(receivePacket.getData());

System.out.println("FROM SERVER:" + modifiedSentence);

clientSocket.close();

}

}