Tristan BRAU
Léo PHAV

Java pour les réseaux - TP 2

1.1 UDP Server

Tristan BRAU Part:

```java
package TP2;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.io.IOException;
import java.net.DatagramPacket;

public class UDP_Server{
    private DatagramSocket datagramSocket;// Doit tj être ouvert pour recevoir ou envoyer contrairement a datagrampacket qui n'a pas besoin d'êt
    private byte[] buffer = new byte[256];


    public UDP_Server(DatagramSocket datagramSocket){
        this.datagramSocket = datagramSocket;
    }

    public void receiveThenSend(){
        while (true){
            try{
                DatagramPacket datagramPacket = new DatagramPacket(buffer, buffer.length);
                datagramSocket.receive(datagramPacket);
                InetAddress inetAddress = datagramPacket.getAddress();//récupérer l'adresse IP de l'expéditeur du paquet UDP reçu.
                int port = datagramPacket.getPort();
                String messagefromClient = new String(datagramPacket.getData(),datagramPacket.getOffset(),datagramPacket.getLength());
                System.out.println("Message from client : "+ messagefromClient + "\n");
                datagramPacket = new DatagramPacket(buffer, buffer.length, inetAddress,port);
                datagramSocket.send(datagramPacket);
            }catch (IOException  e) {
                e.printStackTrace();
                break;
            }
        }
    }

    Run | Debug
    public static void main(String[] args) throws SocketException{
        DatagramSocket datagramSocket = new DatagramSocket(8080);// port = 8080
        UDP_Server  server = new UDP_Server(datagramSocket);
        server.receiveThenSend();
    }
```

To test this server, you can first use the netcat command (nc), from another terminal.

```
┌──(brau㉿Brau)-[~/Documents/Visual_studio_code/Java_pour_les_réseaux]
└─$ java TP2/UDP_Server

Message from client : dd


Message from client : dd


Message from client : ddd
```

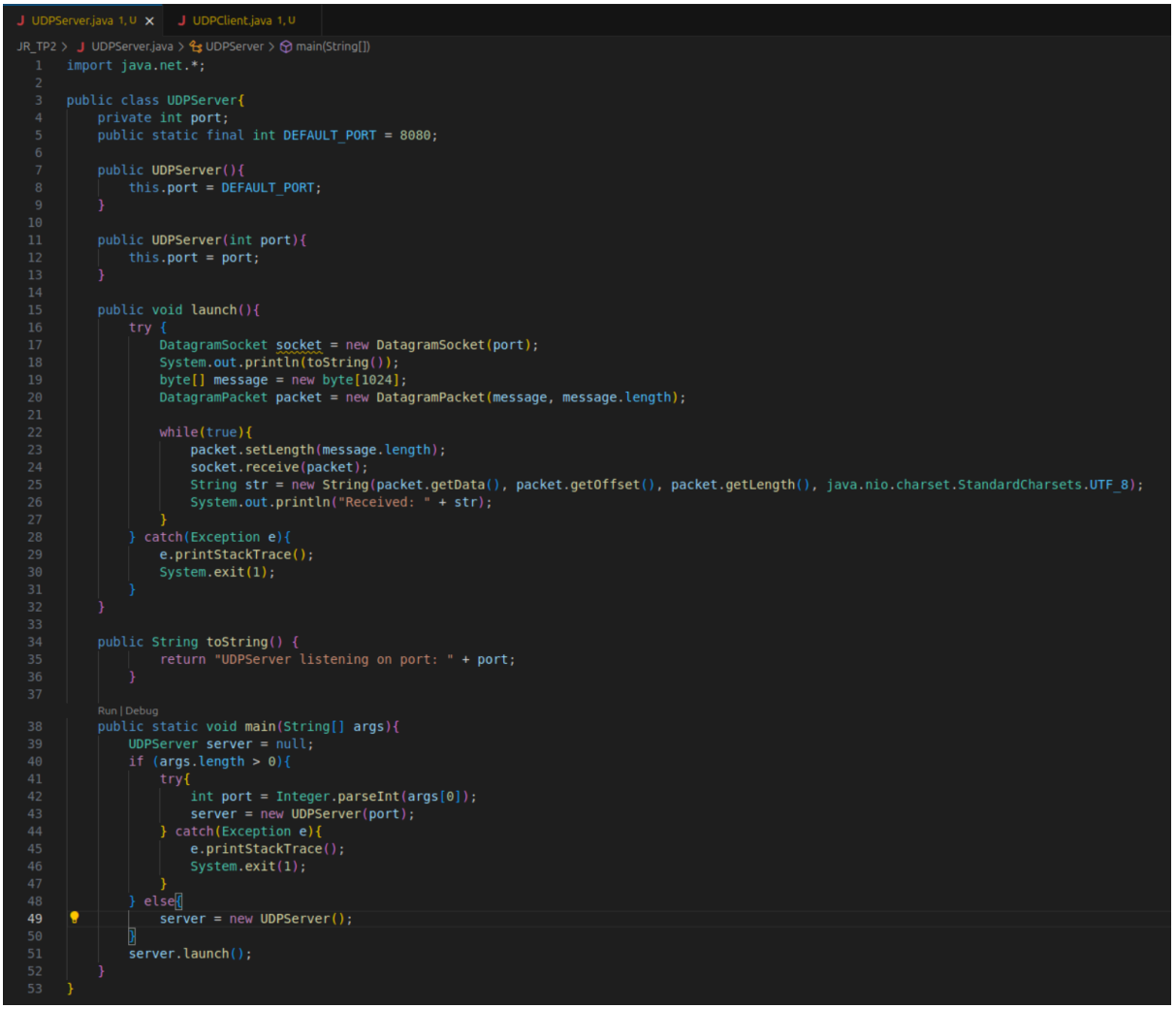
Tristan BRAU
Léo PHAV

```
┌──(brau㉿Brau)-[~]
└─$ nc -u 127.0.0.1 8080

dd
dd
dd
dd
ddd
ddd
```

Léo PHAV Part:

The program **UDPServer.java** creates a simple UDP server that listens for and displays messages sent by a client. When launched, it opens a UDP socket on a specific port, either the default port 8080 or one provided as an argument. The server continuously waits for incoming datagram packets, decodes the received data in UTF-8, and prints the message to the console.



```java
import java.net.*;

public class UDPServer{
    private int port;
    public static final int DEFAULT_PORT = 8080;

    public UDPServer(){
        this.port = DEFAULT_PORT;
    }

    public UDPServer(int port){
        this.port = port;
    }

    public void launch(){
        try {
            DatagramSocket socket = new DatagramSocket(port);
            System.out.println(toString());
            byte[] message = new byte[1024];
            DatagramPacket packet = new DatagramPacket(message, message.length);

            while(true){
                packet.setLength(message.length);
                socket.receive(packet);
                String str = new String(packet.getData(), packet.getOffset(), packet.getLength(), java.nio.charset.StandardCharsets.UTF_8);
                System.out.println("Received: " + str);
            }
        } catch(Exception e){
            e.printStackTrace();
            System.exit(1);
        }
    }

    public String toString() {
            return "UDPServer listening on port: " + port;
        }

    Run | Debug
    public static void main(String[] args){
        UDPServer server = null;
        if (args.length > 0){
            try{
                int port = Integer.parseInt(args[0]);
                server = new UDPServer(port);
            } catch(Exception e){
                e.printStackTrace();
                System.exit(1);
            }
        } else{
            server = new UDPServer();
        }
        server.launch();
    }
}
```

Tristan BRAU
Léo PHAV

Once I finished the code, I launched the server and tried to send informations with the command: *nc -u localhost [port]*

First, I tried with the default port (8080), the server received correctly the messages:

Then, with port 9090:

Tristan BRAU
Léo PHAV

1.2 UDP Client

Tristan BRAU Part:

```java
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Scanner;
import java.net.UnknownHostException;


public class UDP_Client {
    private DatagramSocket datagramSocket;
    private InetAddress inetAdress;
    private byte[] buffer ;

    public UDP_Client(DatagramSocket datagramSocket,InetAddress inetAdress){
        this.datagramSocket = datagramSocket;
        this.inetAdress = inetAdress;
    }

    public void sendThenReceive(){
         Scanner scanner = new Scanner(System.in);
         while(true){
            try{
                String messageToSend = scanner.nextLine();
                buffer = messageToSend.getBytes();
                DatagramPacket datagramPacket = new DatagramPacket(buffer, buffer.length,inetAdress,8080);
                datagramSocket.send(datagramPacket);
                datagramSocket.receive(datagramPacket);
                String messageFromServer = new String(datagramPacket.getData(),0,datagramPacket.getLength());
                System.out.println("The server say you say" + messageFromServer);
            } catch(IOException e){
                    e.printStackTrace();
                    break;
            }
        }
    }

    Run | Debug
    public static void main(String[] args) throws SocketException, UnknownHostException{
        DatagramSocket datagramSocket =  new DatagramSocket();
        InetAddress inetAddress = InetAddress.getByName("localhost");
        UDP_Client client = new UDP_Client(datagramSocket,inetAddress);
        System.out.println("Send datagram packets to a server UDP ");
        client.sendThenReceive();
    }
}
```

```
┌──(brau㉿Brau)-[~/Documents/Visual_studio_code/Java_pour_les_réseaux]
└─$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDeta
les_réseaux_c7927af9/bin TP2.UDP_Server
Message from client : Hi How are you server today
```

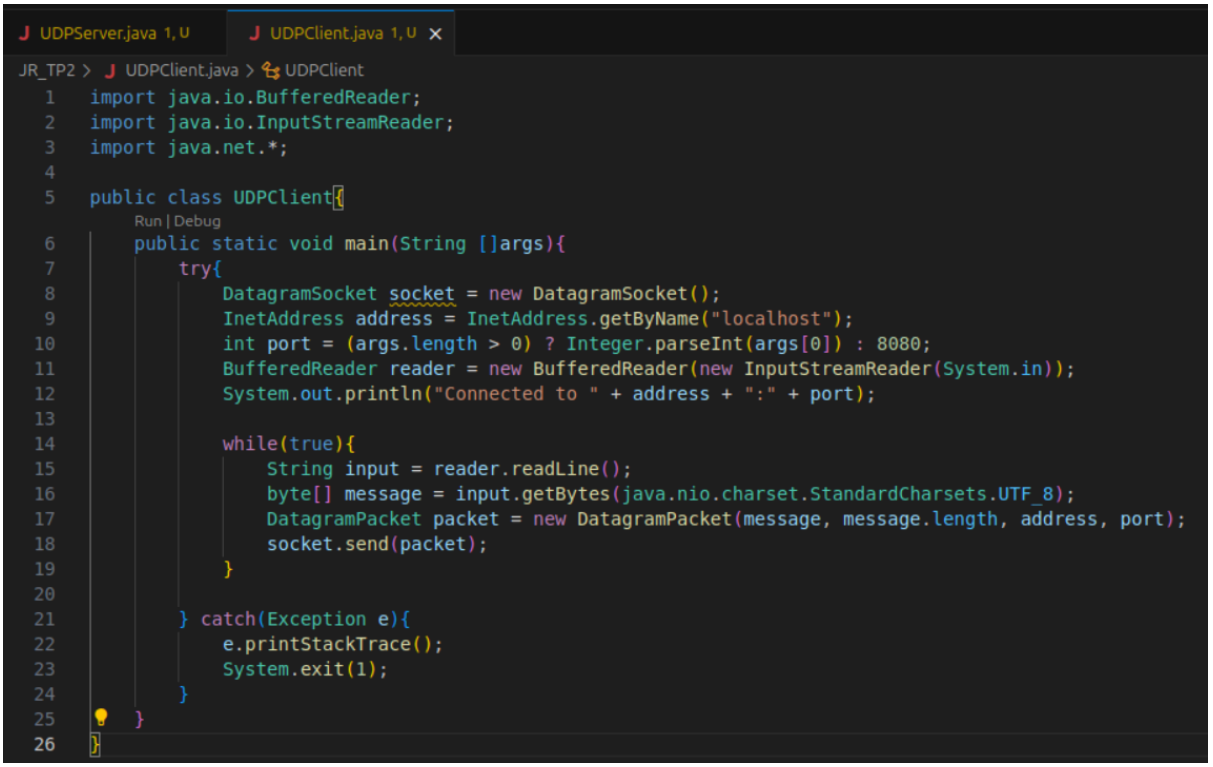
```
┌──(brau㉿Brau)-[~/Documents/Visual_studio_code/Java_pour_les_réseaux]
└─$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetai
les_réseaux_c7927af9/bin TP2.UDP_Client
Send datagram packets to a server UDP
Hi How are you server today
The server say you sayHi How are you server today
```

We successfully receive the messages sent from the client on our UDP server, and the server correctly sends them back to the client.

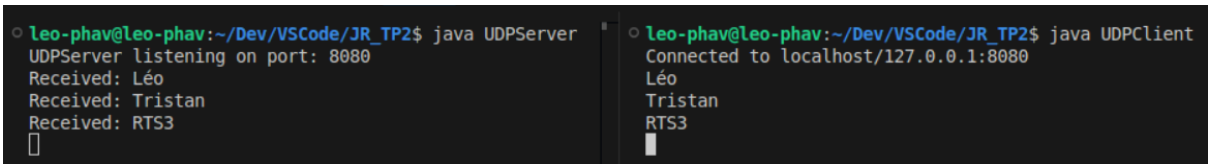
Tristan BRAU
Léo PHAV

Léo PHAV Part:

The program **UDPClient.java** implements a simple client that sends messages to a UDP server. When executed, it creates a UDP socket and connects to a server located at localhost on a specific port — either the default port 8080 or a port provided as a command-line argument. The program then reads user input from the keyboard in a continuous loop. Each line entered is converted into bytes encoded in UTF-8 and sent to the server using a DatagramPacket.



Here, I did the same manipulation but with the UDP Client:



All messages have been received with no errors.

1.3 Additional Parts

Tristan BRAU Part:

Add sequence numbers to messages

```
┌──(brau㉿Brau)-[~/Documents/Visual_studio_code/Java_pour_les_réseaux]
└─$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCodeDetailsInExceptionMes
les_réseaux_c7927af9/bin TP2.UDP_Server
Message from client : Seq = 0;ok mec

Message from client : Seq = 0 | ok mec

Message from client : Seq = 1 | ok mec

Message from client : Seq = 2 |
```

```
┌──(brau㉿Brau)-[~/Documents/Visual_studio_code/Java_pour_les_réseaux]
└─$  /usr/bin/env /usr/lib/jvm/java-21-openjdk-amd64/bin/java -XX:+ShowCod
les_réseaux_c7927af9/bin TP2.UDP_Client
Send datagram packets to a server UDP
ok mec
Seq = 0 | The server say you say Seq = 0 | ok mec
ok mec
Seq = 1 | The server say you say Seq = 1 | ok mec

Seq = 2 | The server say you say Seq = 2 |
```

Léo PHAV Part:

To implement a sequence number, I created a new variable:
String seqMessage = sequenceNum + " | " + input;

I added a sequence number into my messages but when I sent empty messages, the sequence number still increment:

```
leo-phav@leo-phav:~/Dev/VSCode/JR_TP2$ java UDPServer
UDPServer listening on port: 8080
Received: 1 | Hi
Received: 2 | Léo
Received: 3 | RTS
Received: 4 |
Received: 5 |
Received: 6 |
Received: 7 |
Received: 8 | 12345
```

```
leo-phav@leo-phav:~/Dev/VSCode/JR_TP2$ java UDPClient
Connected to localhost/127.0.0.1:8080
Hi
Léo
RTS



12345
```

By adding this command before I create seqMessage variable, I solve the issue:
if (input == null || input.isEmpty()) continue;

Tristan BRAU
Léo PHAV