NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
“IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE”

Faculty of Informatics and Computer Engineering

Department of Computer Engineering

Distributed Information Systems

Lab №1 Java Networking

Through the classes in java.net, Java programs can use TCP or UDP to communicate over the Internet. The URL, URLConnection, Socket, and ServerSocket
classes all use TCP to communicate over the network. The DatagramPacket, DatagramSocket, and MulticastSocket classes are for use with UDP.

Student, group IM-14 FIOT

MEHMET KULUBECİOGLU

Reviewer YULİA TİMOFEEVA

Kyiv – 2024

Task

1) Write and test Client and Server projects

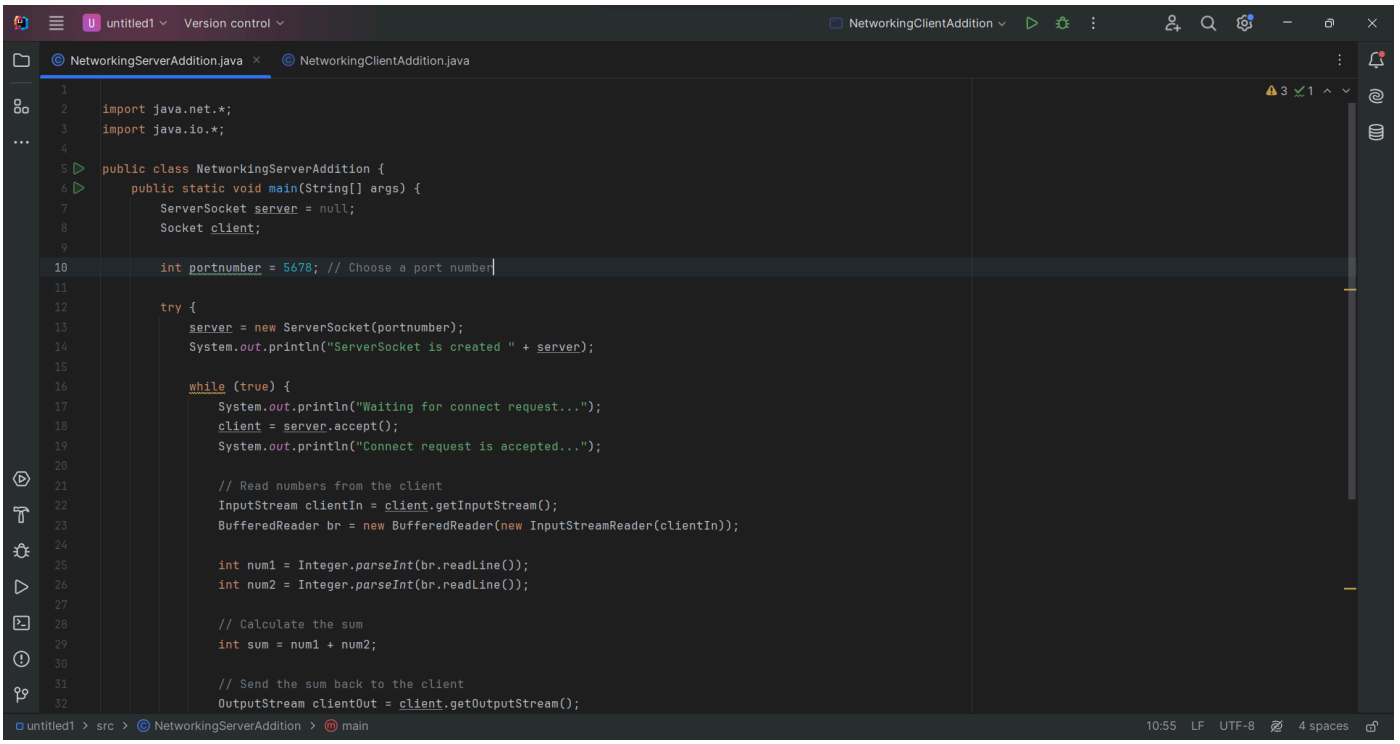
2) Write and test Multicast Client and Server projects

3) Create your own NetworkingServer and NetworkingClient applications as

following. The client send two numbers and the server sends back addition of the two numbers.

4) Write report with screenshots from tasks 1-3 and your code from task 3

MY FULL CODES:



```java
import java.net.*;
import java.io.*;

public class NetworkingServerAddition {
    public static void main(String[] args) {
        ServerSocket server = null;
        Socket client;

        int portnumber = 5678; // Choose a port number

        try {
            server = new ServerSocket(portnumber);
            System.out.println("ServerSocket is created " + server);

            while (true) {
                System.out.println("Waiting for connect request...");
                client = server.accept();
                System.out.println("Connect request is accepted...");

                // Read numbers from the client
                InputStream clientIn = client.getInputStream();
                BufferedReader br = new BufferedReader(new InputStreamReader(clientIn));

                int num1 = Integer.parseInt(br.readLine());
                int num2 = Integer.parseInt(br.readLine());

                // Calculate the sum
                int sum = num1 + num2;

                // Send the sum back to the client
                OutputStream clientOut = client.getOutputStream();
```

```java
            client = server.accept();
            System.out.println("Connect request is accepted...");

            // Read numbers from the client
            InputStream clientIn = client.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(clientIn));

            int num1 = Integer.parseInt(br.readLine());
            int num2 = Integer.parseInt(br.readLine());

            // Calculate the sum
            int sum = num1 + num2;

            // Send the sum back to the client
            OutputStream clientOut = client.getOutputStream();
            PrintWriter pw = new PrintWriter(clientOut, autoFlush: true);
            pw.println("Sum: " + sum);

            // Close sockets
            client.close();
        }
    } catch (IOException | NumberFormatException e) {
        e.printStackTrace();
    }
  }
}
```

```java
    try {
        server = new ServerSocket(portnumber);
        System.out.println("ServerSocket is created " + server);

        while (true) {
            System.out.println("Waiting for connect request...");
            client = server.accept();
            System.out.println("Connect request is accepted...");

            // Read numbers from the client
            InputStream clientIn = client.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(clientIn));

            int num1 = Integer.parseInt(br.readLine());
            int num2 = Integer.parseInt(br.readLine());

            // Calculate the sum
            int sum = num1 + num2;

            // Send the sum back to the client
            OutputStream clientOut = client.getOutputStream();
            PrintWriter pw = new PrintWriter(clientOut, autoFlush: true);
            pw.println("Sum: " + sum);

            // Close sockets
            client.close();
        }
    } catch (IOException | NumberFormatException e) {
        e.printStackTrace();
    }
  }
}
```

```
"C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.4\lib\idea_rt.jar=51362:C:\Program Files\JetBrains
ServerSocket is created ServerSocket[addr=0.0.0.0/0.0.0.0,localport=5678]
Waiting for connect request...
```

```java
import java.io.*;
import java.net.*;

public class NetworkingClientAddition {
    public static void main(String[] args) {
        Socket client = null;
        int portnumber = 5678; // Use the same port number as the server

        try {
            // Create a client socket
            client = new Socket(InetAddress.getLocalHost(), portnumber);
            System.out.println("Client socket is created " + client);

            // Create output stream
            OutputStream clientOut = client.getOutputStream();
            PrintWriter pw = new PrintWriter(clientOut, autoFlush: true);

            // Create input stream
            InputStream clientIn = client.getInputStream();
            BufferedReader br = new BufferedReader(new InputStreamReader(clientIn));

            // Read two numbers from the user
            BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter the first number:");
            int num1 = Integer.parseInt(stdIn.readLine().trim());
            System.out.println("Enter the second number:");
            int num2 = Integer.parseInt(stdIn.readLine().trim());

            // Send numbers to the server
            pw.println(num1);
            pw.println(num2);
```


```java
            BufferedReader br = new BufferedReader(new InputStreamReader(clientIn));

            // Read two numbers from the user
            BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter the first number:");
            int num1 = Integer.parseInt(stdIn.readLine().trim());
            System.out.println("Enter the second number:");
            int num2 = Integer.parseInt(stdIn.readLine().trim());

            // Send numbers to the server
            pw.println(num1);
            pw.println(num2);
```


```java

            // Receive and print the sum from the server
            String sumMessage = br.readLine();
            System.out.println("Server response: " + sumMessage);

            // Close sockets
            pw.close();
            br.close();
            client.close();
        } catch (IOException | NumberFormatException e) {
            e.printStackTrace();
        }
    }
}
```

```java
            BufferedReader br = new BufferedReader(new InputStreamReader(clientIn));

            // Read two numbers from the user
            BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Enter the first number:");
            int num1 = Integer.parseInt(stdIn.readLine().trim());
            System.out.println("Enter the second number:");
            int num2 = Integer.parseInt(stdIn.readLine().trim());

            // Send numbers to the server
            pw.println(num1);
            pw.println(num2);

            // Receive and print the sum from the server
            String sumMessage = br.readLine();
            System.out.println("Server response: " + sumMessage);

            // Close sockets
            pw.close();
            br.close();
            client.close();
        } catch (IOException | NumberFormatException e) {
            e.printStackTrace();
        }
    }
}
```

```
"C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.exe" -agentlib:jdwp=transport=dt_socket,address=127.0.0.1:51379,suspend=y,server=n -javaagent:C:\Users\mehme\AppDat
Connected to the target VM, address: '127.0.0.1:51379', transport: 'socket'
Client socket is created Socket[addr=mehmet/192.168.56.1,port=5678,localport=51381]
Enter the first number:
145
Enter the second number:
364
Server response: Sum: 509
Disconnected from the target VM, address: '127.0.0.1:51379', transport: 'socket'

Process finished with exit code 0
```

NetworkingServerAddition Application:

1- Starting Server and Waiting for Connection:

- A TCP server was created using the ServerSocket and Socket classes.

```java
    public static void main(String[] args) {
        ServerSocket server = null;
        Socket client;

        int portnumber = 5678; // Choose a port number

```

```java
        server = new ServerSocket(portnumber);
        System.out.println("ServerSocket is created " + server);

```

```java
System.out.println("Waiting for connect request...");
client = server.accept();
```

2- Reading and Adding Numbers from the Client:

- Used BufferedReader and InputStream to read numbers from the client.

```java
// Read numbers from the client
InputStream clientIn = client.getInputStream();
BufferedReader br = new BufferedReader(new InputStreamReader(clientIn));

int num1 = Integer.parseInt(br.readLine());
int num2 = Integer.parseInt(br.readLine());

```

```java
// Calculate the sum
int sum = num1 + num2;

```

3- Sending Results and Closing Sockets:

- Used PrintWriter and OutputStream to send the calculated total to the client.

```java
// Send the sum back to the client
OutputStream clientOut = client.getOutputStream();
PrintWriter pw = new PrintWriter(clientOut, autoFlush: true);
pw.println("Sum: " + sum);
```

```java
// Close sockets
client.close();
```

NetworkingClientAddition Implementation:

1- Creating a Client Socket and Connecting to the Server:

- A TCP client socket was created using the Socket class.

```java
Socket client = null;
int portnumber = 5678; // Use the same port number as the server
```

```java
// Create a client socket
client = new Socket(InetAddress.getLocalHost(), portnumber);
System.out.println("Client socket is created " + client);
```

2- Sending Numbers and Receiving Response from Server:

- Two numbers received from the user are prepared to be sent to the server.

```java
// Create output stream
OutputStream clientOut = client.getOutputStream();
PrintWriter pw = new PrintWriter(clientOut, autoFlush: true);
```

```java
// Read two numbers from the user
BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));
System.out.println("Enter the first number:");
int num1 = Integer.parseInt(stdIn.readLine().trim());
System.out.println("Enter the second number:");
int num2 = Integer.parseInt(stdIn.readLine().trim());
```

- Used PrintWriter and OutputStream to send numbers to the server.

```
// Send numbers to the server
pw.println(num1);
pw.println(num2);
```

- The response from the server was received using BufferedReader and InputStream and written to the screen.

```
// Receive and print the sum from the server
String sumMessage = br.readLine();
System.out.println("Server response: " + sumMessage);
```

3- Closing Sockets:

- After the client completes the transaction, the sockets are closed.

```
// Close sockets
pw.close();
br.close();
client.close();
} catch (IOException | NumberFormatException e) {
```

Running the Application and Result:

- The server application is started on the specified port number and goes into connection waiting.
- The client application connected to a server for which it knew the server socket number, received two numbers from the user, and wrote the result to the screen.