**TUTORIAL NO. 02**

**AIM:** Implement calculator program performing operation as addition, subtraction, multiplication, division considering Multithreaded client server**.**

**THEORY:**

* **Design assumptions:**

In the the section called “The TCP Client/Server” we developed a full client/server application. The server we developed could only handle one connection at a time. It was structured to create a ConnectionHandler object for each conneciton that was made to the server, however, since the DateServer waits for the ConnectionHandler object to deal with the DateClient then it cannot handle any further connections at the same time. You can see this if you run the server and then connect to it with two clients at the same time - the server will refuse the connection as it is busy serving the current client on port 5050 and has not started listening again.

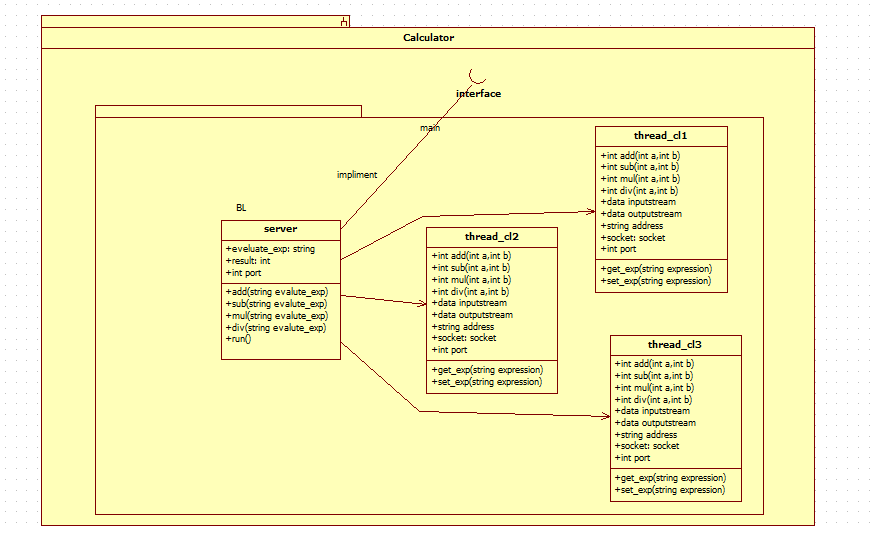
However, the Server has been structured correctly to make it easily threaded by keeping the connection handler separate from the Server. All that has to be done is to make this connection handler (the HandleConnection class) threaded. Once it is threaded the server can create a separate thread for every client that connects, where the thread deals completely with that client. When the client disconnects this thread can finish.

To make the server threaded we can take the ConnectionHandler class from the section called “The TCP Client/Server” and modify it, by making it inherit from the Thread class and over-riding the run() method of the Thread class.

The DateServer has to be modified slightly to create an object of our new ThreadedConnectionHandler class and to call the start(), method rather than calling the thread's run() method directly.

* **UML Design Architecture:**

1. **Multithreaded Client- Server:**

****

* **Code:**

1. **MyServer.java-**

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.StringTokenizer;

public class MyServer

{

public static int add(int a, int b){

int r = a + b;

return r;

}public static int sub(int a, int b){

int r = a - b;

return r;

}

public static int mul(int a, int b){

int r = a \* b;

return r;

}

public static int div(int a, int b){

int r = a / b;

return r;

}

public static void main(String args[]) throws IOException

{

// Step 1: Establish the socket connection.

ServerSocket ss = new ServerSocket(4444);

Socket s = ss.accept();

// Step 2: Processing the request.

DataInputStream dis = new DataInputStream(s.getInputStream());

DataOutputStream dos = new DataOutputStream(s.getOutputStream());

while (true)

{

// wait for input

String input = dis.readUTF();

if(input.equals("bye")){

dis.close();

dos.close();

break;

}

System.out.println("Equation received:-" + input);

int result;

// Use StringTokenizer to break the equation into operand and

// operation

StringTokenizer st = new StringTokenizer(input);

int oprnd1 = Integer.parseInt(st.nextToken());

String operation = st.nextToken();

int oprnd2 = Integer.parseInt(st.nextToken());

// perform the required operation.

if (operation.equals("+"))

{

result = add(oprnd1, oprnd2);

}

else if (operation.equals("-"))

{

result = sub(oprnd1, oprnd2);

}

else if (operation.equals("\*"))

{

result = mul(oprnd1, oprnd2);

}

else

{

result = div(oprnd1, oprnd2);

}

System.out.println("Sending the result...");

// send the result back to the client.

dos.writeUTF(Integer.toString(result));

}

}

}

1. **MyClientM.java**

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.IOException;

import java.net.InetAddress;

import java.net.Socket;

import java.net.UnknownHostException;

import java.util.Scanner;

class newthread implements Runnable{

public static void Client() throws IOException

{

InetAddress ip = InetAddress.getLocalHost();

int port = 4444;

Scanner sc = new Scanner(System.in);

// Step 1: Open the socket connection.

Socket s = new Socket(ip, port);

// Step 2: Communication-get the input and output stream

DataInputStream dis = new DataInputStream(s.getInputStream());

DataOutputStream dos = new DataOutputStream(s.getOutputStream());

while (true)

{

// Enter the equation in the form-

// "operand1 operation operand2"

System.out.print("Enter the equation in the form: ");

System.out.println("'operand operator operand'");

String inp = sc.nextLine();

if (inp.equals("bye"))

break;

// send the equation to server

dos.writeUTF(inp);

// wait till request is processed and sent back to client

String ans = dis.readUTF();

System.out.println("Answer=" + ans);

}

}

Thread t;

newthread(){

t= new Thread(this, "Client");

System.out.println("Thread" +t);

t.start();

}

public void run(){

try{

for (int i=0; i<3; i++){

System.out.println("Client " +(i+1));

try {

Client();

}

catch (IOException e) {

// Do something here

}

Thread.sleep(500);

}

}catch (InterruptedException e){

System.out.println("Client interrupted");

}

System.out.println("Exiting");

}

}

class MyClientM{

public static void main(String args[]){

new newthread();

try{

Thread.sleep(1000);

}catch (InterruptedException e){

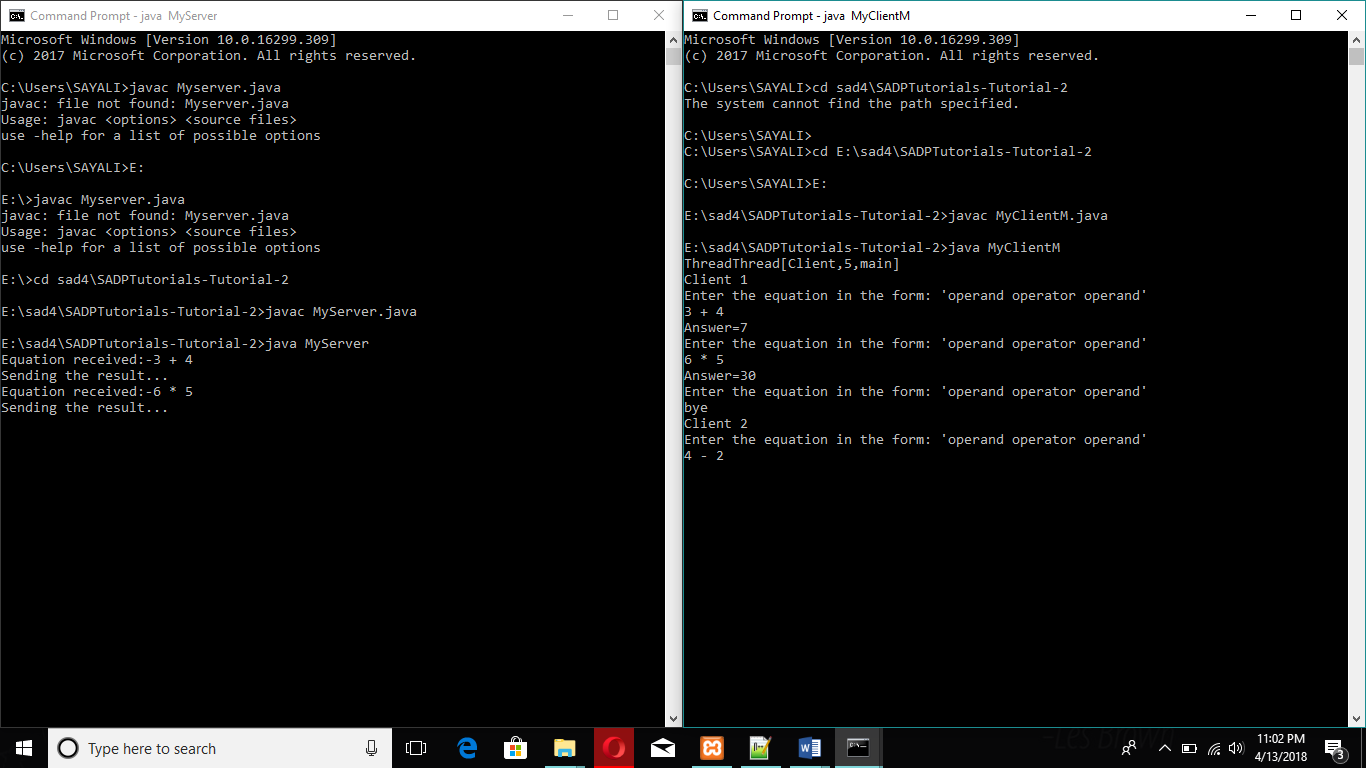
System.out.print(" ");

}

}

}

**OUTPUT:**



**OBERVATION:**

Thus we implemented calculator program performing operation as addition, subtraction, multiplication, division considering Multithreaded client server