SOCKET PROGRAMMING

CALCULATOR

MCA – SY

ATUL JAIN

ROLL NO.: 15

GR\_NO.: 17C014

SOCKET PROGRAMMIG:

A socket is one of the most fundamental technologies of **computer network programming**. Sockets allow network software applications to communicate using standard mechanisms built into network hardware and operating systems.

CALCULATOR:

Something used for making mathematical calculations, and combined with socket programming we can provide input from one system .

CLIENT CODE:

import java.net.\*;

import java.io.\*;

import java.util.Scanner;

import java.io.BufferedInputStream;

import java.io.DataInputStream;

public class GreetingClient {

public static void main(String [] args) {

String serverName = args[0];

int port = Integer.parseInt(args[1]);

try {

System.out.println("Connecting to " + serverName + " on port " + port);

Socket client = new Socket(serverName, port);

System.out.println("Just connected to " + client.getRemoteSocketAddress());

OutputStream outToServer = client.getOutputStream();

DataInputStream inputLine=null;

DataOutputStream out = new DataOutputStream(outToServer);

//DataInputStream in = new DataInputStream(inFromServer);

Scanner input = new Scanner(System.in);

out.writeUTF("Hello from " + client.getLocalSocketAddress());

InputStream inFromServer = client.getInputStream();

DataInputStream in = new DataInputStream(inFromServer);

System.out.println("Enter the First element of calculator:");

int a = input.nextInt();

out.writeUTF(Integer.toString(a));

System.out.println("Enter the Second element of calculator:");

int b = input.nextInt();

out.writeUTF(Integer.toString(b));

System.out.println("Enter the Operator of calculator:");

String c = new String(input.next());

out.writeUTF(c);

System.out.println("\nThe result is:"+in.readUTF());

client.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

SERVER CODE:

// File Name GreetingServer.java

import java.net.\*;

import java.io.\*;

import java.util.\*;

import java.io.DataInputStream;

import java.io.IOException;

import java.io.PrintStream;

import java.net.ServerSocket;

import java.net.Socket;

public class GreetingServer extends Thread {

private ServerSocket serverSocket;

// private String name,age,food;

String line;

PrintStream os;

public GreetingServer(int port) throws IOException {

serverSocket = new ServerSocket(port);

serverSocket.setSoTimeout(100000);

}

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

return(a+b);

}

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

return(a-b);

}

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

return(a/b);

}

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

return(a\*b);

}

public static int Calculator(int a,int b,String c)

{

int res=0;

if (c.equals("+")){

res =addition(a,b);

}

else if (c.equals("-")){

res = subtraction(a,b);

}

else if (c.equals("/")){

res = division(a,b);

}

else if (c.equals("\*")){

res = multiplication(a,b);

}

return(res);

}

public void run() {

while(true) {

try {

System.out.println("Waiting for client on port " +

serverSocket.getLocalPort() + "...");

Socket server = serverSocket.accept();

System.out.println("Just connected to " + server.getRemoteSocketAddress());

DataInputStream in = new DataInputStream(server.getInputStream());

System.out.println(in.readUTF());

DataOutputStream out = new DataOutputStream(server.getOutputStream());

int a = Integer.parseInt(in.readUTF());

int b = Integer.parseInt(in.readUTF());

String c = in.readUTF();

int result = Calculator(a,b,c);

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

server.close();

} catch (SocketTimeoutException s) {

System.out.println("Socket timed out!");

break;

} catch (IOException e) {

e.printStackTrace();

break;

}

}

}

public static void main(String [] args) {

int port = Integer.parseInt(args[0]);

try {

Thread t = new GreetingServer(port);

t.start();

} catch (IOException e) {

e.printStackTrace();

}

}

}

OUTPUT:



