ASURITE ID: vyaganti
ASU ID: 1223441505

SOCKET PROJECT MILESTONE

COMPUTER NETWORKS – CSE433

DESCRIPTION:

In this implementation I've come to come communicate with server and client using Java. I've used a package called DatagramPackage for this communication. In this implementation, first server will be asked for a port number, then when client program is compiled a username and port number should be give in a single line. When the client receives this information it passes the username to server and establishes a connection. When server is connected, it sends a message saying "Server is Responding…". And in the server side we keep a count on number of times a ping was made from the client using a local variable *i*. Implementation can be seen below.

SERVER IMPLEMENTATION:

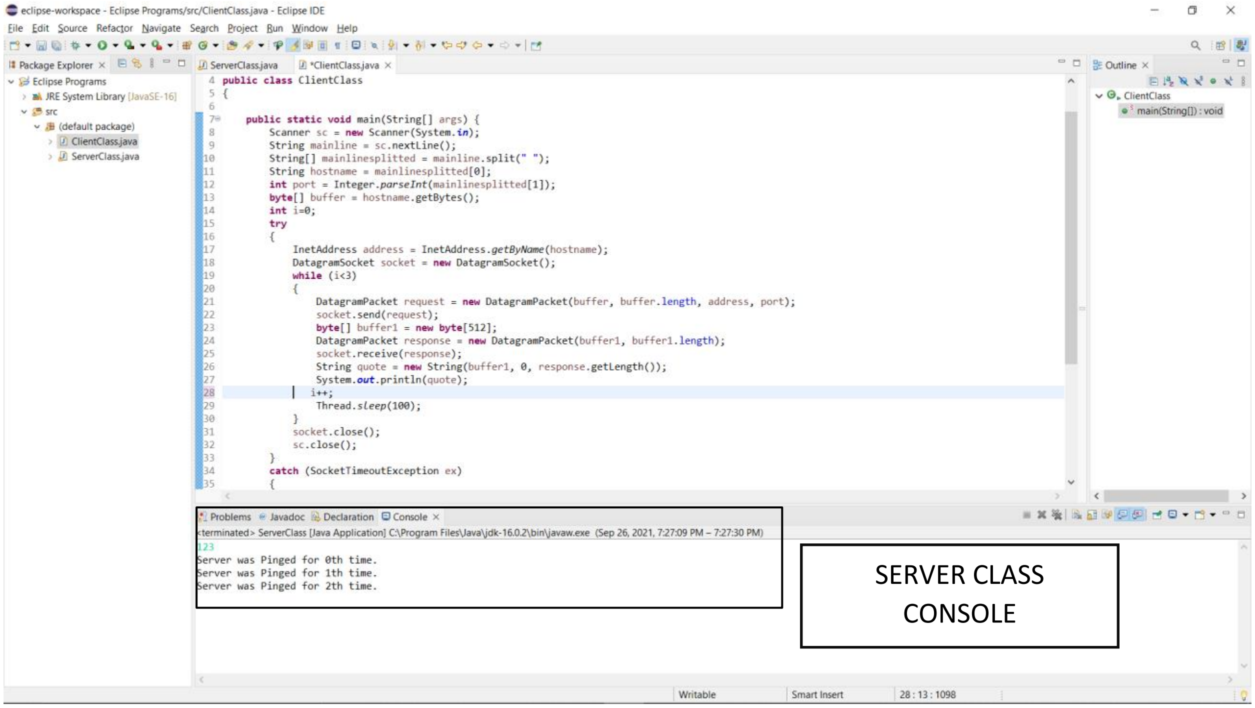
```
import java.io.*;
import java.net.*;
import java.util.*;

public class ServerClass {
    private DatagramSocket socket;
    HashMap<String, String[]> peers = new HashMap<String, String[]>();
    public ServerClass(int port) throws SocketException {
        socket = new DatagramSocket(port);
    }
    static int i=0;
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
   int port = sc.nextInt();
        try {
            ServerClass server = new ServerClass(port);
            server.service();
        } catch (SocketException ex) {
            System.out.println("Socket error: " + ex.getMessage());
        } catch (IOException ex) {
            System.out.println("I/O error: " + ex.getMessage());
        }
        sc.close();
    }
```

```
        private void service() throws IOException {

            peers.put("user1", new String[] {"127.0.0.1", "123", "456", "111"});
            peers.put("user2", new String[] {"127.0.0.2", "456", "789", "222"});
            peers.put("user3", new String[] {"127.0.0.3", "789", "123", "333"});
                while (i<3)
                {
                        String key = (String) peers.keySet().toArray()[i];
                        String hostname = key;
                        byte[] buffer1 = hostname.getBytes();
//                      InetAddress address = InetAddress.getByName(peers.get("user1")[i]);
                        DatagramPacket request = new DatagramPacket(buffer1,
buffer1.length);// address, Integer.parseInt(peers.get(key)[1]));//,address,
Integer.parseInt(peers.get("user1")[1])
                        socket.receive(request);
                        System.out.println("Server was Pinged for " + i + "th time.");
                        byte[] buffer2 = new byte[1024];
                        buffer2 = request.getData();
                        //System.out.println("Received: "+new String(buffer2));
                        String quote = "Server is Responding...";
                        byte[] buffer3 = quote.getBytes();
                        InetAddress clientAddress = request.getAddress();
                        int clientPort = request.getPort();
                        DatagramPacket response = new DatagramPacket(buffer3, buffer3.length,
clientAddress, clientPort);
                        socket.send(response);
                        i++;
                }
        }
}
```

CLIENT IMPLEMENTATION:

```
import java.io.*;
import java.net.*;
import java.util.Scanner;
public class ClientClass
{

    public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            String mainline = sc.nextLine();
            String[] mainlinesplitted = mainline.split(" ");
        String hostname = mainlinesplitted[0];
        int port = Integer.parseInt(mainlinesplitted[1]);
        byte[] buffer = hostname.getBytes();
        int i=0;
        try
        {
            InetAddress address = InetAddress.getByName(hostname);
            DatagramSocket socket = new DatagramSocket();
            while (i<3)
            {
                DatagramPacket request = new DatagramPacket(buffer, buffer.length, address, port);
                socket.send(request);
                byte[] buffer1 = new byte[512];
                DatagramPacket response = new DatagramPacket(buffer1, buffer1.length);
                socket.receive(response);
                String quote = new String(buffer1, 0, response.getLength());
                System.out.println(quote);
                i++;
                Thread.sleep(100);
            }
            socket.close();
            sc.close();
        }
        catch (SocketTimeoutException ex)
        {
            System.out.println("Timeout error: " + ex.getMessage());
            ex.printStackTrace();
        }
        catch (IOException ex)
        {
            System.out.println("Client error: " + ex.getMessage());
            ex.printStackTrace();
        }
        catch (InterruptedException ex)
        {
            ex.printStackTrace();
        }
    }
}
```

SERVER-SIDE OUTPUT:

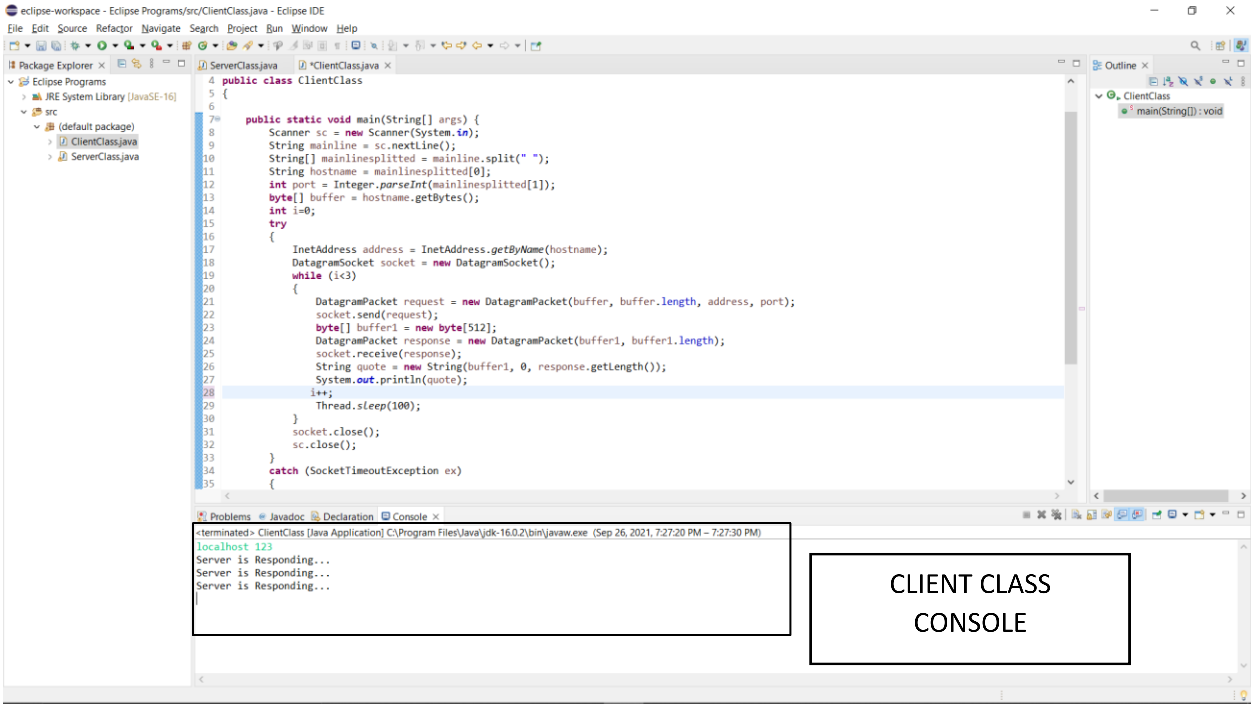


```java
public class ClientClass
{

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String mainline = sc.nextLine();
        String[] mainlinesplitted = mainline.split(" ");
        String hostname = mainlinesplitted[0];
        int port = Integer.parseInt(mainlinesplitted[1]);
        byte[] buffer = hostname.getBytes();
        int i=0;
        try
        {
            InetAddress address = InetAddress.getByName(hostname);
            DatagramSocket socket = new DatagramSocket();
            while (i<3)
            {
                DatagramPacket request = new DatagramPacket(buffer, buffer.length, address, port);
                socket.send(request);
                byte[] buffer1 = new byte[512];
                DatagramPacket response = new DatagramPacket(buffer1, buffer1.length);
                socket.receive(response);
                String quote = new String(buffer1, 0, response.getLength());
                System.out.println(quote);
               i++;
                Thread.sleep(100);
            }
            socket.close();
            sc.close();
        }
        catch (SocketTimeoutException ex)
        {
```

```
<terminated> ServerClass [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe  (Sep 26, 2021, 7:27:09 PM – 7:27:30 PM)
123
Server was Pinged for 0th time.
Server was Pinged for 1th time.
Server was Pinged for 2th time.
```

SERVER CLASS CONSOLE

CLIENT-SIDE OUTPUT:



```java
public class ClientClass
{

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String mainline = sc.nextLine();
        String[] mainlinesplitted = mainline.split(" ");
        String hostname = mainlinesplitted[0];
        int port = Integer.parseInt(mainlinesplitted[1]);
        byte[] buffer = hostname.getBytes();
        int i=0;
        try
        {
            InetAddress address = InetAddress.getByName(hostname);
            DatagramSocket socket = new DatagramSocket();
            while (i<3)
            {
                DatagramPacket request = new DatagramPacket(buffer, buffer.length, address, port);
                socket.send(request);
                byte[] buffer1 = new byte[512];
                DatagramPacket response = new DatagramPacket(buffer1, buffer1.length);
                socket.receive(response);
                String quote = new String(buffer1, 0, response.getLength());
                System.out.println(quote);
               i++;
                Thread.sleep(100);
            }
            socket.close();
            sc.close();
        }
        catch (SocketTimeoutException ex)
        {
```

```
<terminated> ClientClass [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe  (Sep 26, 2021, 7:27:20 PM – 7:27:30 PM)
localhost 123
Server is Responding...
Server is Responding...
Server is Responding...
```

CLIENT CLASS CONSOLE

ASURITE ID: vyaganti
ASU ID: 1223441505

PUBLIC VISIBILITY LOCATIONS:

1. GITHUB REPOSITORY: https://github.com/vyaganti1/scasucn
2. YOUTUBE VIDEO LINK: https://www.youtube.com/watch?v=ss5CuDyUnPg

NOTE:

- In github repository, scasucn stands for Server Client ASU Computer Networks.