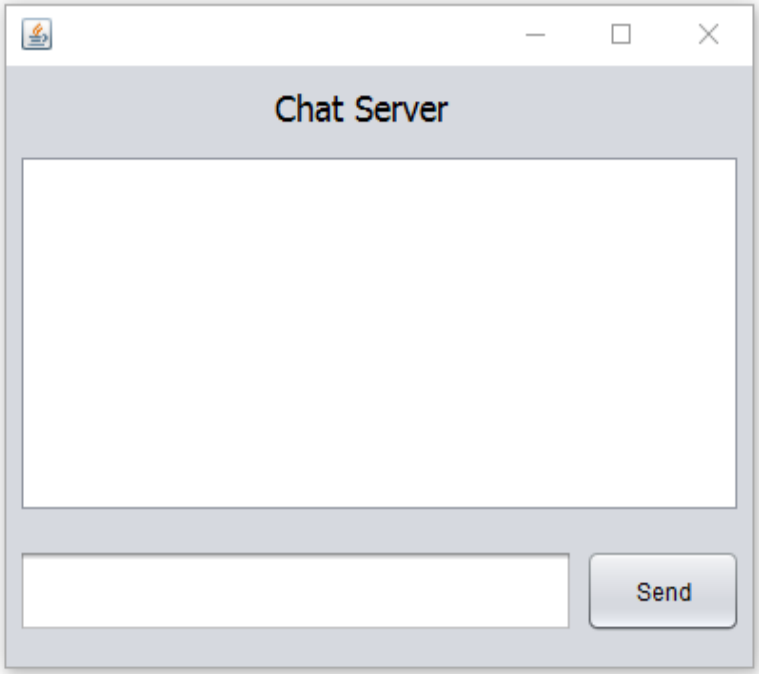
CHAT APPLICATION USING SOCKETS

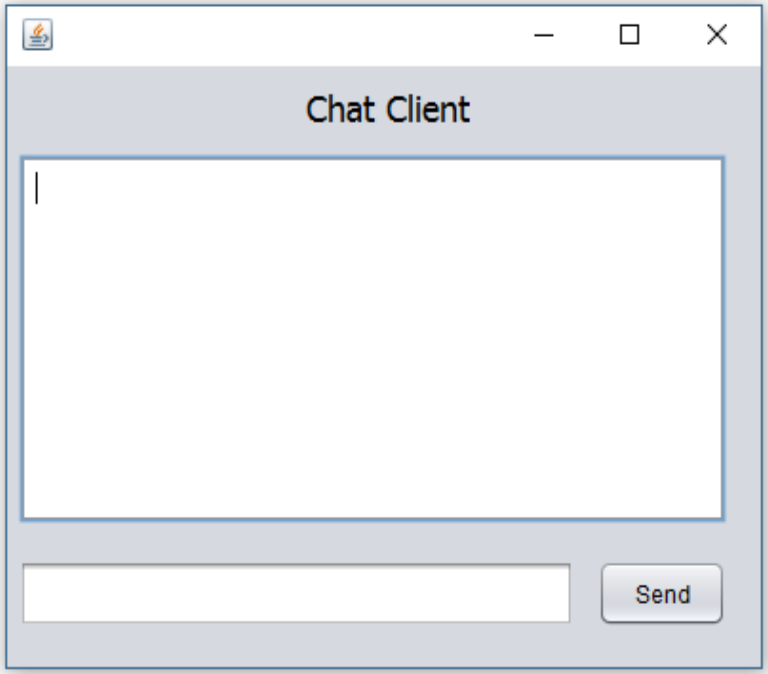
Presented By:

Zeeshan Shabbir Qureshi

15-ARID-1635

STRUCTURE





PACKAGES

- import java.io.DataInputStream;
- import java.io.DataOutputStream;
- import java.net.ServerSocket;
- import java.net.Socket;

WHAT IS A SOCKET?

- The combination of an IP address and a port number. (RFC 793 ,original TCP specification)
- A network socket is one endpoint in a communication flow between two programs running over a network.

SOCKET I/O

- Socket I/O is based on the Java I/O support
  - in the package `java.io`
- InputStream and OutputStream are abstract classes
  - common operations defined for all kinds of InputStreams, OutputStreams…

SERVER SIDE

- Create Server Socket
- Accept Socket Incoming Connection
- Prepare DataInputStream
- Prepare DataOutputStream
- Read Message from Stream
- Display Message on Screen
- Read Message from Screen
- Write Message to Stream

INITIALIZATION

- Server
  - static ServerSocket ss;
  - static Socket s;
  - static DataInputStream din;
  - static DataOutputStream dout;
- Client
  - static Socket s;
  - static DataInputStream din;
  - static DataOutputStream dout;

SERVER CODE

```
String msgin = "";
try
{
    ss = new ServerSocket(1201);
    s = ss.accept();
    din = new DataInputStream(s.getInputStream());
    dout = new DataOutputStream(s.getOutputStream());

    while(!msgin.equals("exit"))
    {
        msgin = din.readUTF();
        msg_area.setText(msg_area.getText().trim()+"\nClient\t"+msgin);
    }

}
catch(Exception e)
{
     System.out.println("Error Occured :"+e.toString());
}
```

SERVER SEND CODE

```
private void btn_sendActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try
    {
        String msgout = "";
        msgout = msg_text.getText().trim();
        dout.writeUTF(msgout);
    }
    catch(Exception e)
    {
        System.err.println("Error : "+e.toString());
    }
```

CLIENT SIDE

- Create and Connect Socket
- Prepare DataInputStream
- Prepare DataOutputStream
- Read Message from Stream
- Display Message on Screen
- Read Message from Screen
- Write Message to Stream

CLIENT CODE

```
try
{
    s = new Socket("127.0.0.1",1201);
    din = new DataInputStream(s.getInputStream());
    dout = new DataOutputStream(s.getOutputStream());
    String msgin = "";
    while(!msgin.equals("exit"))
    {
        msgin = din.readUTF();
        msg_area.setText(msg_area.getText().trim()+"\nServer:\t"+msgin);
    }

}
catch(Exception e)
{
    System.out.println("Error : "+e.toString());
}
```

CLIENT SEND CODE

```
private void btn_sendActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try
    {
        String msgout = "";
        msgout = msg_text.getText().trim();
        dout.writeUTF(msgout);
    }
    catch(Exception e)
    {
        System.err.println("Error : "+e.toString());
    }
```


Thank You!