Name: Akshay Subhash Gadhave

Gr No. : 21920090

Roll No : 221076

SY COMP A3

Assignment 2

Aim: Implement chat server using socket programming

Theory:

Java Socket programming is used for communication between the applications running on different JRE.

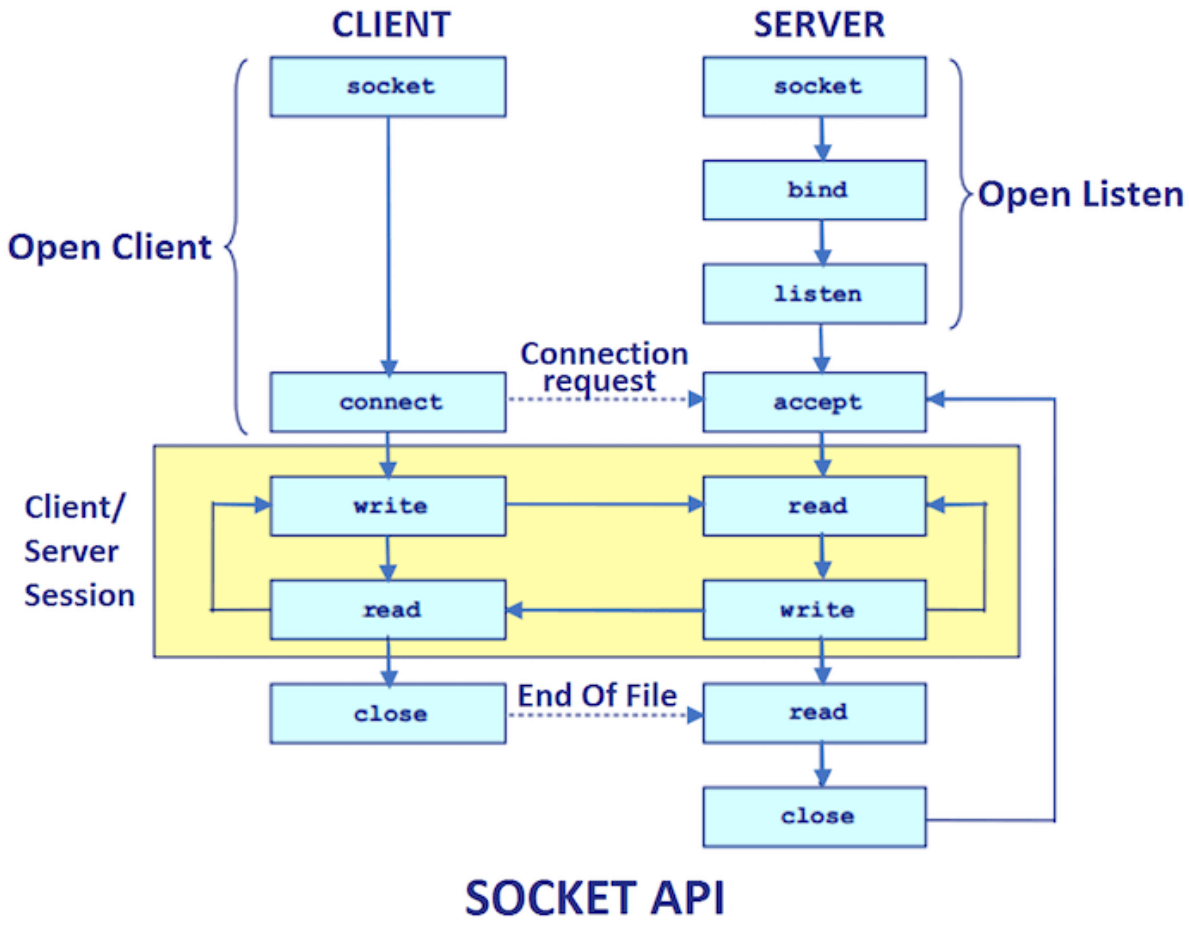
Java Socket programming can be connection-oriented or connection-less.

Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming.

The client in socket programming must know two information:

1. IP Address of Server, and
2. Port number.

Here, we are going to make one-way client and server communication. In this application, client sends a message to the server, server reads the message and prints it. Here, two classes are being used: Socket and ServerSocket. The Socket class is used to communicate client and server. Through this class, we can read and write message. The ServerSocket class is used at server-side. The accept() method of ServerSocket class blocks the console until the client is connected. After the successful connection of client, it returns the instance of Socket at server-side.



Code:

1. Server.java file

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;



public class Server{

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

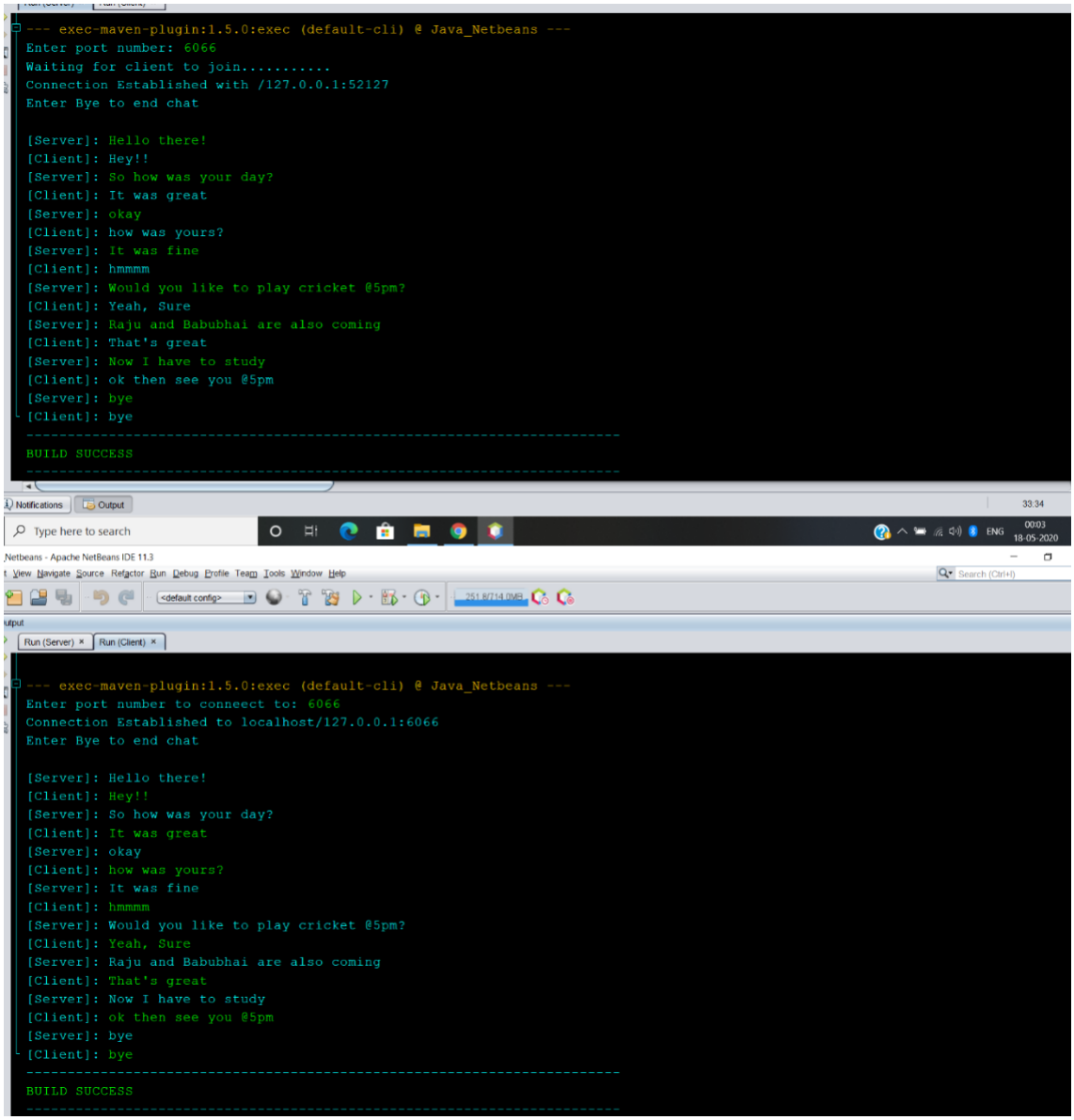
```
        System.out.print("Enter port number: ");
        Scanner s = new Scanner(System.in);
        int port = s.nextInt();
        s.nextLine();
        ServerSocket server = new ServerSocket(port);
        System.out.println("Waiting for client to join.....");
        Socket socket = server.accept();
        System.out.println("Connection Established with
"+socket.getRemoteSocketAddress());
        System.out.println("Enter Bye to end
        chat\n"); String stext,ctext;
        while(true){

            OutputStream out = socket.getOutputStream();
            DataOutputStream dout = new DataOutputStream(out);
            System.out.print("[Server]: "); stext = s.nextLine();

            dout.writeUTF(stext);

            InputStream in = socket.getInputStream();
            DataInputStream din = new DataInputStream(in);
            ctext = din.readUTF();
            System.out.println("[Client]: "+ctext);


    if(stext.equals("Bye")||stext.equals("bye")||ctext.equals("Bye")||ctext.equals("
    b ye"))
                break;
        }
        socket.close();

      }

}
```

2. Client.java file

```
    import java.io.DataInputStream;
    import java.io.DataOutputStream;
    import java.io.IOException;
    import java.io.InputStream;
    import java.io.OutputStream;
    import java.net.Socket;
    import java.util.Scanner;
```

```
public class Client{
   public static void main(String[] args) throws
      IOException{ System.out.print("Enter port number to
      conneect to: "); Scanner s = new Scanner(System.in);
      int port = s.nextInt();
      s.nextLine();
      String servername="localhost";
      Socket socket = new Socket(servername,port);
      System.out.println("Connection Established to
"+socket.getRemoteSocketAddress());
      System.out.println("Enter Bye to end
      chat\n"); String stext,ctext;
      while(true){

         InputStream in = socket.getInputStream();
         DataInputStream din = new DataInputStream(in);
         stext = din.readUTF();
         System.out.println("[Server]: "+stext);

         OutputStream out = socket.getOutputStream();
         DataOutputStream dout = new DataOutputStream(out);
         System.out.print("[Client]: "); ctext = s.nextLine();

         dout.writeUTF(ctext);

if(stext.equals("Bye")||stext.equals("bye")||ctext.equals("Bye")||ctext.equals("
b ye"))
            break;
      }
      socket.close();
   }
}
```

Output:


--- exec-maven-plugin:1.5.0:exec (default-cli) @ Java_Netbeans ---
Enter port number: 6066
Waiting for client to join..........
Connection Established with /127.0.0.1:52127
Enter Bye to end chat
[Server]: Hello there!
[Client]: Hey!!
[Server]: So how was your day?
[Client]: It was great
[Server]: okay
[Client]: how was yours?
[Server]: It was fine
[Client]: hmmmm
[Server]: Would you like to play cricket @5pm?
[Client]: Yeah, Sure
[Server]: Raju and Babubhai are also coming
[Client]: That's great
[Server]: Now I have to study
[Client]: ok then see you @5pm
[Server]: bye
[Client]: bye
BUILD SUCCESS
Notifications
Output
33:34
Type here to search
ENG
00:03
18-05-2020
Netbeans - Apache NetBeans IDE 11.3
View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default config>
Run (Server)
Run (Client)
--- exec-maven-plugin:1.5.0:exec (default-cli) @ Java_Netbeans ---
Enter port number to conneect to: 6066
Connection Established to localhost/127.0.0.1:6066
Enter Bye to end chat
[Server]: Hello there!
[Client]: Hey!!
[Server]: So how was your day?
[Client]: It was great
[Server]: okay
[Client]: how was yours?
[Server]: It was fine
[Client]: hmmmm
[Server]: Would you like to play cricket @5pm?
[Client]: Yeah, Sure
[Server]: Raju and Babubhai are also coming
[Client]: That's great
[Server]: Now I have to study
[Client]: ok then see you @5pm
[Server]: bye
[Client]: bye
BUILD SUCCESS