```java
package thread.chatting;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Collections;
import java.util.HashMap;
import java.util.Iterator;

public class MultiServer {
    HashMap clientMap;
    ServerSocket serverSocket = null;
    Socket socket = null;

    //생성자
    public MultiServer(){
        clientMap = new HashMap();
        Collections.synchronizedMap(clientMap);
    }//생성자----

    public void init(){
        try{
            serverSocket = new ServerSocket(9999);
            System.out.println("서버가 시작되었습니다.");

            while(true){
                socket = serverSocket.accept();

System.out.println(socket.getInetAddress()+":"+socket.getPort());

                Thread msr = new MultiServerRec(socket);
                msr.start();
            }

        }catch(Exception e){
            e.printStackTrace();
        }
    }

    public void sendAllMsg(String msg){

        Iterator it = clientMap.keySet().iterator();

        while(it.hasNext()){
            try{
                DataOutputStream it_out = (DataOutputStream)
clientMap.get(it.next());
```

```java
                it_out.writeUTF(msg);
            }catch(Exception e){
                System.out.println("예외:"+e);
            }
        }
    }

    public static void main(String[] args) {
        MultiServer ms = new MultiServer();
        ms.init();//실행.
    }

    class MultiServerRec extends Thread {

        Socket socket;
        DataInputStream in;
        DataOutputStream out;

        public MultiServerRec(Socket socket){
            this.socket = socket;
            try{
                in = new
DataInputStream(socket.getInputStream());
                out = new
DataOutputStream(socket.getOutputStream());
            }catch(Exception e){
                System.out.println("예외:"+e);
            }
        }

        @Override
        public void run(){

            String name="";
            try{
                name = in.readUTF();

                sendAllMsg(name + "님이 입장하셨습니다.");
                clientMap.put(name, out);
                System.out.println("현재 접속자 수는
"+clientMap.size()+"명 입니다.");

                while(in!=null){
                    sendAllMsg(in.readUTF());
                }
            }catch(Exception e){
                System.out.println(e + "----> ");
            }finally{
                clientMap.remove(name);
                sendAllMsg(name + "님이 퇴장하셨습니다.");
```

```java
                System.out.println("현재 접속자 수는
"+clientMap.size()+"명 입니다.");
            }
        }
    }
}
```

```java
package thread.chatting;

import java.io.IOException;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Scanner;

public class MultiClient {
    public static void main(String[] args) throws
UnknownHostException, IOException {

        System.out.println("이름을 입력해 주세요.");
        Scanner s = new Scanner(System.in);
        String s_name = s.next();

        try{
            String ServerIP = "localhost";
            Socket socket = new Socket(ServerIP, 9999);

            System.out.println("서버와 연결이
되었습니다......");
            Thread sender = new Sender(socket, s_name);

            Thread receiver = new Receiver(socket);
            System.out.println("채팅방에 입장하였습니다.");

            sender.start();
            receiver.start();

        }catch(Exception e){
            System.out.println("예외[MultiClient class]:"+e);
        }
    }
}
```

-

```java
package thread.chatting;
```

```java
import java.io.DataInputStream;
import java.net.Socket;

public class Receiver extends Thread{

    Socket socket;
    DataInputStream in;

    public Receiver(Socket socket){
        this.socket = socket;

        try{
            in = new
DataInputStream(this.socket.getInputStream());
        }catch(Exception e){
            System.out.println("예외:"+e);
        }
    }

    @Override
    public void run(){

        while(in!=null){
            try{
                System.out.println(in.readUTF());

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


```java
package thread.chatting;

import java.io.DataOutputStream;
import java.io.IOException;
import java.net.Socket;
import java.util.Scanner;

public class Sender extends Thread {
    Socket socket;
    DataOutputStream out;
    String name;

    public Sender(Socket socket, String name){
```

```java
        this.socket = socket;
        try{
            out = new
DataOutputStream(this.socket.getOutputStream());
            this.name = name;
        }catch(Exception e){
            System.out.println("예외:"+e);
        }
    }

    @Override
    public void run(){

        Scanner s = new Scanner(System.in);

        try {
            out.writeUTF(name);
        } catch (IOException e) {
            System.out.println("예외:"+e);
        }

        while(out!=null){
            try {

                out.writeUTF("["+name+"] "+s.next());

            } catch (IOException e) {
                System.out.println("예외:"+e);
            }
        }
    }
}
```