conference or public chat

Aim

Write a program to implement public chat

Program description

Algorithm

```
//Server side
create class pserver
    define thread t1, thobj[2] of threadclass, ServerSocket object ser, Socket array soc[20]
    class Threadclass implements Runnable
        define socket s, thread t2
        Constructor of Threadclass assigns a socket to 's' and starts a thread of class Threadclass
         run function of class Threadclass
            create a BufferedReader object br to read data from socket's'
            loop
                read data msg from socket using br
                for(j=1;j<i;j++)
                    create PrintWriter object pw to write data to soc[j]
                    write msg to pw
            end loop
        end of run function of class Threadclass
    end of class Threadclass
    run function of the pserver class
        loop
            assign socket of newly connected socket to soc[i]
            create a Threadclass object thobj[i] using soc[i]
            start thobj[i]
            i=i+1
        end loop
    end of run function of the pserver class
    main function of the pserver class
        create object 'server' using pserver class
        server.ser is a new serversocket object
        convert server object to thread and assign it to t1
        start t1
    end of main function of the pserver class
end of class pserver
```

///Client side

```
Create a class pclient
    Define Socket s, BufferedReader br,br1, PrintWriter pw, Thread t1,t2
    in the constructor of client create objects of class socket,br, br1,pw
    run function
        if thread is t1 read data from the keyboard and write it into the socket
        if thread is t2 read data from the socket and write it into the screen
    end of run
```

main function
create an object of pclient class create a thread from client and assign it to t1
create an object of pclient class create a thread from client and assign it to t2
start t1 and t2
end of main function
end of class pclient

Program

```
//Server side program
import java.io.*;
import java.net.*;

//conference or public chat server program
class pserver implements Runnable
{
 int i=1;
 Thread t1=null;
 ServerSocket ser;
 Socket soc[]=new Socket[20];
 Threadclass thobj[]=new Threadclass[20];
 class Threadclass implements Runnable
{
       int j,k;
       Socket s;
       String msg;
       Thread t2=null;
       BufferedReader br;
       PrintWriter pw;
       public Threadclass( Socket socket)
            {
             s=socket;
             t2=new Thread(this);
             t2.start();
            }
      public void run()
            {
            if(Thread.currentThread()==t2)
                  { try
                       {do
                             {br=new BufferedReader(new InputStreamReader(s.getInputStream()));
                              msg=br.readLine();
                              System.out.println(msg);
                              for(j=1;j<i;j++) if(soc[j]==s)  k=j;
                              msg="client "+k+"says:"+msg;
                              for(j=1;j<i;j++)
                              if(soc[j]!=s)
                                    {  pw=new PrintWriter(soc[j].getOutputStream(),true);
                                       pw.println(msg);
```

```
                                    }
                                }while(!msg.equals("quit"));
                            }    catch(Exception e){}
                        }//if
                    }//run
}
public pserver()
{
        try
         {
                ser=new ServerSocket(8000);
                System.out.println("server listening");
         }
        catch(Exception e){}
}


public void run()
{try
       {
        for(;;)
             {
            soc[i]=ser.accept();
            System.out.println("client"+i+"connected");
            thobj[i]=new Threadclass(soc[i]);
            i++;
             }
       }
      catch(Exception e){}
}

//Main
public static void main(String args[])
    { pserver server=new pserver();
    server.t1=new Thread(server);
    server.t1.start();
    }
}
```

///Client side program

```
import java.io.*;
import java.net.*;
/* conference or public chat in clients through the server*/
class pclient implements Runnable
{
 String msg;
 Socket s;
 BufferedReader br,br1;
```

```
 PrintWriter pw;
 Thread t1=null,t2=null;

public pclient()
{
  try
   {
    s=new Socket("localhost",8000);
    System.out.println("client connected");

    br=new BufferedReader(new InputStreamReader(System.in));
    br1=new BufferedReader(new InputStreamReader(s.getInputStream()));
    pw=new PrintWriter(s.getOutputStream(),true);
   }
  catch(Exception e){}
}

public void run()
      {
       try
            {
            do
                 { if(Thread.currentThread()==t1)
                      {msg=br.readLine();
                       pw.println(msg);
                      }
                  else
                      {msg=br1.readLine();
                       System.out.println(msg);
                      }
                 }while(!msg.equals("quit"));
            }catch(Exception e){}
      }

public static void main(String args[])
     {
       pclient client=new pclient();
       client.t1=new Thread(client);
       client.t2=new Thread(client);
       client.t1.start();
       client.t2.start();
     }
}
```