

Socket klient

Adskilt facade/model og
brugergrænseflade
(Remote facade)

Uddrag af klassen EchoServerFacade

```
private void sendToServer(string text)
{
    writer.WriteLine(text);
    writer.Flush();
}
private string receiveFromServer()
{
    try
    {
        return reader.ReadLine();
    }
    catch
    {
        return null;
    }
}

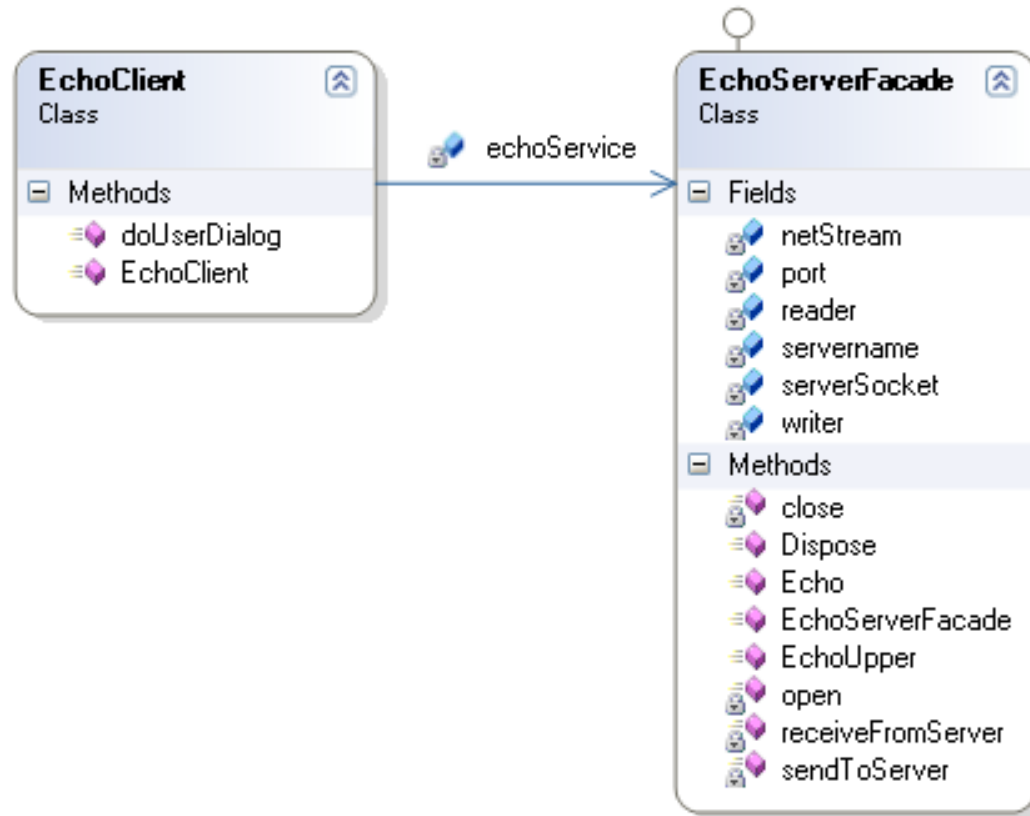
// Remote metoder
public string Echo (string text)
{
    #if TELNET_DIALOG          // Hvis compilerdirektiv er sat medtages koden
        string komandoklarbesked = receiveFromServer(); // vent på klarbesked
        Console.WriteLine("Komando-klarbesked fra server:" + komandoklarbesked);
    #endif

    sendToServer("echo");
    sendToServer(text);
    string response = receiveFromServer();
    return response;
}
```

SocketClientMedRemoteFasade_ConsolApplikation

- I denne udgave vises basalt, hvordan man kan opdele i en fasade, som sørger for kommunikationen med serveren og brugergrænsefladen.
- Denne udgave anvender en consol brugergrænseflade.
- UDGAVEN ANVENDER CONNECTED FORBINDELSE OG SERVER SKAL PASSE TIL

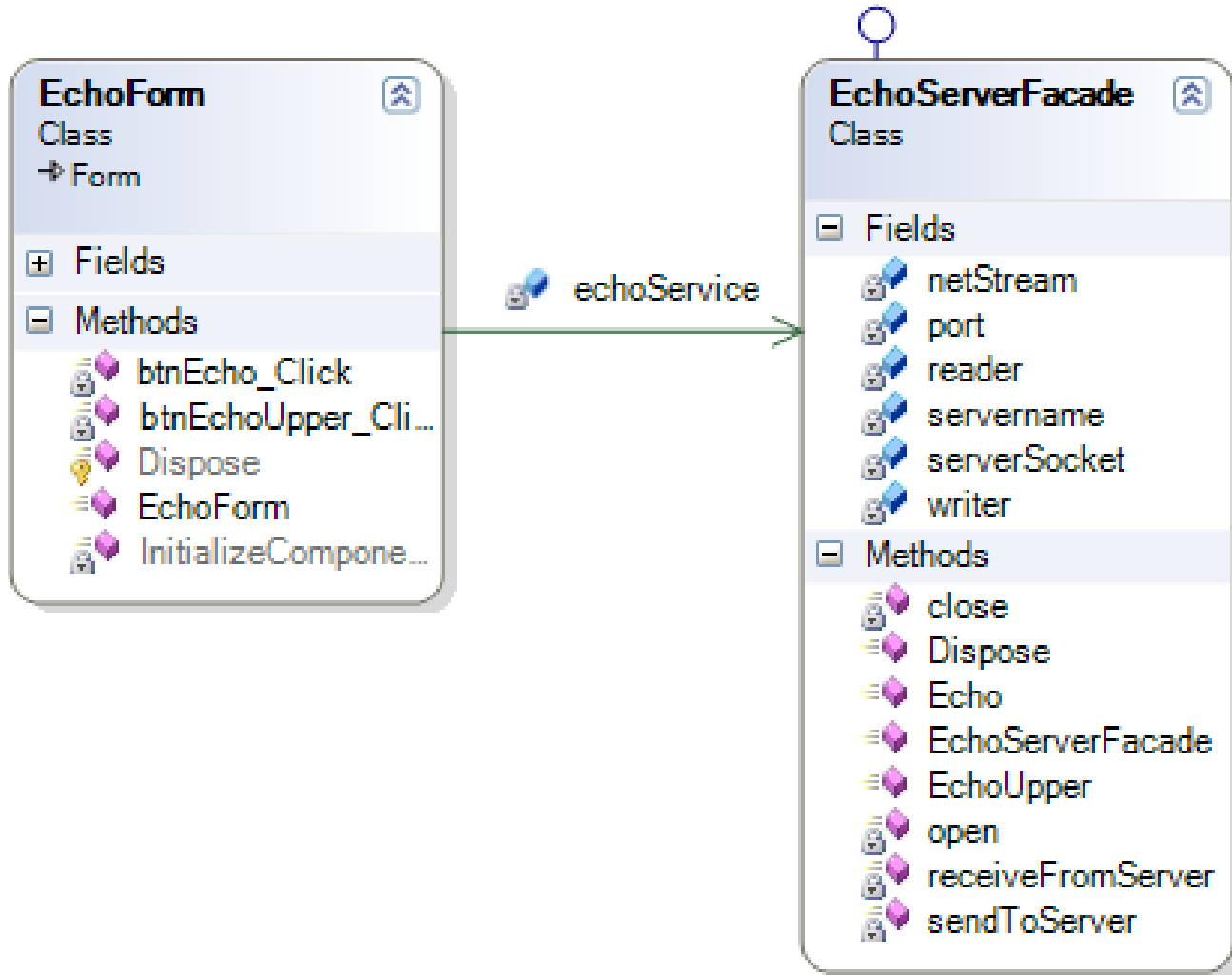
SocketClientMedRemoteFasade_ConsoleApplikation



SocketClientMedRemoteFasade_WindowsApplication

- Samme opdeling som i SocketClientMedRemoteFasade_ConsoleApplication, men med en windows brugergrænseflade.
- **UDGAVEN ANVENDER CONNECTED FORBINDELSE OG SERVER SKAL PASSE TIL**

SocketClientMedRemoteFasade_WindowsApplication



Socket server

Adskilt facade/model og
kommunikation til klient

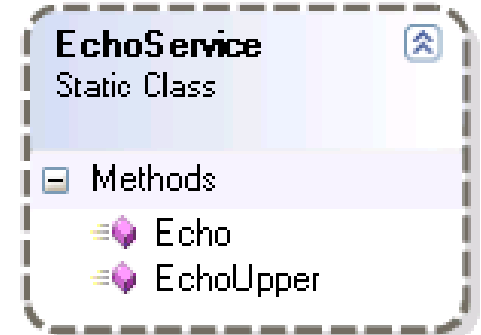
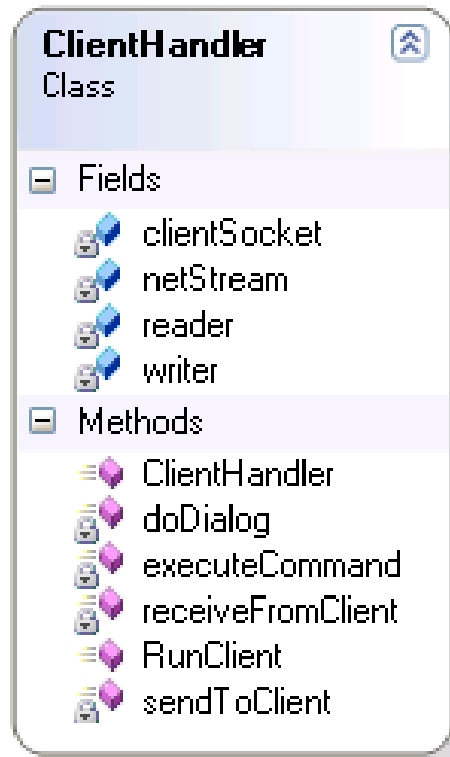
Socket server

Stateless services

SocketServerCmdBasedStatic

- I denne udgave vises basalt, hvordan man kan opdele i kommunikationen med klienten og behandlingen af komandoer.
- Komando behandlingen er udskildt i klassen EchoService med statiske metoder - ingen hukommelse.
- Desuden er selve dialogen opdelt i etableringen og afvilkingen af komandoerne.

SocketServerCmdBasedStatic



Socket server

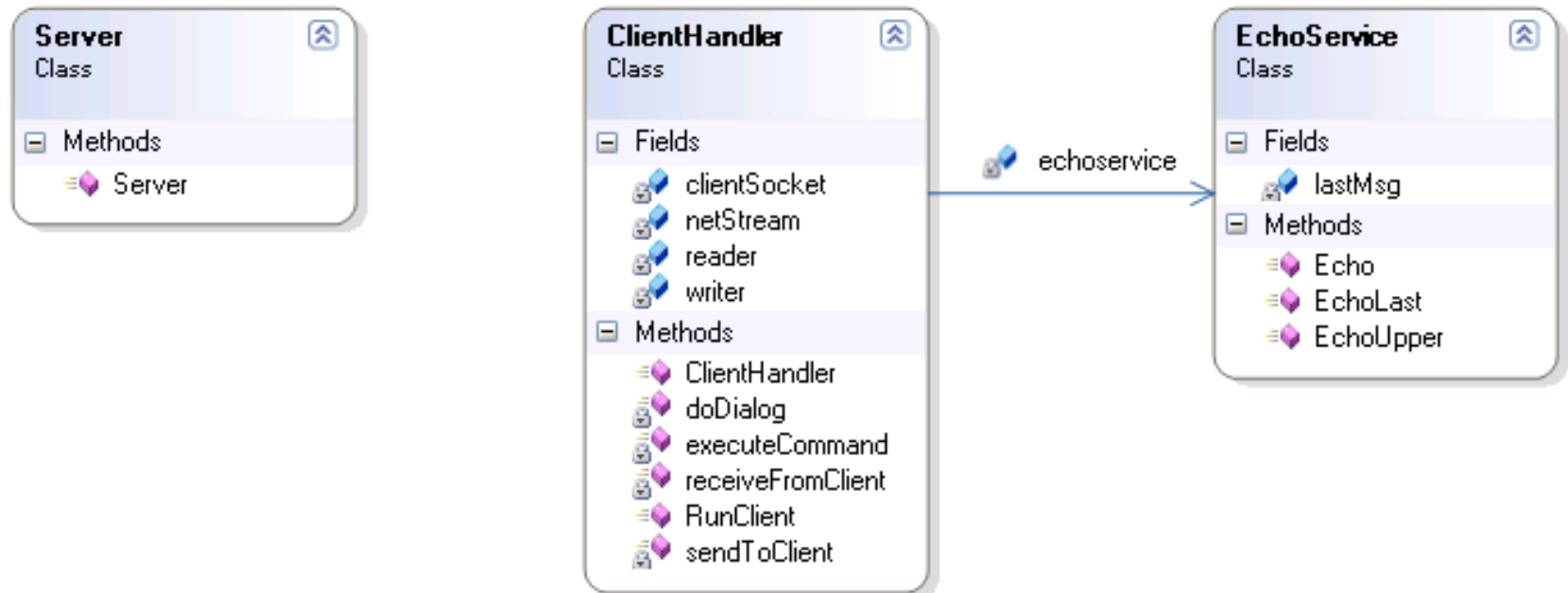
Statefull services

Under åben socket-forbindelse

SocketServerCmdConnected

- I denne udgave er der EchoService klassen ændret, så der istedet bruges et objekt, der desuden nu er
- forsynet med en hukommelse.
- Under dialogen holdes socket-forbindelsen åben og service-objektet lever under hele dialogen,
- hermed bliver det en statefull dialog.

SocketServerCmdConnected



Socket server

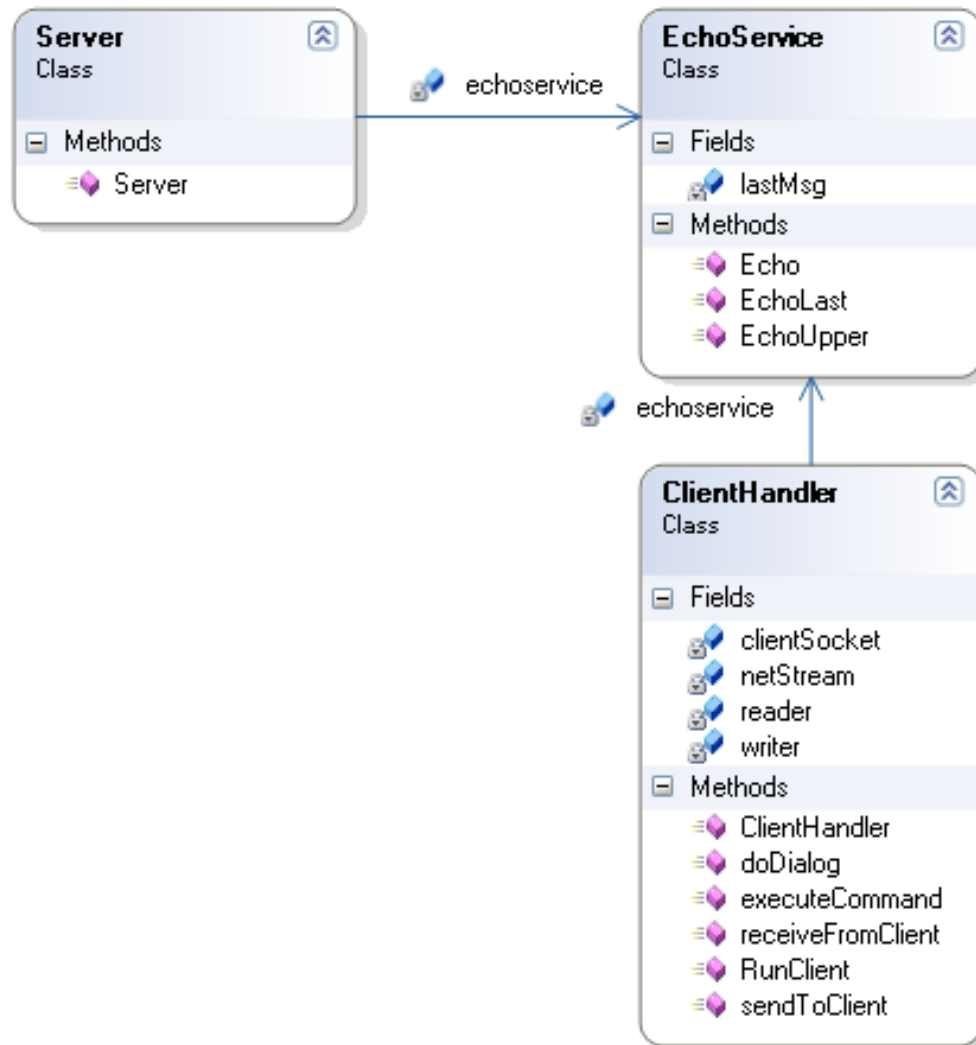
Statefull services

Fælles service for alle klienter

SocketServerCmdDisconnected_V01

- I denne udgave lukkes sokketforbindelsen mellem hver komando.
- For at få service-objektet til at overleve, er det flyttet til server-klassen.
- Der findes eksisterer i udgaven kun et service-objekt, som så nu bliver fælles for alle klienter(forbindelser).
- I udgaven er ikke sikret synkronisering af tilgang til service-objektet.
- Udgaven skal demonstrere, at der kan etableres en form for statefull kommunikation.

SocketServerCmdDisconnected_V01



Socket server

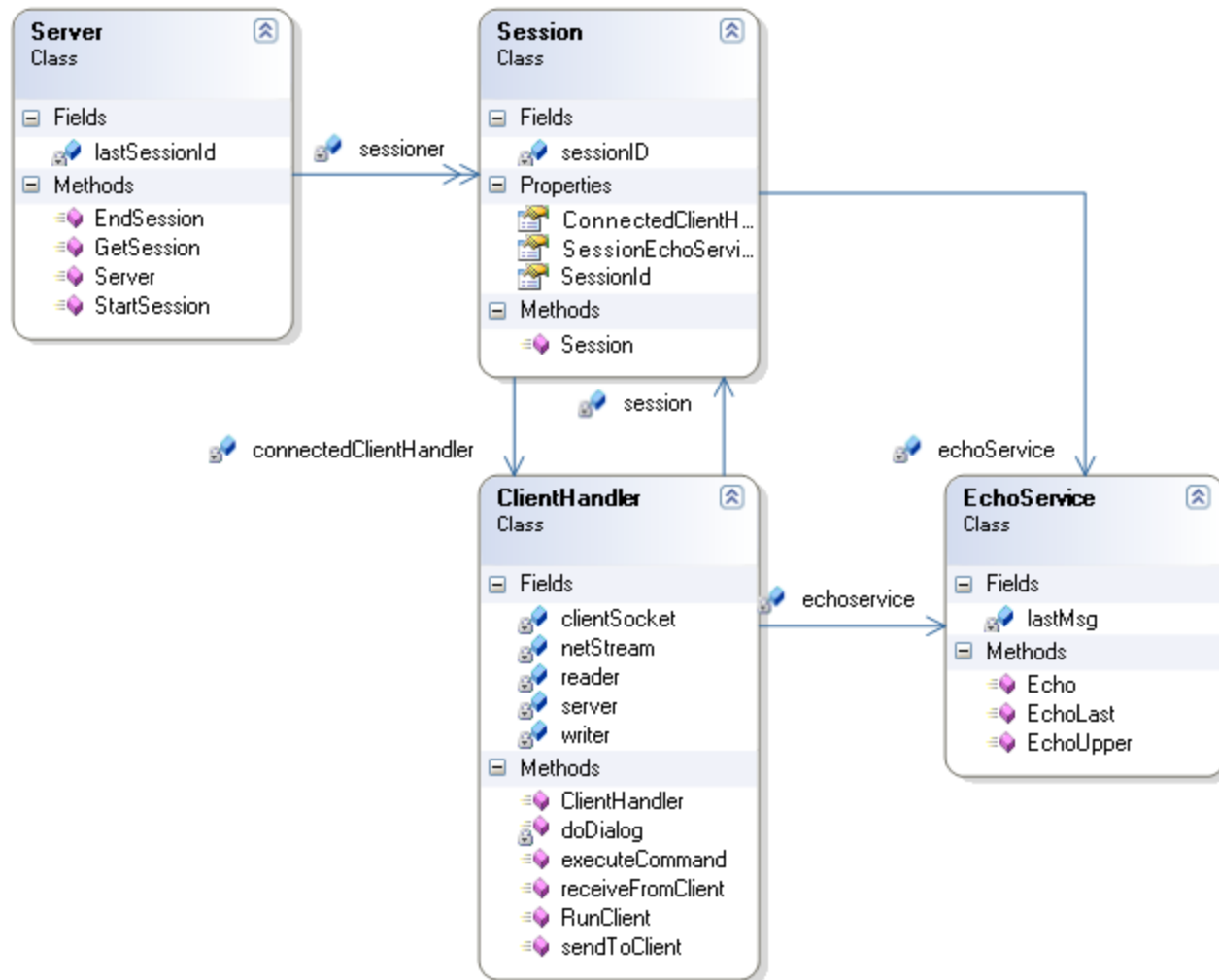
Statefull services for hver klient
(session)

Socket-forbindelse lukkes efter hver
komando

SocketServerCmdDisconnectedSession

- I denne udgave er der tilføjet en ny klasse, som skal indeholde informationer for den enkelte klient.
- Socket-serveren indeholder en samling af disse objekter og giver den enkelte klient adgang til at kunne
- oprette et objekt, tilgå sit objekt og nedlægge det. Hermed kan hver klient nu få sit eget objekt og
- blive statefull, uden det influerer på de øvrige klienter.
- Socket-serveren "holder" på samlingen, der således "lever" i hele socket-serverens levetid.
- Forbindelsen lukkes efter udførelsen af hver komando.
- I udgaven er ikke sikret synkronisering af tilgang til service-objektet.

SocketServerCmdDisconnectedSession



Socket server

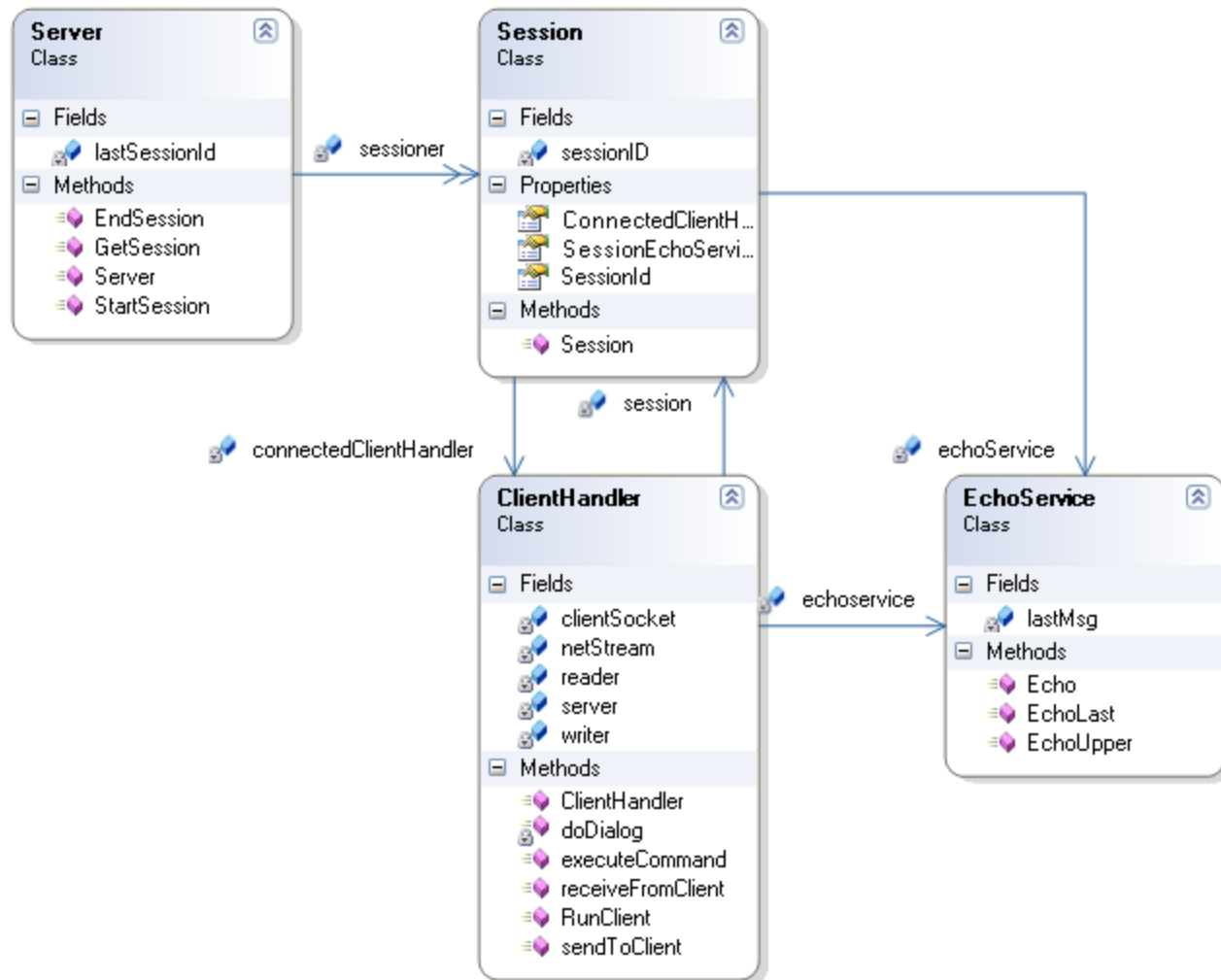
Statefull services for hver klient
(session)

Socket-forbindelse kan holdes åben
eller lukkes efter komando

SocketServerCmdDisconnectedOnOff

- Denne udgave udvider den foregående, ved at forbindelsen kan holdes åben for flere komandoer,
- men også kan afbrydes (ekstra komando "pause").
- Ændringerne er minimale og viser at der nu kan opnås en statefull kommunikation med både en
- opretholdt socket-forbindelse og ved en afbrudt. Løsningen har altså større muligheder end den hvor
- socket-forbindelsen skulle holdes åben.
- I udgaven er ikke sikret synkronisering af tilgang til service-objektet.

SocketServerCmdDisconnectedOnOff



SocketServerCmdDisconnectedOnOff

Session klassen

```
public class Session
{
    private string sessionID;
    private EchoService echoService;
    private ClientHandler connectedClientHandler;

    public Session(string sessionID)
    {
        this.sessionID = sessionID;
    }
    public string SessionId
    {
        get { return this.sessionID; }
    }
    public EchoService SessionEchoService    // ikke trådsikker
    {
        get { return this.echoService; }
        set { this.echoService = value; }
    }
    public ClientHandler ConnectedClientHandler
    {
        get { return this.connectedClientHandler; }
        set { this.connectedClientHandler = value; }
    }
}
```

SocketServerCmdDisconnectedOnOff

data og metoder fra Server klassen – NB ikke trådsikrede i udgaven

```
private Dictionary<String, Session> sessioner = new Dictionary<string, Session>();  
private int lastSessionId = 0;
```

```
public Session StartSession() // ikke synkroniseret  
{  
    ++lastSessionId;  
    Session nySession = new Session(lastSessionId.ToString());  
    sessioner.Add(nySession.SessionId, nySession);  
  
    #if DEBUG // Hvis compilerdirektiv er sat medtages koden  
    Console.WriteLine("SessionId: " + nySession.SessionId + " er nu startet");  
    #endif  
  
    return nySession;  
}  
public void EndSession(Session session) // ikke synkroniseret  
{  
    sessioner.Remove(session.SessionId);  
  
    #if DEBUG // Hvis compilerdirektiv er sat medtages koden  
    Console.WriteLine("SessionId: " + session.SessionId + " er nu afsluttet");  
    #endif  
}  
public Session GetSession(string sessionId) // ikke synkroniseret  
{  
    if (sessioner.ContainsKey(sessionId))  
        return sessioner[sessionId];  
    return null;  
}
```


SocketServerCmdDisconnectedOnOff

doDialog metoden fra ClientHandler klassen

```
private void doDialog()
{
    #if TELNET_DIALOG          // Hvis compilerdirektiv er sat medtages koden
        sendToClient("Server klar"); // opstarts besked til klient
        sendToClient("Indtast SessionId eller 0 for ny"); // opstarts besked til klient
    #endif

    // Skaf session objekt
    string oldSessionId = receiveFromClient();
    session = server.GetSession(oldSessionId); // hent evt. session-objekt
    if (session == null)
    {
        // fantes ikke
        session = server.StartSession(); // start ny session
        session.SessionEchoService = new EchoService();
    }
    echoservice = session.SessionEchoService; // hent stateful modelobjekt / facade

    #if TELNET_DIALOG          // Hvis compilerdirektiv er sat medtages koden
        sendToClient("SessionId:");
    #endif

    sendToClient(session.SessionId); // Send anvendt sessionid til klient

    #if DEBUG          // Hvis compilerdirektiv er sat medtages koden
        Console.WriteLine("SessionId: " + session.SessionId + " er nu aktiv");
    #endif

    //          executeCommand(); den disconnectede tog kun en komando
    while (executeCommand()); // I denne udgave kan der modtages flere
}
```

SocketServerCmdDisconnectedOnOff

uddrag fra metoden executeCommand i ClientHandler klassen

```
public bool executeCommand() //returner false hvis null eller bye
{
    string command;
    command = receiveFromClient();
    if (command == null)
        return false;          // ikke mere input

    switch (command.Trim().ToLower())
    {
        case "echo":
            { // local scope så variable ikke ses uden for
                string inputMsg = receiveFromClient(); // hent supplerende parametre fra klient
                string response = echoservice.Echo(inputMsg); // udfør metode
                sendToClient(response); // send returværdi til klient
            }
            break;
        .....
        .....
        case "bye":
            server.EndSession(session); // Session nedlægges
            return false;
        case "pause":
            // Her afbrydes uden at session nedlægges
            return false;
        default:
            sendToClient("Ukendt komando");
            break;
    }
    return true;
}
```

SocketEchoServiceLibrary

Klasser for service og remote tilgang tilrettet til proxy-mønster

