

Chat server design

ChatServerDesign_00...

Alt designet i én klasse

ChatServerDesign_00_NoTryCatch
+
ChatServerDesign_00



Alt laves i en klasse, idet klienthåndteringen er udskilt i selvstændig metode, så der startes flere klienter ved brug af tråde.

Klassen har en collection af klienternes output streams, der kan til gennemløb ved broadcast.

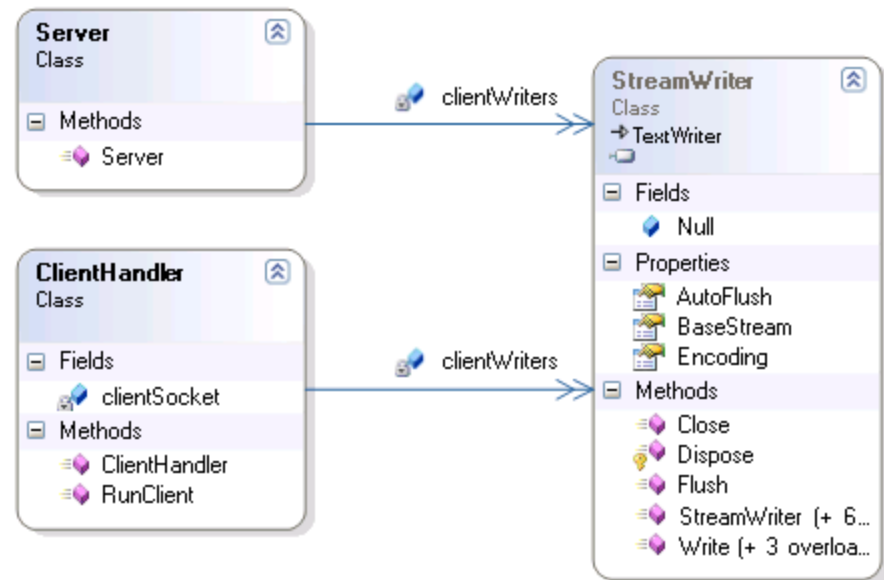
ChatServerDesign_01_1

Klientbehandlingen er udskilt i selvstændig klasse

ChatServerDesign_01_1

Her er opdelt i klient behandlingen udskilt i en sin egen klasse, således der kan oprettes et objekt for hver klientbehandling, med opstart af tilsvarende en tråd pr objekt.

Alle ClientHandler objekterne får adgang til samlingen at output strømme for alle klient handlere.



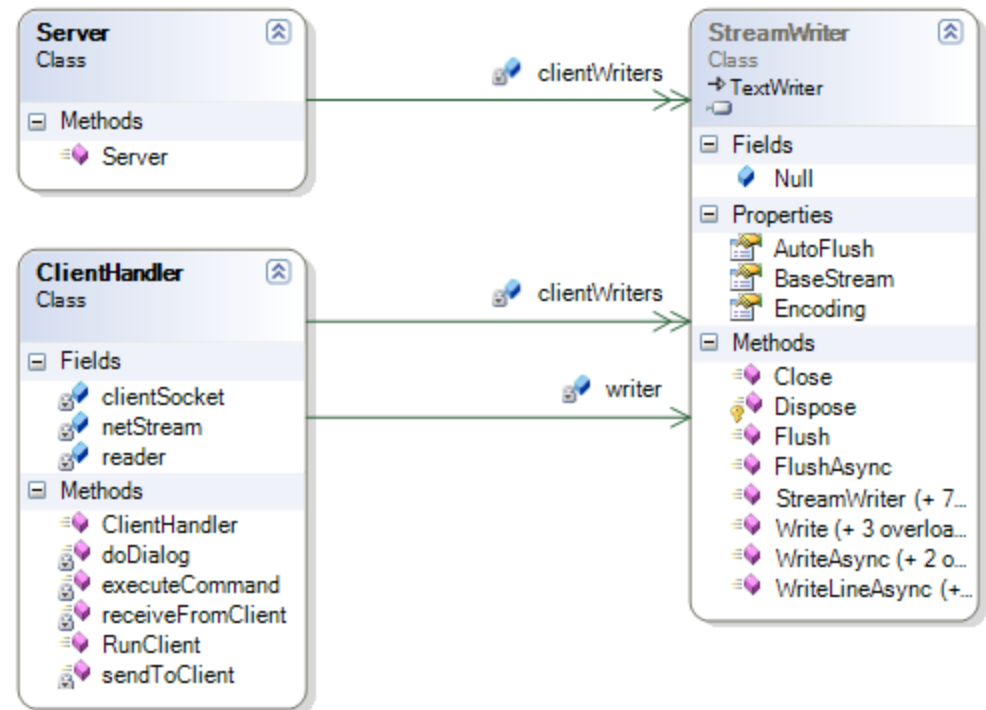
ChatServerDesign_01_1

Klientbehandlingen er udskilt i selvstændig klasse

ChatServerDesign_01_2

Som foregående, men her er opdelt i behandlingen i klient klassen opdelt i flere metoder.

Her er én metode udskilt til at tage sig af behandlingen af kommandoer fra klienten



ChatServerDesign_02_NoLock

Behandlingen af chat-servicen udskilt i selvstændig klasse

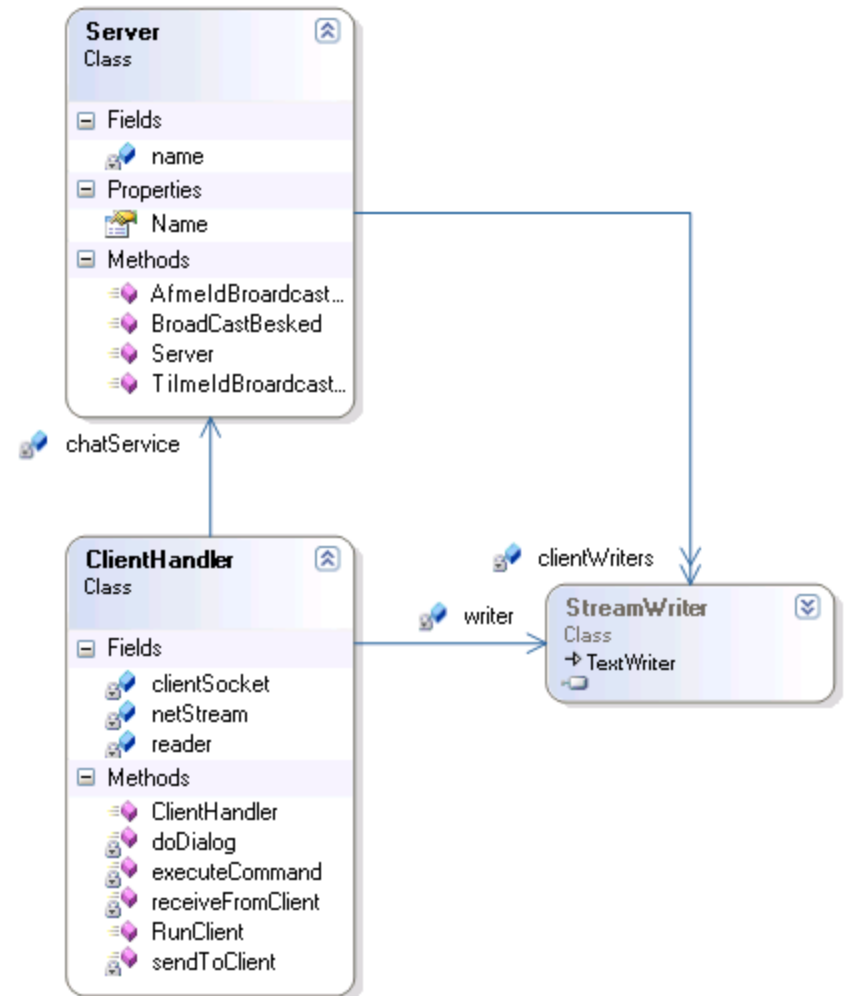
ChatServerDesign_02_NoLock

Her er opdelt i dialog med klienten og bearbejdningen af klientens kommandoer.

Brug af facade-/ model-objekt.

Klienterne sender ikke mere selv broadcast til de øvrige, men overlader opgaven til ChatServicen objektet, der i denne udgave tilgår klienternes StreamWriter objekter direkte.

Ansaret / servicen er nu flyttet til server-objektet, der også var creator.



ChatServerDesign_03_NoLock

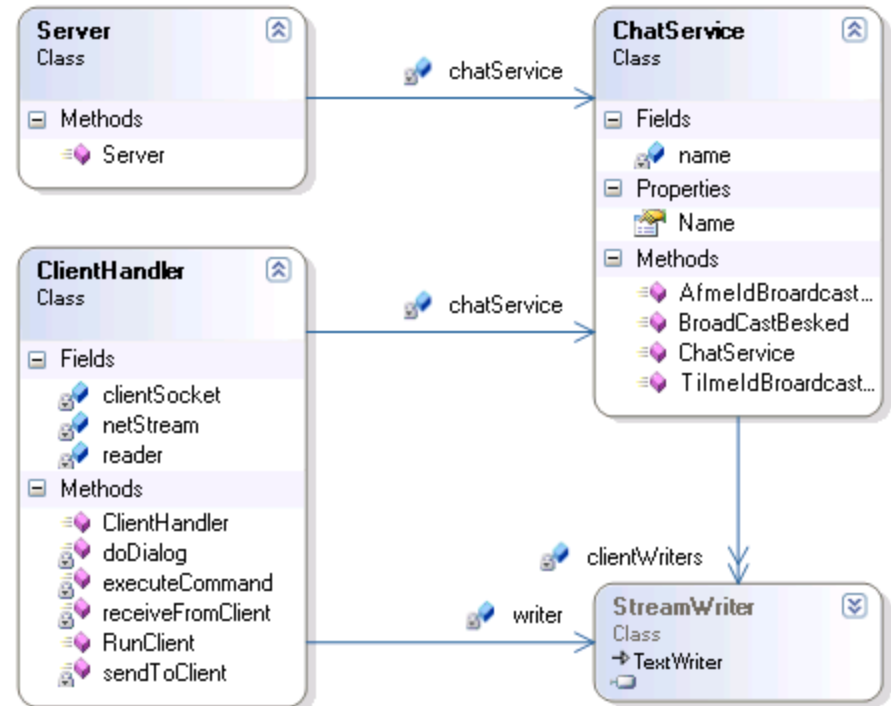
Behandlingen af chat-servicen udskilt i selvstændig klasse

ChatServerDesign_03_NoLock

Her er opdelt i dialog med klienten og bearbejdningen af klientens kommandoer.

Brug af facade-/ model-objekt.

Den fælles ressource der skal tilgås fra flere tråde er nu fjernet skilt fra serverklassen, der havde en aktiv Trådmetode.



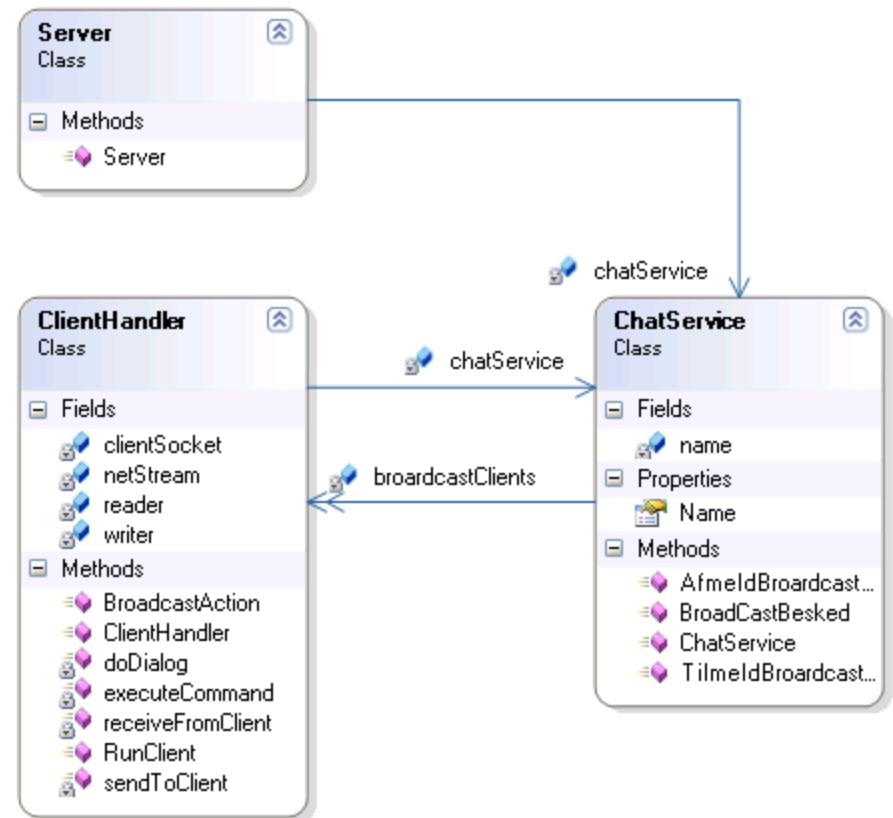
ChatServerDesign_04_NoLock

Behandlingen af chat-servicen udskilt i selvstændig klasse Brug af observer-pattern

ChatServerDesign_04_NoLock

Her er lavet en forbedring i ansvaret, idet CharService'en ikke tilgår klienternes StreamWriter objekter selv, men overlader dette til klienterne selv gennem brug af observer-pattern.

I løsningen her har ChatService-objektet en collection af de ClientHandler objekter, der skal broadcastes til.



ChatServerDesign_05_NoLock

Behandlingen af chat-servicen udskilt i selvstændig klasse
Brug af observer-pattern løst med delegater

ChatServerDesign_05_NoLock

Her er lavet en forbedring i ansvaret, idet CharService'en ikke tilgår klienternes StreamWriter objekter selv, men overlader dette til klienterne selv gennem brug af observer-pattern.

I løsningen her har ChatService-objektet en delegat med de metoder, der skal kaldes ved broadcast.

Ved brug af delegater mindskes koblingen, og her en uheldig fra model til dialog-grænsefladen.

