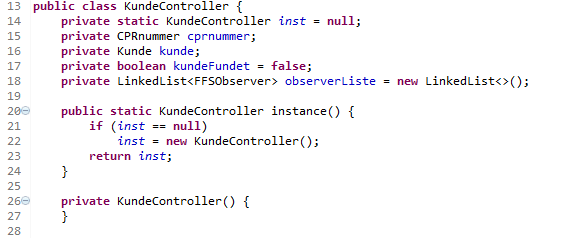
# Anvendelse af singleton-pattern

## Hvorfor singleton?

Vi har til alle vores controllere valgt at anvende singleton-pattern. Idéen opstod tidligt i udviklingsforløbet, da vi indså vi havde brug for kun at lave én instans af hver controller, og så nemt kunne genbruge denne fra flere logik implementationer. Et eksempel på implementationen finder man i KundeController klassen, som set nedenfor. 

Et af kendetegnene er den private contructor, der kun tillader klassen selv at oprette en instans. Constructorkaldet sker i instance() metoden, der, hvis ikke der allerede findes en instans, laver constructorkaldet, og returnerer dette. Hvis objektet allerede er blevet instantieret, returneres instansen.

Et eksempel på instantieringen finder vi i LånetilbudPanel, som vist nedenfor.



I KundePanel linje 43 findes vi præcis den samme linje, der også her forsøger at instantiere. Uanset hvad der får held med at oprette instansen først, har vi nu to klasser der begge har tilgang til den samme controller, og dermed den samme datakerne. Dette anvendes blandt andet i LånetilbudController, hvis vi får adgang til den allerede oprettede KundeController, og dermed let kan få returneret de instansvariabler der ligger gemt. I nedenstående tilfælde er det en instans af Kunde vi ønsker (se figur xx, kodelinje 68).

