Øving 11

Oppgave 1)

Jeg mener at oppgaven hovedsakelig vil trenge tre klasser: en som representerer selve eiendommen, en som representerer et register som lagrer objekter av typen eiendom, og en klient som behandler kommunikasjonen med brukeren og tar i bruk klassene.

Eiendomsregisteret bør kun behandle lagringen og behandlingen av eiendommene i sitt eget register. Eiendomsklassen bør kunne opprette eiendommer og redigere eksisterende eiendommer.

Klienten bør ikke ha direkte tilgang til objektvariabler gjennom selve objektene, men kan vise fram objektenes attributter gjennom aksessor-metoder, og endre de gjennom mutator-metoder.

Oppgave 2a)

Jeg valgte å gi klassen to konstruktører, én som krever alle attributtene og én som krever alle utenom bruksnavn. Dette fordi oppgaven spesifiserte at bruksnavn ikke er obligatorisk for en eiendom.

Jeg valgte å gi klassen Property aksessor-metoder for alle attributter, og mutator-metoder for attributtene kommunenavn, bruksnavn, eiendomsareal og navnet på eier. Kommunenavnet kan endre seg gjennom sammenslåing eller splittelse. Bruksnavnet kan endres dersom eieren ønsker det. Eiendomsarealet kan øke eller minke, etter kommunale bestemmelser. Navnet på eieren vil endre seg ved for eksempel død, salg eller arv.

Oppgave 2b)

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Oppgave 3a)

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Oppgave 3b)

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Oppgave 3c)

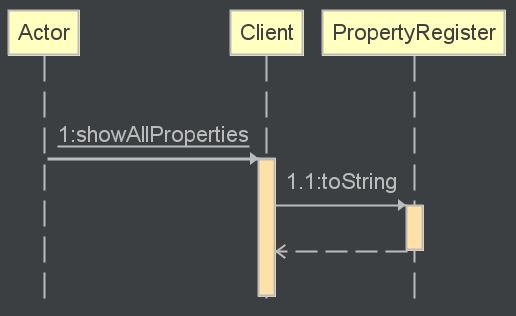
Et bilde som inneholder tekst

Automatisk generert beskrivelse

Oppgave 4a)

Se Client.java.

Oppgave 4b)



Oppgave 5

Jeg mener at mine klasser er relativt solide og godt designet. Komposisjon er brukt der det er hensiktsmessig, for eksempel mellom Property- og PropertyRegister-klassene. PropertyRegister sine metoder avhenger av at det finnes et objekt av klassen Property, mens Property-klassen ikke har noen spesiell tilstedeværelse utenom registerer i PropertyRegister-klassen.

Jeg mener at klassene har god samstemthet, ved at de har klare hensikter for hva de skal inneholde og kunne utføre av operasjoner. Register-klassen skal behandle registeret (ved å legge til/fjerne eiendommer fra selve registeret), mens Property-klassen har i oppgave å behandle selve objektene sine. PropertyRegister-klassen har ingen metoder for å manipulere selve eiendommene og deres attributter, disse er forbeholdt Property-klassen. Ingen andre klasser enn Client-klassen kan kommunisere direkte med brukeren av programmet, ettersom at ingen av de andre klassene snakker direkte med brukeren (for eksempel gjennom å be om input/printe ut til konsollen).

Jeg mener at jeg har god og lav kobling mellom klassene, ettersom at selve registeret ikke blir påvirket av for eksempel at et objekt sine verdier endres. I tillegg vil ikke selve objektet bli påvirket av registeret, med mindre koden gjøres om slik at dette er tilfelle.