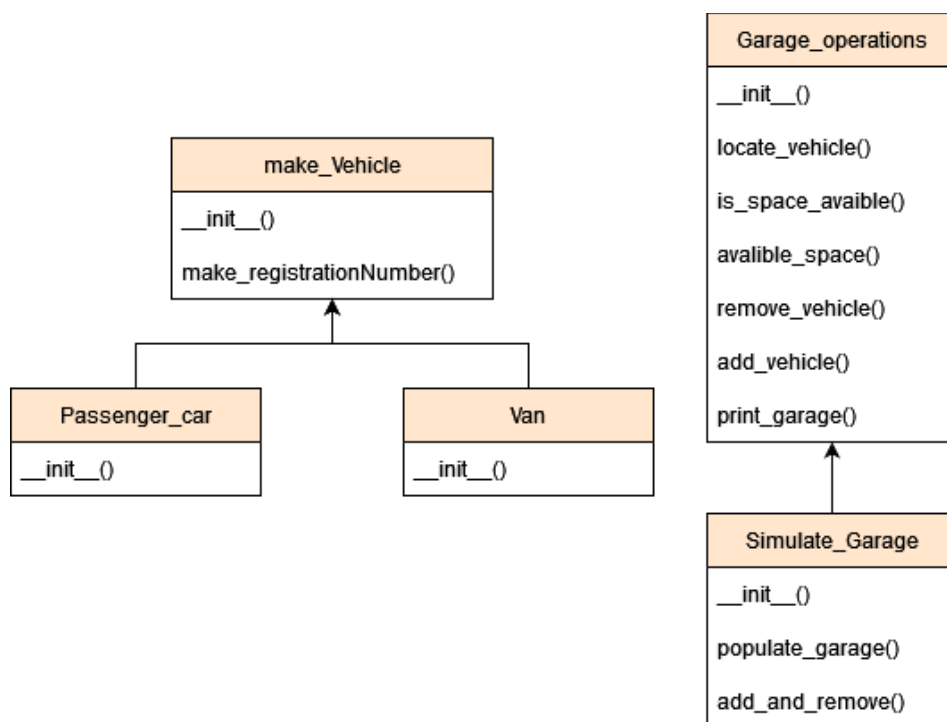


# INF-1400 Objekt Orientert Programmering

## Eksamen

### Oppgave 1

Et klassediagram som forteller relasjonene mellom klassene, og hvilke **funksjoner** klassene inneholder.



### Oppgave 2

For å endre den eksisterende koden, flyttet jeg hovedsakelig hele kode inni klasser, og skiller dem mellom 2 hovedklasser. Der den ene klassen lager bilene, og den andre klassene holder alle operasjonene man kan gjøre på en «garasje»/liste hvor bilene befinner seg i.

Tanken er at du har én hovedklasse som lager bilene. Der denne hovedklassen har 2 barn («Passenger\_car» og «Van») som arver fra foreldreklassen, og disse barnene inneholder forskjellige attributter. Den andre hovedklassen inneholder alle operasjonene man kan gjøre på garasjen bilene ligger i. Den har også ett barn («Simulate\_Garage») som arver fra denne foreldreklassen, hvor dette barnet simulerer hele programmet.

Polymorfien som blir brukt i programmet er hovedsakelig i hovedklassen

«Garage\_operation», men den har en del forventinger. For det første forventer klassen og få inn en liste av en tilfeldig størrelse, der denne listen inneholder objekter av typen «make\_Vehicle». Hvis disse antagelsene blir møtt, gir funksjonene inni hovedklassen forskjellige resultater. Ett eksempel kan være funksjonen som skriver ut innholdet i garasjen, den skriver ut både registreringsnummeret og typen til bilen, uavhengig av verdiene de holder.

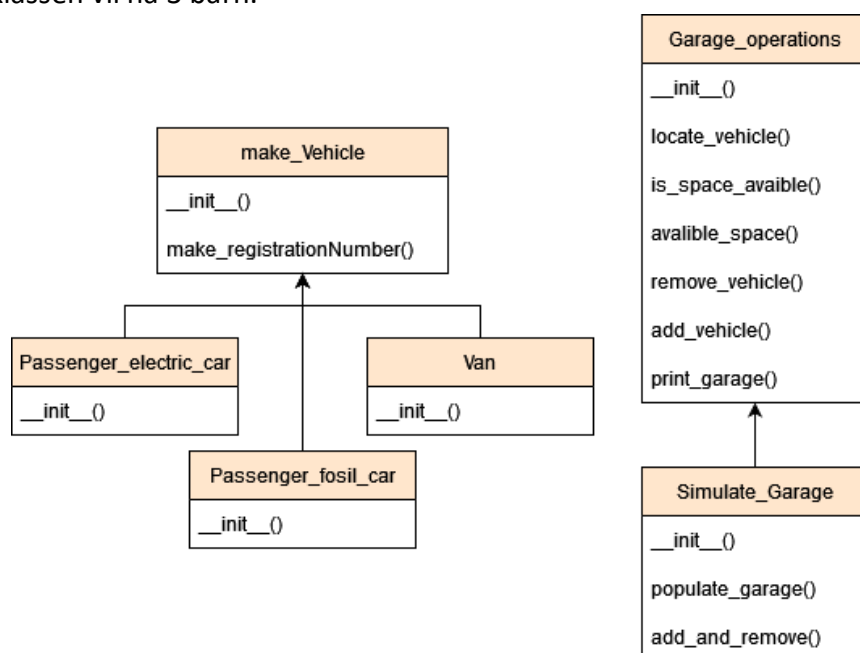
### Oppgave 3

Se vedlegg (code\_task3.py).

### Oppgave 4

Det skal skilles mellom elbiler og fossilbiler, og man skal kunne holde styr på kapasiteten til disse bilene.

- 1) Jeg vil plassere disse nye klassene i det eksisterende klassehierarkiet, ved å fjerne den eksisterende «Passenger\_car» klassen, og legge til to nye klasser som heter «Passenger\_fosil\_car» og Passenger\_electric\_car». Det vil da si at «make\_vehicle» hovedklassen vil ha 3 barn.



- 2) De nye klassene trenger bare ett nytt attributt, sammenliknet med den gamle klassen, som holder på kapasiteten til bilene.
- 3) Se vedlegg (code\_task4.py).