

## Rapport

### Innkapsling

Jeg har valgt å bruke private instansvariabler, gettere og konstruktører for å samhandle med instansvariablene. Jeg har valgt å ikke bruke settere, fordi jeg ikke vil at det skal være mulig å endre verdiene til objektene mine, og det blir redundant når jeg allerede bruker konstruktører.

### Unntak

Jeg har i all hovedsak håndtert unntak ved å kaste de lengre opp i kjeden, og bruke try catch i Main metoden min når jeg kaller på readFile, writeToDB, og program metodene mine. Dette gjør at jeg slipper å bruke mange try catcher i metodene som blir kalt på fra hoved metodene mine. Jeg har også brukt rollback i writeToDB metoden for å sørge for at all informasjonen skal sendes riktig til SQL databasen.

### Arv

Jeg har implementert arv i oppgaven ved å lage en vehicle superklasse som inneholder alle felles instansvariablene til electricCar, fossilCar og motorcycle. Jeg har bare ett eksempel på polymorfi i klassene, og det er toString metodene til de ulike kjøretøyene.

### Erfaringer

Jeg hadde ingen spesielle utfordringer underveis i oppgaven, og syntes at den var ganske oversiktlig å lage. Jeg hadde jobbet en del med tidligere eksamener, og kunne derfor gjøre mye likt i forhold til hvordan jeg hadde gjort det der.