

# Obligatorisk oppgave 2: Bilhierarki

INF1010

Frist: mandag 20. februar 2017 kl. 12:00

Versjon 1.0.1 (1f2668e ). Endringer siden versjon 1.0

## Innhold

<b>1 Innledning</b>	<b>1</b>
<b>2 Tegning av klassehierarkiet</b>	<b>2</b>
<b>3 Implementasjon</b>	<b>2</b>
<b>4 Hovedprogram</b>	<b>3</b>
4.1 Innlesning fra fil . . . . .	3
4.2 Informasjon om bilene . . . . .	3
4.3 Datastrukturtegning . . . . .	4
<b>5 Oppsummering</b>	<b>4</b>

## 1 Innledning

I denne oppgaven skal du lage et klassehierarki av biler. Noen biler er drevet av fossilt brennstoff, andre er drevet av elektrisitet fra et batteri. Når vi skal modellere dette i et program, trenger vi å se på hva bilene har til felles og organisere dem i et klassehierarki slik at vi kan samle det som er felles i superklasser, så kan vi lage mer spesialiserte subclasser for de spesifikke biltypene.

Du skal skrive et program for en venn som har et lite bilutsalg. Han har beskrevet behovet sitt slik:

- Alle biler har et skilt (også kalt *kjennemerke* av Statens vegvesen) hvor bilens registreringsnummer står. Dette registreringsnummeret kan bestå av både bokstaver og tall.

- Alle biler er enten elbiler eller fossilbiler.
  - Elbiler har et batteri med en kapasitet målt i kWh.
  - Fossilbiler har et gitt CO2-utslipp målt i g/km. Alle fossilbiler er enten personbiler eller lastebiler.
    - \* Personbiler har et antall seter som er godkjent for passasjerer.
    - \* Lastebiler har en nyttevekt, målt i kg, som bestemmer hvor tungt den kan lastes.

Med *personbil* mener vi altså personbil drevet av fossilt drivstoff.

#### Til ettertanke.

Heldigvis for deg selger ikke din venn hybridbiler, men hvor ville du plassert en hybridbil i dette hierarkiet?

Du skal altså i denne oppgaven **ikke** ta hensyn til at din venn kanskje senere skal selge hybridbiler. Vi kommer til å ta opp dette med hybridbiler i denne sammenhengen senere i INF1010.

## 2 Tegning av klassehierarkiet

Tegn opp klassehierarkiet som er beskrevet i innledningen. Det er ikke nødvendig å ta med noe annet enn navnene på klassene.

## 3 Implementasjon

Skriv programmet til de fem klassene. Du står fritt til å lage hvilke variabler du måtte ønske for å ta vare for de egenskapene som beskrevet i oppgaveteksten.

I tillegg skal alle biler kunne skrive ut informasjon om seg selv.

**Tips.** Hvis du implementerer denne metoden i alle klassene, kan du kalle superklassens metode for å slippe å gjenta mye kode.

## 4 Hovedprogram

### 4.1 Innlesning fra fil

Skriv et program som leser en fil med data om biler og oppretter et objekt for hver bil (hver linje) i filen. Navnet på filen oppgis som et kommandolinjeargument (i `String[] args`). Referansene til disse objektene skal legges inn i en `ArrayList<Bil>` eller et array av typen `Bil`. Hvis du bruker et array, kan du anta at det ikke er mer enn 100 biler i filen. Hver linje i filen starter med et ord som identifiserer klassen objektet tilhører. Deretter kommer verdiene på variablene i objektet. Du kan anta at det ikke er feil i filen. Filen vil inneholde linjer som er på formen

```
EL <reg.nr.> <batterikapasitet>
PERSONBIL <reg.nr.> <utslipp> <seter>
LASTEBIL <reg.nr.> <utslipp> <nyttevekt>
```

Her er en eksempelfil, `1.in`. Du finner denne filen og flere andre eksempelfiler på <http://folk.uio.no/inf1010/v17/oblig/2/samples/>.

```
EL EK43536 31.5
LASTEBIL SR87875 452.5 2550.4
PERSONBIL AR34345 119.5 8
EL EL14545 60
PERSONBIL DK65437 135.7 4
```

### 4.2 Informasjon om bilene

Når hele filen er lest inn, skal du gå gjennom alle bilene og skrive ut data om et filtert utvalg av alle bilene (de andre bilene skal du ikke skrive ut noen informasjon om). Hvilket filter som skal brukes, bestemmes av et kommandolinjeargument. Programmet ditt skal støtte følgende filtre:

1. Alle biler (intet kommandolinjeargument)
2. Bare elbiler (EL som kommandolinjeargument)
3. Bare fossilbiler (FOSSIL som kommandolinjeargument)

Programmet skal altså kunne brukes som følger:

```
$ java Oblig2 <inputfil> [filter]
```

Her er noen kjøreeksampler:

```
$ java Oblig2 2.in
Type motorvogn:      Personbil
Reg.nr: BS95857
CO2-utslipp: 72.2000
Seter: 5
Type motorvogn:      Lastebil
```

```
Reg.nr: YG48124
CO2-utslipp: 603.500
Nyttevekt: 5003.20
```

```
Type motorvogn:      Elbil
Reg.nr: EK10100
Batterikapasitet (kWh): 90.0000
```

```
$ java Oblig2 2.in EL
Type motorvogn:      Elbil
Reg.nr: EK10100
Batterikapasitet (kWh): 90.0000
```

```
$ java Oblig2 2.in FOSSIL
Type motorvogn:      Personbil
Reg.nr: BS95857
CO2-utslipp: 72.2000
Seter: 5
```

```
Type motorvogn:      Lastebil
Reg.nr: YG48124
CO2-utslipp: 603.500
Nyttevekt: 5003.20
```

#### Merk.

Dette er bare et eksempel på hvordan utskriften kan se ut. Utskriften fra programmet ditt kan godt være formatert annerledes, så lenge den samme informasjonen kommer med.

### 4.3 Datastrukturtegning

Tegn datastrukturen slik den ser ut rett før programmet terminerer når du har lest inn filen over (1.in).

Du trenger ikke å tegne inn metoder i objektene.

## 5 Oppsummering

Du skal:

- Tegne klassehierarkiet
- Implementere klassene
- Lage et hovedprogram som oppretter instanser av biler basert på input fra

en fil og etterpå skriver ut info om alle de innleste bilene, eller bare fossilbilene eller bare elbilene.

- Tegne datastrukturen som oppstår under kjøring av hovedprogrammet

Tegningen av klassehierarkiet og datastrukturen skal leveres sammen med koden på Devilry.

**Lykke til!**

*Stein Gjessing, Stein Michael og Kristian*