Biludlejningsopgave

Beregning af registreringsafgift:

1. For en Benzinbil er afgiften afhængig af kilometer pr liter. Hvis den kører mellem 20 km/l og 50 km/l er prisen 330 kr, mellem 15 km/l og 20 km/l er prisen 1050 kr, mellem 10 km/l og 15 km/l er prisen 2340 kr, mellem 5 km/l og 10 km/l er prisen 5500kr, og under 5 km/l er prisen 10470 kr.
2. For en Elbil gælder de samme regler som for en benzinbil, blot skal man først omregne watt-timer pr kilometer til km/l. Det gøres ved at dividere Wh/km med 91,25 og dernæst dividere 100 med dette tal. Se evt. formlen her: https://fdm.dk/alt-om-biler/dine-rettigheder/boder-afgifter/ejerafgift-elbil.
3. For en Dieselbil er der samme afgift som for benzinbilen, plus en udligningsafgift, som også er afhængig af km/l. Hvis bilen kører mellem 20 km/l og 50 km/l er udligningsafgiften 130 kr, kører den mellem 15 km/l og 20 km/l er den 1390 kr, kører den mellem 10 km/l og 15 km/l er den 1850 kr, kører den mellem 5 km/l og 10 km/l er den 2770 kr, og kører den under 5 km/l er den 15260 kr. Der er desuden en partikeludledningsafgift på 1000 kr hvis bilen ikke har et partikelfilter monteret.

I skal lave et system til en biludlejningsvirksomhed. Systemet skal automatisk kunne beregne den samlede registreringsafgift for hele flåden af biler, som udlejningsfirmaet besidder.

Bilerne

Der skal være følgende interfaces og klasser til at repræsentere biler:

1. Interface Car. Skal have følgende metoder:
   a. String getRegistrationNumber(); //The number on the number plate
   b. String getMake(); // The make of the car e.g. Audi
   c. String getModel(); // The model of the car e.g. A6
   d. Int getNumberOfDoors(); // The number of doors
   e. Int getRegistrationFee(); // Calculates the registration fee for the car
2. Abstrakt klasse ACar. Skal implementere Car interfacet og alle metoderne i dette, undtagen getRegistrationFee(), som skal implementers i de konkrete klasser (i punkt 4,5 og 6). Lav klasse attributter til at holde styr på registreringsnummer (nummerplade), mærke, model og antal døre. (Husk at bruge engelske navne til alt). Overvej hvilke attributter det giver mening at lave final.
3. Abstrakt Klasse AFuelCar. Denne klasse skal nedarve fra Acar. Den skal tilføje to metoder:
   a. abstract String getFuelType(); // should return “Gasoline” or “Diesel”
   b. int kmPrLitre(); // should return how many kilometres the car can drive on 1 litre of fuel
   Lav en attribut til at gemme kilometer per liter og initialisér den i konstruktøren.
4. Klasse GasolineCar. Denne klasse nedarver fra AFuelCar og skal implementere de to abstrakte metoder getFuelType() og getRegistrationFee(). Registreringsafgiften skal beregnes ud fra beskrivelsen i toppen af opgaven.
5. Klasse DieselCar. Denne klasse nedarver også fra AFuelCar og skal også implementere de to abstrakte metoder getFuelType() og getRegistrationFee(). Herudover skal der være en

metode, hasParticleFilter(), der fortæller om bilen har et partikelfilter monteret. Registreringsafgiften skal beregnes ud fra beskrivelsen i toppen af opgaven.

6. Klasse ElectricCar. Denne klasse nedarver fra ACar. Den skal tilføje metoderne:
   a. getBatteryCapacityKWh(); // returns the battery capacity in kilowatt hours
   b. getMaxRangeKm(); // returns the maximum range in kilometres.
   c. getWhPrKm(); // returns the power consumption in watt hours per driven kilometre.

   Lav attributer til battery capacity og max range, som initialiseres i konstruktøren. Beregn watt-timer per kilometer ud fra disse to attributter.

De tre konkrete (ikke-abstrakte) klasser, GasolineCar, DieselCar og ElectricCar skal overskrive toString() metoden, så den returnerer en String, der repræsenterer bilen på en overskuelig måde. Du kan evt. overskrive toString() i Acar klassen, så den returnerer en String med de attributter, der er fælles for alle biler og overskrive toString() i FuelCar klassen så den bruger super klassens toString() og tilføjer km/l. Herefter kan de tre konkrete klasser kalde deres super-klassers toString() metode og blot tilføje de ekstra attributter, der er i den konkrete biltype.

Flåden af biler

Der skal laves en klasse FleetOfCars, der indeholder en ArrayList<Car>, der kan indeholde alle bilerne, som udlejningsfirmaet råder over. Klassen skal indeholde en metode til at tilføje en bil til flåden. Den skal overskrive toString(), så den returnerer en String, der lister alle bilerne i flåden. Sidst men ikke mindst skal den have en metode, getTotalRegistrationFeeForFleet(), der beregner den samlede registreringsafgift for hele bilflåden.

Skriv en Main-klasse med en main-metode der instantierer et FleetOfCars-objekt og et antal bil-objekter af de 3 konkrete typer. Tilføj bilerne til flåden. Skriv alle bilerne ud, og kald også getTotalRegistrationFeeForFleet() -metoden og skriv resultatet ud.

Aflevering: IntelliJ-projektet eksporteres som zip-fil og uploades på Moodle.