

EKSAMENSFORSIDE

Skoleeksamen med tilsyn

Emnekode: PB1090	Emnenavn: Objektorientert programmering	
Dato: 11. oktober 2023	Tid fra / til: 09:00	Ant. timer: 13:00
Ansvarlig faglærer: Lars Erik Opedal (Vestfold) & Joakim Bjørk (Kongsberg)		
Campus: Vestfold og Kongsberg	Fakultet: TNM	
Antall oppgaver: 10	Antall vedlegg: 0	Ant. sider inkl. forside og vedlegg: 3
Tillatte hjelpemidler (jfr. emneplan): Ingen		
Opplysninger om vedlegg:		
Merknader: Studenter på Campus Bakkenteigen skriver Java syntaks Studenter på Campus Kongsberg skriver C++ syntaks Tips. I Wiseflow; bruke vedlegg -> kode og velge Java eller C++ og evt. et farge-tema for syntaks.		

Ved eksamen på papir:
Kryss av for type eksamenspapir

Ruter ☐

Linjer ☐

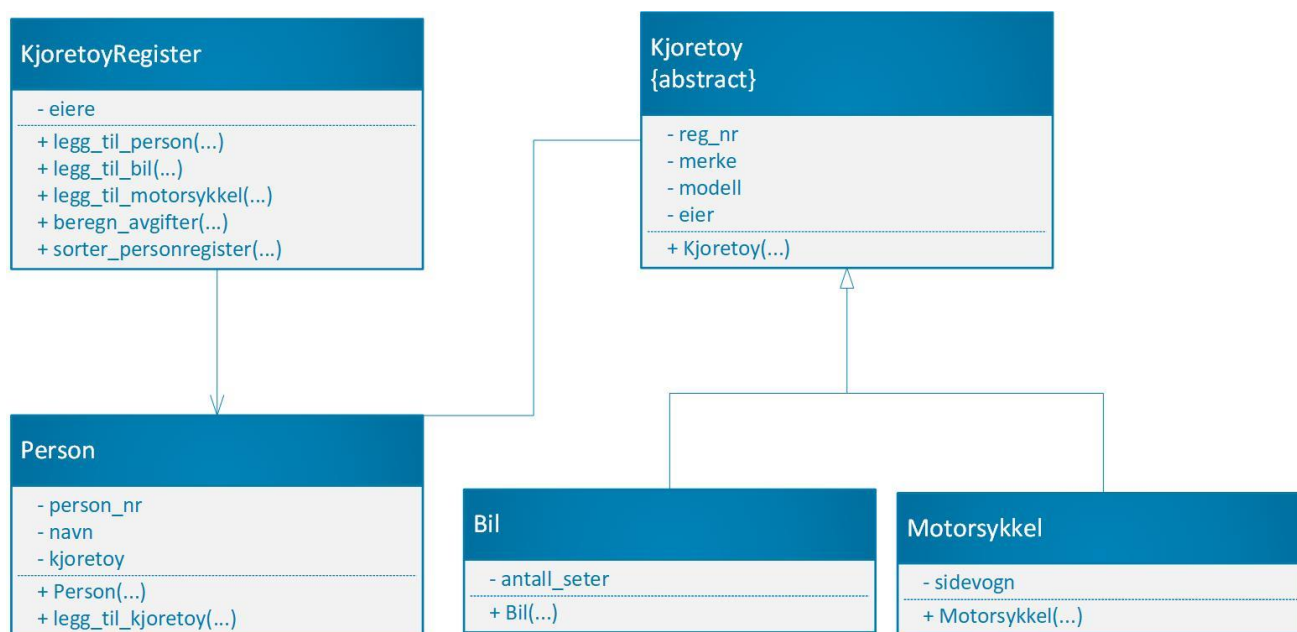
KANDIDATEN MÅ SELV KONTROLLERE AT OPPGAVESETTET ER FULLSTENDIG

Kjøretøyregister (100%)

Statens Vegvesen har behov for et nytt register over registrerte kjøretøy og eiere. En systemarkitekt har kommet opp med designet i diagrammet under og det er din jobb å implementere systemet. Funksjoner og variabler som er private er markert med – og det som er public er markert med +.

I funksjoner med ... som parameterliste må du finne ut hvilke parametere du trenger. Du må også velge en passende datatype for medlemsvariablene og returtyper for funksjonene.

Du kan lage funksjoner/metoder som ikke står oppført i diagrammet om du trenger det.



Oppgave1 (10%) Kjøretøy

Implementer klassen Kjøretøy med konstruktør som gir alle medlemsvariablene verdier. Klassen skal være abstrakt.

Oppgave 2 (10%) Bil og Motorsykkkel

Implementer klassene Bil og Motorsykkkel som begge arver fra Kjøretøy. Du skal også implementere konstruktører som gir alle medlemsvariablene verdier.

Oppgave 3 (10%) Person

Implementer klassen Person. Merk at en person skal kunne eie flere kjøretøy. Du kan selv velge hva slags datastruktur du bruker for å lage en kjøretøyliste. Du skal også lage en konstruktør til klassen.

Oppgave 4 (7%) legg_til_kjøretøy

Implementer funksjonen legg_til_kjøretøy. Funksjonen skal ta en peker/referanse til et kjøretøy og legge det til i listen over kjøretøy personen eier.

Oppgave 5 (7%) Kjøretøyregister

Implementer klassen Kjøretøyregister. Det skal kunne registreres flere eiere i systemet. Du kan selv velge hva slags datastruktur du vil bruke for å opprette eierlisten.

Oppgave 6 (6%) legg_til_person

Implementer funksjonen legg_til_person. Funksjonen skal opprette et nytt person-objekt og legge en peker/referanse til det i listen over eiere.

Oppgave 7 (10%) legg_til_bil

Implementer funksjonen legg_til_bil. Funksjonen skal blant annet ta et personnummer som parameter. Om personnummeret finnes i registeret over eiere så skal et nytt Bil-objekt lages og en peker/referanse til dette legges til i listen over kjøretøy personen eier. Om personnummeret ikke er registrert skal det kastes et egendefinert unntak som skal arve fra exception-klassen.

Oppgave 8 (10%) legg_til_motorsykkkel

Implementer funksjonen legg_til_motorsykkkel. Den skal fungere på samme måte som legg_til_bil.

Oppgave 9 (15%) beregn_avgifter

Implementer funksjonen beregn_avgifter. Funksjonen skal gå gjennom alle eiere og finne ut hvor mye de skal betale i avgift. Personnummer, navn og beløp for hver eier i systemet skal skrives til fil. Det skal også skrives ut hvor mye som skal betales totalt for alle eiere til sammen. Om vi kaller funksjonen slik:

```
biltilsynet.beregn_avgifter(3000.0, 1500.0);
```

Kan filen f.eks. se slik ut:

```
12129412345 Gunnar Jensen skal betale 4500kr  
05088711548 Berit Hansen skal betale 6000kr  
10049978901 Helle Vik skal betale 0kr
```

```
Totalt skal det betales 10500kr
```

Oppgave 10 (15%) sorter_personregister

Implementer funksjonen sorter_bilregister. Funksjonen skal sortere listen over eiere etter hvor mange kjøretøy de har registrert på seg. Personen med flest kjøretøy skal ligge først i lista osv. Du kan gjerne bruke en innebygd sorteringsfunksjon.

Lykke til!