

IN1010 Objektorientert programmering – eksamen våren 2024

Oppgavetekst fra UiOs eksamen våren 2024, IN1010 (overskriften er eksamenssett.no sin; utsnittene under er klippet uendret ut av UiOs publiserte dokument)

Eksamen IN1010, våren 2024

I denne oppgaven skal du skrive deler av et lite program som behandler fly.

Les gjennom hele oppgaven før du begynner på besvarelsen. Tegning av klassehierarkiet skal leveres i oppgave 1, samtlige andre svar (det vil si all programkode) skrives i svarfeltet for oppgave 2–5. Angi gjerne hvilken oppgave du svarer på som kommentarer i programkoden.

Du kan bruke norske bokstaver i denne besvarelsen.

Oppgave 1. Klassehierarki for fly (5 poeng)

Fly representeres av klassen **Fly** som har følgende metoder:

hentId returnerer en String som identifiserer dette flyobjektet, f. eks. «LN-ABC».

hentMotorAntall returnerer et heltall som angir hvor mange motorer flyet har.

hentMTOW returnerer et heltall som angir flyets maksimale startvekt i kilogram som på engelsk heter *Maximum Take Off Weight* forkortet til MTOW.

MTOW, antal motorer (heltall) og deres totale trekkraft (heltall) i kW (kilowatt) og identifikatoren skal være parametere til klassens konstruktør. Flyobjekter skal kunne hektes sammen i en enkeltlenket liste i en senere oppgave, så du skal også ha med instansvariabler i klassen som gjør dette mulig.

I dette eksamenssettet er alt flymateriell enten av klassen **MotorFly** eller klassen **SeilFly**. Alle MotorFly-objekter er enten av klassen **LasteFly** eller klassen **PassasjerFly**.

Fly kan implementere et interface **Motordrevet** med følgende grensesnitt:

trekkraft som returnerer maksimal trekkraft av flyets motorer i antall kW som et heltall.

LasteFly har en maks tillatt lastevekt, angitt som et heltall i kilogram, og PassasjerFly har et heltall som representerer maks antall passasjerer det er plass til i flyet. Seilfly har et heltall som representerer minimum synkehastighet målt i cm/s.

Disse instansvariablene skal være parametere til konstruktørene til klassene.

- ⇒ Tegn et klassehierarki for situasjonen beskrevet over.

Oppgave 2. Programmere klasser og interface (15 poeng)

I denne oppgaven skal du programmere klasser og interface som beskrevet i oppgave 1.

- ⇒ Programmer klassene **Fly**, **MotorFly**, **SeilFly**, **LasteFly**, **PassasjerFly** og **Motordrevet**.

Oppgave 3. Klassen Flyformasjon (28 poeng)

I oppgave 3 skal du skrive en klasse **Flyformasjon** med variabler og metoder. En Flyformasjon er en enkelthetet lenket liste av Flyobjekter og du skal bruke instansvariablene i disse objektene til å lage denne listen. I denne oppgaven kan du aksessere instansvariabler i Fly-klassen og dens subklasser.

3a) 1 poeng

- ⇒ Skriv konstruktør og instansvariabler for klassen Flyformasjon med peker til første Fly-objekt i listen.

3b) 3 poeng

- ⇒ Skriv metoden **leggTil** i klassen Flyformasjon som legger til et Flyobjekt først i listen. Parameter til metoden er en referanse til objektet som skal legges til.

3c) 2 poeng

- ⇒ Skriv metoden **erMed** i klassen Flyformasjon som sjekker om et gitt fly er med i Flyformasjonen. Parameter til metoden er en tekststreng som er flyets identifikasjon. Metoden skal returnere true, hvis flyet er med i Flyformasjonen, og ellers false.

3d) 6 poeng

- ⇒ Skriv metoden **taUt** i klassen Flyformasjon som tar ut et element i listen. Parameter til metoden er en tekststreng som er identifikasjonen til flyet som skal tas ut. Metoden skal returnere en referanse til det objektet som tas ut. Hvis flyet ikke er med, så skal det returnes NULL.

3e) 8 poeng

- ⇒ Skriv en iterator over alle Fly-objektene i en Flyformasjon.

3f) 8 poeng

- ⇒ Skriv metoden **PassasjerFly[] hentPassasjerFly()** i klassen Flyformasjon som returnerer en array med referanser til alle PassasjerFly-objekter i Flyformasjonen.
Hint: Tell antall passasjerfly før du oppretter arrayen.

Oppgave 4. Maksimum vekt (16 poeng)

4a) 8 poeng

Legg til en metode, **totalVekt**, til Flyformasjon. Metoden skal summere MTOW for alle fly i Flyformasjonen. Metoden skal bruke din iterator fra 3 e).

- ⇒ Skriv metoden **totalVekt**.

4b) 8 poeng

Legg til en metode, **maksVekt**, til Flyformasjon. Metoden skal returnere den største MTOW blant alle fly i Flyformasjonen. Metoden skal gå gjennom listen av fly rekursivt.

- ⇒ Skriv metoden **maksVekt** rekursivt.

Oppgave 5. Tråder (36 poeng)

I oppgave 5 skal du skrive én trådklasse som heter **Flygeleder** og én som heter **Pilot**. Begge disse skal implementere Runnable. Du skal også skrive en klasse **Rullebane** som skal være en monitor. (Du skal altså ikke skrive noe fullstendig program som starter opp noen tråder – du skal bare skrive disse tre klassene med metoder og variabler.)

5 a) 10 poeng

Klassen **Rullebane** skal være en monitor hvor Flygeleder kan se hvilke fly som ønsker å starte og gir tillatelse til fly, så de kan starte. Flygelederen gir starttillatelse til ett fly om gangen ved å kalle **sjekkAvganger**. Etter retur fra **sjekkAvganger** venter Flygelederen 1 minutt (så flyet kan starte og forlate rullebanen) og kaller deretter **sjekkAvganger** igjen – for så å gi det neste flyet tillatelse ☺

Klassen **Rullebane** må ha følgende metoder:

sjekkAvganger som kalles av Flygelederen for å sjekke om det er fly som venter på starttillatelse. Hvis ikke returner Flygelederen – venter 1 minutt og deretter kaller **sjekkAvganger** igjen.

Hvis det er et eller flere fly som venter, så gir flygelederen tillatelse til å starte ved å signalere ett av de ventende flyene via en Condition variabel, **avventStartTillatelse**. Deretter returner Flygelederen – venter 1 minutt og deretter kaller **sjekkAvganger** igjen.

Metoden har ingen parametere.

hentStartTillatelse som kalles av Pilotene, når de gjerne vil starte. De venter i en Conditionvariabel **avventStartTillatelse**, som Flygelederen signalerer når et fly har fått starttillatelse.

Metoden har én parameter som er et Flyobjekt.

⇒ Skriv monitoren **Rullebane**, bruk **Condition** og **ReentrantLock**; *ikke* bruk synchronized.

5 b) 7 poeng

⇒ Skriv trådklassen **Flygeleder**, som beskrevet i 5 a).

5 c) 7 poeng

⇒ Skriv trådklassen **Pilot**, som beskrevet i 5 a). Parameter til konstruktøren til Pilot er en referanse til objekt. Antag at en pilot kun har ETT fly.

5 d) 12 poeng

Flygelederen bruker «busy waiting» og kaller **sjekkAvganger** hvert minutt.

Modifiser **sjekkAvganger** så HVIS det ikke er et fly som venter på starttillatelse, så skal Flygelederen vente på en Condition **avventFly**. Denne signaleres så av Piloter når de gjerne vil ha starttillatelse.

Modifiser **hentStartTillatelse** så Piloterne signalerer flygelederen, som så gir tillatelse.

⇒ Omskriv **sjekkAvganger** og **hentStartTillatelse** som beskrevet ovenfor.