Konteeksamen IN1010, august 2024

Introduksjon til seilfly

Du skal skrive deler av et lite program om seilfly. Seilfly er fly som ikke er avhengige av motorer for å fly – de fleste seilfly har ikke en motor. I stedet er de avhengige av naturlig forekommende oppvind generert av solvarme som en «strømkilde».



Et seilfly

Akkurat som noen seilbåter har en liten motor for å hjelpe når det ikke er vind, eller når de kommer inn i en trang havn, har noen seilfly en motor enten for å kunne fly deler av tiden som et motorfly, eller kun brukes til start, eller brukes på slutten av dagen, når oppvind som genereres av solen, svekkes eller forsvinner på slutten av dagen – motoren lar deretter seilflyet fly tilbake til hjemmeflyplassen. Hvis motoren er veldig liten, kan den bare brukes til å opprettholde vannrett flyvning – vanligvis brukt når oppvindene dør ut på slutten av en flydag og deretter brukes motoren til å fly tilbake til hjemmeflyplassen – disse typene seilfly kalles Self-Sustaining Gliders (SSG). Andre seilfly har kraftigere motorer som lar seilflyet ta av fra en flyplass og deretter slå av motoren og fortsette flyturen som et motorløst seilfly – disse typene seilfly kalles Self-Launching Gliders (SLG). Enda en kategori av seilfly har enda kraftigere motorer og er i stand til å fly mange timer ved å bruke motoren som hovedkraftkilde – de er i hovedsak nesten små motorfly – disse seilflyene kalles Touring Motor Gliders (TMG). Oppsummert:

TMG (TMG er et akronym for Touring Motor Glider) representerer seilfly som har kraftige motorer og kan fly lange avstander som et motorfly.

SLG (SLG er et akronym for Self-Launching Gliders) representerer seilfly som har en motor som er kraftig nok til at seilflyet kan ta av på egen kraft. Vanligvis, etter å ha klatret til 500 meter, slås motoren av og propellen og motoren trekkes inn i kroppen til seilflyet. Vanligvis er motoren og propellen montert på en stang som trekkes inn i kroppen til seilflyet, hvoretter seilflyet i hovedsak er den samme som en motorløs glider med lignende ytelse (selv om seilflyet er litt tyngre på grunn av motoren).

SSG (SSG er et akronym for Self-Sustaining Gliders) representerer seilfly som ikke kan starte selv, men som har en liten, lett motor som er (som SLG-er) montert på en stang som trekkes ut av seilflyets kropp og trekkes tilbake når motoren ikke brukes. Motoren brukes vanligvis på slutten av en flydag for å komme tilbake til flyplassen.

Instruksjoner
Lese hele settet før du forsøker å svare *noe som helst*. Din tegning av klassehierarki skal være levert som problem 1 og alle de annen svarene (*dvs.* all programkode) må være skrevet i svarboksen for oppgave 2–5. Vennligst oppgi hvilken oppgave du løser som kommentarer i programkoden.

Du kan bruke norsk bokstaver (æ, ø, å) i ditt svar.

Problem 1. Et klasshierarki for seilfly (6 poeng)

Du skal etablere et klassehierarki som representerer ulike typer seilfly.

Seilfly skal være representert ved en abstrakt klasse **Seilfly**, som har følgende metoder:

hentId returnerer en String som identifiserer dette objekt.

HentGlidetall returnerer et heltall, glidetall, som spesifiserer hvor mange meter seilflyet flyr fremover for hver meter det synker. Gode seilfly har glidetall på 40 mens seilfly i verdensklasse har glidetall oppover til 60 eller 70.

hentVingespenn returnerer et heltall som indikerer vingespennet til seilflyet i meter. Typiske vingespenn er 15 meter opp til 31 meter.

Seilfly kan kobles til i en dobbeltlenket liste i et seinere problem, så du vil også ha behov for variabler i klassen for å gjøre dette mulig.

Rene seilfly, seilfly uten motorer, er representert som en underklasse av **Seilfly** kalt **EkteSeilfly.** Seilfly med motor er representert av en abstrakt underklasse av **Seilfly** , kalt **MotorSeilfly**, og det er tre underklasser: **TMG** , **SLG** , **SSG**.

Objekter av klasse **MotorSeilfly** må implementere et grensesnitt **Motordrevet** med disse metodene:

trekkraft returnerer et heltall som viser motorens totale effekt gitt i kW (kilowatt).
motortype returnerer en tekst som angir motortypen. Motortyper er "stempel", "elektrisk" eller "jet".

- ⇨ Tegne et klassehierarki med de klassene som er beskrevet over. Inkluder **Motordrevet-**grensesnittet.

Problem 2. Programmer klasser og grensesnitt (16 poeng)

I denne oppgaven skal du programmere klasser og grensesnitt beskrevet i problem 1. Du skal inkludere passende variabler og du skal skrive både en konstruktør og metodene til klassene.

- ⇨ Programmer klassene **Seilfly, MotorSeilfly , EkteSeilfly , TMG, SLG, SSG** og grensesnittet **Motordrevet**

Problem 3. Klasse Konkurransegruppe (28 poeng)

I oppgave 3 skal du skrive en klasse **Konkurransegruppe** («gruppe av seilfly som deltar i en konkurranse») med dens variabler og metoder. En **Konkurransegruppe** inneholder en dobbeltlenket liste av **Seilfly**-objekter. I denne oppgaven har du tilgang til variablene i klassen **Seilfly** og dens underklasser. Du kan anta at hver instans av **Seilfly** er med i kun ÉN **Konkurransegruppe.**
Du får bruk for tilgang til de variablene i **Seilfly** som handler om dobbeltlenket liste. Sjekk at du har dem med i løsningen til problem 2.

3a) 1 poeng

- ⇨ Skriv en konstruktør og variabler til klassen **Konkurransegruppe.**

3b) 3 poeng

⇨ Skriv en metode **leggTil** i klassen **Konkurransegruppe.** Metoden er å legge til et **Seilfly-**objekt på slutten av listen. Parameteren til metoden er en referanse til **Seilfly-**objektet som skal legges til.

3c) 2 poeng

⇨ Skriv en metode **erMed** i klassen **Konkurransegruppe.** Parameteren til metoden er en tekststreng som inneholder identifikatoren til et **Seilfly** -objekt. Metoden returnerer **true**, hvis det gitte objektet er i **Konkurransegruppen** , ellers returnerer metoden **false**.

3d) 6 poeng

⇨ Skriv en metode **taUt** i klassen **Konkurransegruppe.** Parameteren til metoden er en tekststreng som inneholder identifikatoren til et **Seilfly-**objekt som skal fjernes fra listen. Metoden skal returnere en referanse til det fjernede **Seilfly-**objektet. Hvis seilflyet ikke er i listen, returnerer metoden **null**.

3e) 8 poeng

⇨ Skriv en iterator som itererer over alle **Seilfly-**objektene i en **Konkurransegruppe.**

3f) 8 poeng

⇨ Skriv en metode **Seilfly [ ] hent EkteSeilfly ()** i klassen **Konkurransegruppe.** Metoden skal returnere en array som inneholder referanser til alle **EkteSeilfly-**objekter i **Konkurransegruppen.**

Hint : Tell antall **EkteSeilfly** før du oppretter arrayet.

Problem 4. Iterasjon og rekursjon (22 poeng)

4a) 5 poeng

Skriv en metode **besteGlidetall** i **Konkurransegruppe.** Metoden skal returnere glidetallet til seilflyet i **Konkurransegruppen** som har best (høyest) glidetall. Metoden må bruke din iterator fra 3 e). Returner **null** hvis **Konkurransegruppen** er tom.

⇨ Skriv metoden **besteGlidetall**.

4b) 7 poeng

Skriv en metode **størstSpennvidde** i **Konkurransegruppe.** Metoden skal returnere spennet til seilflyet i **Konkurransegruppen** som har størst spenn. Metoden må gå gjennom listen *rekursivt.* Returner **null** hvis **konkurransegruppen** er tom.

⇨ Skriv metoden **størstSpennvidde** rekursivt.

4c) 10 poeng

Skriv en metode **histogramSpennvidde** i **Konkurransegruppe** Metoden må returnere en array med heltall der elementet med indeks *i* inneholder antall seilfly i gruppen med spenn *i.* Du kan anta at spennet for en hvilket som helst seilfly er i området 10 til 99 – hvis et seilfly har et spenn utenfor dette området**,** ignorer det. Du er velkommen til å bare bruke en array med fast størrelse.

⇨ Skriv metoden **histogramSpennvidde.**

Problem 5. Tråder (28 poeng)

Ved en seilflykonkurranse er de seilflyene som ønsker å konkurrere, oppført i en **Konkurransegruppe.** Seilflyene stilles opp av en plassjef som setter dem inn i en startkø, én om gangen. Når tiden er inne for å starte konkurransen, aktiverer plassjefen et sett slepefly (vanlige små énmotorsfly med slepekrok i halen). Hvert slepefly ber om et seilfly fra startkøen, sleper det opp i luften, lander igjen og ber om et annet seilfly å slepe inntil køen er tom.

Her vil vi ha plassjefen representert ved en instans av trådklassen **Plassjef** og slepeflyene av en annen trådklasse **SlepePilot**. Begge klassene må implementere Runnable. Du må også skrive en klasse **KlarTilStart** (for «seilfly klare til å ta av») som skal være monitor. (Du trenger ikke å skrive et komplett program som starter tråder – du skal bare å skrive de tre klassene.)

Monitoren **KlarTilStart** skal opprettholde en kø av seilfly som er klare for start og en kø av slepefly som er klare til å slepe så snart det er et seilfly som er klar for start.

Når en **Plassjef** opprettes, oppretter objektet først et antall slepefly (antallet er gitt som en parameter til konstruktøren av **Plassjef)**. Deretter itererer **Plassjefen** gjennom en **Konkurransegruppe**, som er en parameter for **Plassjef**, og legger inn hver av seilflyene ett etter ett inn i **KlarTilStart**-monitoren (som også er en parameter for Plassjefen) via en metode, **seilflyklar**, som har en enkelt parameter, en instans av **Seilfly** som er klar for start. Etter at **Plassjef-**instansen har lagt inn alle seilfly, avsluttes den.

Konstruktøren av klassen **SlepePilot** har en enkelt parameter, en instans av klassen **KlarTilStart.**
Når en instans av **SlepePilot** starter, vil den be gjentatte ganger om et seilfly fra køen ved å kalle en metode **nesteSeilfly** (i instansen av **KlarTilStart**), som returnerer neste **Seilfly** som er klar for start. Hvis det ikke er noe seilfly som venter på å ta av, venter slepeflyet i en slepeflykø. Etter hvert slep venter den 6 minutter (simulerer at et slep tar 6 minutter) før den ber om et annet seilfly å slepe.
Hvis det ikke er flere **Seilfly** i seilflykøen, venter slepeflyet i slepeflykøen. (Foreløpig bekymrer vi oss ikke for å tømme slepeflykøen på slutten av dagen.)

5 a) 14 poeng

⇨ Skriv metoden **KlarTilStart ,** som beskrevet ovenfor, bruk bare **Condition** og **ReentrantLock ;** *ikke bruk* **synchronized**.

5 b) 7 poeng

⇨ Skriv trådklassen **Plassjef**, som beskrevet ovenfor.

5 c) 7 poeng

⇨ Skriv trådklassen **SlepePilot**, som beskrevet ovenfor.