

i Om eksamen**Eksamen IN1010/INF1010 våren 2018****PRAKTISK INFORMASJON**

Hvis du synes noen deler av oppgaven er uklare, kan du legge dine egne forutsetninger til grunn og gjøre dine egne rimelige antagelser. Gjør i så fall rede for disse forutsetningene og antagelsene.

Les alle deloppgavene før du begynner å besvare dem. Det er informasjon i senere deloppgaver som kan brukes i besvarelsen av tidligere oppgaver.

Unngå å bruke en stor del av tiden din på oppgaver som gir deg få poeng.

Faglærerne prøver å besøker eksamenslokalet mellom klokka 10 og 11 for å oppklare eventuelle uklarheter og feil i oppgaveteksten.

Tillatte hjelpemidler

Alle trykte og skrevne hjelpemidler er tillatt, samt en enkel lommekalkulator.

Tegning på papir

I oppgave 3 skal svaret tegnes på papir (skisseark). Instruksjon om utfylling av skisseark finner du på pulten din. Husk å notere kodennummeret og annen informasjon med én gang; du vil ikke få tid til å gjøre dette etter at eksamen er over. Det blir *ikke* gitt ekstratid for å fylle ut informasjonsboksene på skisseark (engangskoder, kandidatnummer etc).

Om du har behov for å tegne andre datastrukturer (tilhørende andre oppgaver enn nr 3) for å vise sensor hvordan du tenker når du skriver algoritmer, så tegner du dette etter svaret på oppgave 3, skriver tydelig på tegningen hvilken/hvilke oppgave(r) tegningen tilhører, og leverer alt som svar på oppgave 3.

Blandede tekst/kodesvar

I noen oppgaver skal du både skrive tekst og program. Du kan da skrive teksten inn i «programboksen» uten å passe på riktig Java-syntaks, dvs. du behøver ikke kommentere ut teksten du skriver.

1 Leiting i tekst

Skriv en klasse **Stringhjelper** som inneholder en statisk metode kalt **inneholder**. Denne metoden skal returnere en **int** og ha to parametre, **s** og **t**, begge av type **String**. Metoden skal undersøke om teksten **t** forekommer sammenhengende i teksten **s**. Om den gjør det, skal metoden returnere indeksen (posisjonen) i **s** der **t** starter. Om **t** ikke finnes i **s**, skal metoden returnere -1.

For eksempel forekommer teksten "EI" sammenhengende i teksten "STEIN", og den starter i 2. posisjon; husk at posisjonene starter fra 0. Teksten "SEN" forekommer ikke sammenhengende i "STEIN".

I denne oppgaven får du ikke bruke andre metoder i **String**-klassen enn **charAt(i)** som henter tegnet i posisjon nr **i** i teksten, og **length()**.

Maks poeng: 5

2 Prioritetskø

Skriv klassen **Prioritetskoe<T>** uten bruk av noe fra Javas bibliotek.

Klassen skal ha tre metoder: **setInn(T inn, int prio)** og **T taUt()** og **int antall()**. Implementasjonen skal være en lenket liste. Prioriteten er et positivt heltall eller 0. 0 er høyeste prioritet.

Den lenkede listen skal være sortert på prioritet slik at alle med høy prioritet kommer foran alle med lav prioritet.

taUt() skal helst være en veldig rask operasjon som ikke behøver lete eller gå gjennom listen, den skal bare ta ut det første elementet i listen og dermed også et av de elementene som har høyeste prioritet.

Rekkefølgen blant de elementene som har lik prioritet spiller ingen rolle.

Maks poeng: 10

3 Datastruktur

Les vedlagte beskrivelse og finn frem til en datastruktur som er egnet til å løse oppgaven. Tegn datastrukturen med objekter og interface-er slik den ser ut for en flyvning med det lille flyet **Mini-1** som bare har tre seterader (se neste oppgave) og der fire passasjerer har reservert plass. Bruk utdelte papirark.

Maks poeng: 10

4 Flytyper

Skriv klassen **Flytype**. Klassen skal inneholde disse elementene:

- en konstruktør
- **String type** er fabrikkens navn på flytypen.
- **String seteinfo** er en kompakt representasjon av hvilke seter som finnes i flyet: vi regner med at alle fly av samme type har samme setekonfigurasjon. For hver seterad angis radnummeret og deretter bokstavene til setene på raden. Følgende spesialtegn brukes:
 - | skiller seteradene
 - * angir en midtgang
 - + etter et sete angir at det har god benplass.

Et eksempel: "2:AC*D+F+|3:C*D|4:AC*DF" angir at

- Rad 1 finnes ikke.
- Rad 2 har fire seter, to av dem (A og C) til venstre for midtgangen og de andre to (D og F) til høyre for midtgangen. De to sistnevnte setene har ekstra god benplass.
- Rad 3 har kun to seter (C og D), ett på hver side av midtgangen.
- Rad 4 har fire seter, to av dem (A og C) til venstre for midtgangen og de to andre (D og F) til høyre for midtgangen.
- **Flygning opprettFlygning()** skal lage et Flygning-objekt og initiere det med alle **Seterad** og **Sete**-objektene til (de foreløpig tomme) seter til alle passasjerene.

Maks poeng: 12

5 Flygninger

Skriv klassen **Flygning** som skal inneholde

- En datastruktur for alle seteradene ombord.
- En konstruktør som oppretter alle **Seterad**- og **Sete**-objektene i flyet i henhold til en **String**-spesifikasjon som angitt i forrige oppgave.

- **String flightNo** er flygningens flight-nummer.
- **boolean book(Passasjer pas)** finner et ledig sete til passasjeren og reserverer det for ham eller henne. Returverdien forteller om det var en ledig plass til vedkommende. Vi tar ikke hensyn til passasjerens høyde.
- **boolean book(Passasjer pas, boolean foretrekkerVindu, boolean foretrekkerMidtgang)** prøver å finne et egnet sete til passasjeren i henhold til vedkommendes ønsker, og at en høy passasjer får et sete med ekstra benplass. Metoden returnerer **true** hvis det var vellykket.
- **boolean book(Passasjer pas1, Passasjer pas2)** prøver å finne sete til de to passasjerene ved siden av hverandre, og ikke på hver sin side av midtgangen. Vi tar ikke hensyn til om noen av de to passasjerene er høye.
- **void skriv()** skal skrive ut (med **System.out.println**) en fullstendig passasjerliste, med navn og seteinformasjon for alle passasjerene.

Maks poeng: 21

6 Seterader

Programmer klassen **Seterad** som inneholder data om én setetrad; den skal blant annet inneholde minst

- **int nr** er radens nummer.
- referanser til setene i denne raden

Maks poeng: 3

7 Seter

Skriv klassene **Sete** og **GodBenplassSete** De skal minst inneholde

- **boolean erVindussete()** angir om setet står i den ene enden av raden.
- en referanse til passasjeren som har reservert setet (eller **null** om setet ennå er ledig).

Skriv også interface-et **MidtgangSete** som skal ha

- **boolean erVedMidtgang()** angir om setet er ved midtgangen.

Maks poeng: 3

8 Passasjerer

Skriv klassen **Passasjer**. Den skal inneholde data om passasjeren, i hvert fall

- **String navn** er passasjerens navn.
- **double hoyde** er passasjerens høyde i cm.
- **boolean harLangeBen()** forteller om passasjeren er 190 cm høy eller mer og dermed helst vil ha et sete med plass til lange ben.

Maks poeng: 3

9 En iterator

Anta at klassen **Flygning** endres slik at den nå **implements Iterable<Sete>** slik at vi kan iterere over alle setene i flyet på denne flygningen . Programmer klassen på nytt med denne endringen.

Maks poeng: 11

10 Søk etter terrorister

Det er et krav at flyreservasjonsprogrammer må kunne brukes til hjelpe politiet med å finne mulige terrorister.

Du skal skrive to trådklasser samt en monitorklasse og et hovedprogram til dette bruk. Den første trådklassen skal hete **Forstelinjevokter**. Alle Forstelinjevokter-trådene skal samtidig søke gjennom hver sin flygning. Når en Forstelinjevokter finner en mistenkelig person (hva det vil si skal vi definere senere), skal tråden legge den mistenkelige personen inn i en monitor av klassen **MistenkeligePersoner**. Objekter av den andre trådklassen, kalt **Andrelinjevokter**, skal hente personer ut av denne monitoren og kalle en statisk metode i hovedprogramklassen din med navn **mistenkelig()** der disse personene kommer til å bli nøye undersøkt av politiet. (Du skal ikke skrive denne metoden.)

Hovedprogrammet skal altså brukes når politiet får tips om en sammenhengende del av navnet til en terrorist som skal ut og fly. Denne navnedelen gis som parameter til programmet og blir dermed tilgjengelig i parameter **args[0]** i hovedprogrammet når det er deklart som

```
public static void main(String[] args) { ... }
```

La programmet gå gjennom alle flygningene og opprette og starte en Forstelinjevokter-tråd for hver flygning. For enkelthets skyld kan du anta at programmet kan få tak i en iterator over alle flygningene til flyselskapet ved å kalle den statiske metoden `alleFlygninger()` i hovedprogramklassen (du skal ikke programmere denne metoden).

Den sammenhengende delen av et navn som politiet får tips om, kalles et *mønster*. Jo tidligere i navnet til den mistenkte personen dette mønsteret forekommer, jo mer sannsynlig er det at personen er en terrorist. Når en Forstelinjevokter-tråd legger en mistenkt person inn i monitoren, skal avstanden fra starten av navnet til starten av mønsteret brukes som prioritet. F.eks. vil STEIN fra oppgave 1 få prioritet 2 når mønsteret er EI. Som i oppgave 2 betyr et lite tall høy prioritet.

Programmet ditt skal også starte 100 Andrelinjevokter-tråder som tar mistenkte ut av monitoren, og da skal de mistenkte med høyest prioritet tas ut først. Om monitoren er tom, skal trådene vente på at en ny person blir lagt inn.

Det er ikke nødvendig å tenke på at trådene skal terminere.

Maks poeng: 22

Vi skal lage et reservasjonssystem for det nye flyselskapet *Billig-tur*. System skal holde oversikt over hvilke fly selskapet har og ta imot bestillinger til gitte flyvninger med disse flyene. Dessuten skal systemet gi mulighet til rasket å lete etter mulige terrorister ombord.

Systemet vårt skal inneholde minst disse klassene/interface-ene:

Flytype definerer en flytype, for eksempel **Airbus A330-300 (333)** eller **Bombardier CRJ-900 (CR9)**.

Flygning definerer en enkeltreise og holder oversikt over hvilke passasjerer som sitter i hvilke seter.

Seterad definerer en seterad i en flygning; de er nummerert 1, 2, 3 osv (men det er ikke sikkert at alle numrene er i bruk; for eksempel unngår mange flyselskap rad nr 13).

Sete definerer et sete på en rad i en flygning; setene er nummerert A, B, C osv, men det varierer hvilke bokstaver som er i bruk, selv på forskjellige rader under samme flygning.

GodBenplassSete er et sete spesielt egnet for personer med lange ben.

MidtgangSete er at et sete er ved midtgangen.

Passasjer er en person som bestiller en flyreise.