Innlevering 1 – PG4100

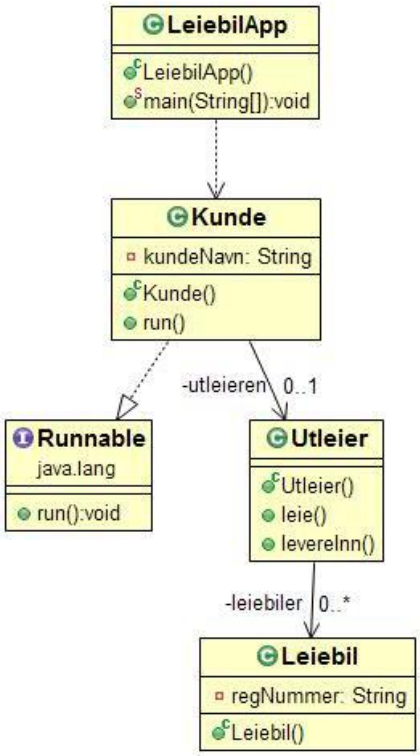
Frist: 8. februar

Oppgaven

Et bilutleiefirma har 5 biler til utleie. Du skal skrive et program som simulerer kunder som kommer og leier bil og siden leverer den inn igjen. Hver `Kunde` skal i programmet implementeres som en tråd. `Utleier`en skal i programmet holde rede på hvilke biler som er utleid og hvilke som er ledige ved å bruke en liste av `Leiebil`-objekter.

`Leiebil` er et VO-objekt (value object) som inneholder opplysninger om leiebilen – blant annet om den er utleid eller ikke.

I klientprogrammet (`LeiebilApp`) skal brukeren av programmet opprette 10 kunder (tråder) ved å skrive inn kundenes navn, en etter en. Kundene skal ikke begynne å leie biler før 5 kunder er opprettet. Hver kunde-tråd skal vente mellom 1 og 10 sekunder før den kontakter utleier for å leie en bil. Når en kunde har leid en bil, skal den vente mellom 1 og 3 sekunder før bilen leveres tilbake igjen. Kundene gjentar (i det "uendelige") å leie og levere inn biler. Klassediagrammet nedenfor viser et forslag til relevante klasser, men du står fritt til å lage de klassene og metodene du selv mener vil simulere utleie på den best mulige måten.



Merk at `LeiebilApp` er den som instansierer `Utleier` og kundetrådene og starter trådene. `Utleier` er klassen med de kritiske dataene. Metoder som manipulerer disse dataene må programmeres trådsikre. Det må hindres at en allerede utleid bil blir leid ut igjen før den er levert inn.

En kundetråd tar kontakt med utleier for å leie en bil ved å kalle på `leie`-metoden i klassen `Utleier`. Hvis ingen biler er ledige, må kunden vente til biler blir ledige.

Ved kall på metoden `levereInn` blir bilen ledig igjen.

Det må altså sørges for at ikke samme bil blir leid ut til flere kunder samtidig, slik som utskriften i eksemplet under viser. For å sjekke at programmet fungerer slik det skal, skal det skrive ut status på alle bilene hver gang en bil blir leid ut og hver gang en bil blir levert inn. Det skal også gi beskjed når en kunde ikke får leie pga at alle biler er utleid. I tillegg skal det lages automatiske tester.

Eksempelutskrift:

```
********** Status for utleiebilene ****************
RF11111 - ledig, RF22222 - ledig, RF33333 - ledig, RF44444 - ledig, RF55555 -
ledig,
********** Status slutt ****************
Per
lars
Bente
Erik
Marit

Bente har leid RF11111.
********** Status for utleiebilene ****************
RF11111 - utleid til Bente, RF22222 - ledig, RF33333 - ledig, RF44444 - ledig,
RF55555 - ledig,
********** Status slutt ****************

Bente har levert inn RF11111.
********** Status for utleiebilene ****************
RF11111 - ledig, RF22222 - ledig, RF33333 - ledig, RF44444 - ledig, RF55555 -
ledig,
********** Status slutt ****************

Marit har leid RF11111.
********** Status for utleiebilene ****************
RF11111 - utleid til Marit, RF22222 - ledig, RF33333 - ledig, RF44444 - ledig,
RF55555 - ledig,
********** Status slutt ****************

Per har leid RF22222.
********** Status for utleiebilene ****************
RF11111 - utleid til Marit, RF22222 - utleid til Per, RF33333 - ledig, RF44444 -
ledig, RF55555 - ledig,
********** Status slutt ****************

Erik har leid RF33333.
********** Status for utleiebilene ****************
RF11111 - utleid til Marit, RF22222 - utleid til Per, RF33333 - utleid til Erik,
RF44444 - ledig, RF55555 - ledig,
********** Status slutt ****************

lars har leid RF44444.
********** Status for utleiebilene ****************
RF11111 - utleid til Marit, RF22222 - utleid til Per, RF33333 - utleid til Erik,
RF44444 - utleid til lars, RF55555 - ledig,
********** Status slutt ****************

lars har levert inn RF44444.
```

```
********** Status for utleiebilene *****************
RF11111 - utleid til Marit, RF22222 - utleid til Per, RF33333 - utleid til Erik,
RF44444 - ledig, RF55555 - ledig,
********** Status slutt *****************
```

Krav til innleveringen

1. Kildekode til klassene og testklassene.
2. Eventuelle referanser til hvilke kilder som er brukt.
3. Et kortfattet notat der du beskriver:
    a. Eventuelle forutsetninger for å kunne teste løsningen.
    b. Kommentarer til eget resultat. Kunne noe vært gjort bedre?
    c. Eventuelle spørsmål du trenger svar på for å komme videre.
4. Valgritt: En video-capture av presentasjon av løsningen (koden og testene); ca 5-10 minutters varighet. **Videoen bør fokusere på hvordan relevante klasser i programmet er implementert trådsikkert. Kritisk område bør i den anledning beskrives.** Screencast-o-matic anbefales som verktøy for video-opptaket, men andre verktøy kan benyttes. En lenke til videoen skal leveres – ikke videoen selv. Husk å åpne for tilgang til videoen («unlisted» i Youtube).
   Hvis video ikke leveres skal man levere et mer utfyllende notat der man gjør rede for løsningen sin, med hovedvekt på forklaring av trådhåndteringen i programmet.

Vurdering

Formålet for innlevering 1 er å gi studenten mulighet til å vise sin måloppnåelse, særlig relatert til trådhåndtering. PG4100 vurderes 60% basert på mappevurdering. Innlevering 1 vil være en del av denne mappen. Studenten vil få en vurdering av innlevering 1 og vil deretter få mulighet til å utbedre løsningen inntil endelig levering av mappen. Se egen vurderingsmatrise for en oversikt over vektleggingen av ulike elementer ved vurdering av oppgaven.

Relevans ift emnebeskrivelsen – innlevering 1

Fra emnebeskrivlesen, PG4100. Punkter som er uthevet har særlig relevans i innlevering 1.

Læringsutbytte

Relevante kunnskaper

Etter å ha fullført emnet skal studenten

- vite hvilke fordeler trådprogrammering kan gi og når det lønner seg å benytte dette

Ferdigheter

Etter å ha fullført emnet skal studenten kunne

- **utvikle trådbaserte applikasjoner**

Generell kompetanse

Etter å ha fullført emnet skal studenten kunne

- **begrunne og presentere sin egen kode**