DAT108 Oblig2 h25 - Tråder

Sist oppdatert av Lars-Petter Helland 08.09.2025

Innleveringsfrist er søndag 21. september. Vi har som mål å rette innleveringene og godkjenne innen 2 uker etter frist.

Øvingen skal gjennomføres i grupper på **2-4** studenter.

NB! Det er nytt gruppesett for hver innlevering.
Gruppesettet for Oblig2 er **«DAT108 Oblig2 gruppe x».**

Det er i utgangspunktet **ikke** anledning til å levere alene. Dere oppfordres til å finne samarbeidspartnere selv, men vi vil hjelpe til ved behov. Grupper med én student kan kun godkjennes etter avtale.

Send en søknad til lærer (Lars-P) ved ønske om gruppestørrelse utenfor 2-4.

Innleveringen er en zip i Canvas.
NB! Zip-en skal hete **Oblig2_gr17.zip** for gruppe 17 osv. (dere forstår) ...
Denne skal inneholde:

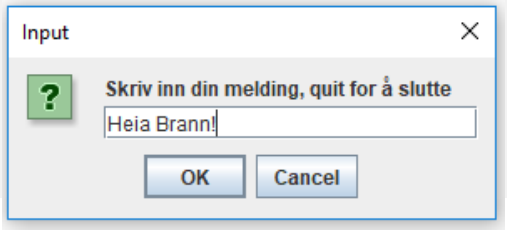
1. **Et pdf-dokument** med:
    a) En liste av hvem som er med i gruppen (for å unngå gratispassasjerer).
    b) Skjermutskrift fra kjøringer av programmene.
    c) Svar på eventuelle teorispørsmål.
2. **Et Eclipse-Java-prosjekt** med løsninger på programmeringsoppgavene.

Oppgave 1 – System.out.println og JOptionPane

Dere skal lage et lite program der vi har én tråd som skriver ut en linje på skjermen hvert 3. sekund, og én tråd som gjentatte ganger lar brukeren taste inn via JOptionPane hva som skal skrives ut. Det skal være en egen «kommando» (f.eks. quit) for å avslutte hele programmet.

Nedenfor er et eksempel på kjøring der programmet først skriver ut standardmeldingen «Hallo verden!», og deretter skriver ut det som bruker har tastet inn («Heia Brann!»). Brukeren skal ha anledning til å taste inn nye meldinger så mange ganger han vil samtidig som forrige melding skrives ut om igjen og om igjen på skjermen.

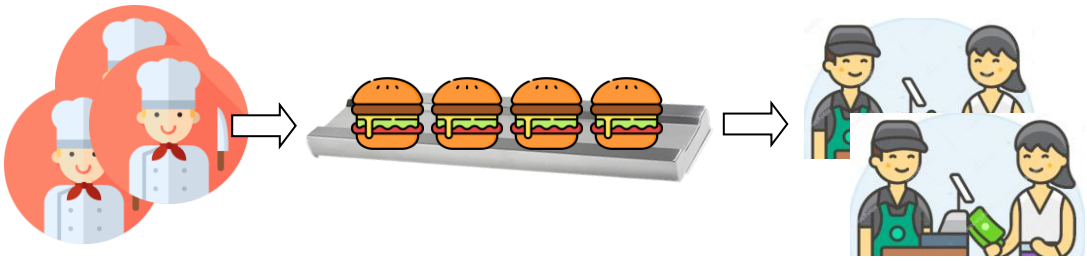
```
Hallo verden!
Hallo verden!
Hallo verden!
Hallo verden!
Hallo verden!
Heia Brann!
Heia Brann!
Heia Brann!
Heia Brann!

...
```



Oppgave 2 – Hamburgersjappe

Dere skal lage et lite program som simulerer en hamburger-sjappe.

Hamburger-sjappen har et **brett** (kø) der ferdige **hamburger**e legges på av **kokk**en(e) etter hvert som de er ferdige og tas av av **servitør**en(e) etter hvert som kunder kjøper en hamburger.



Utskriften fra en simulering skal se ut ca. slik:

```
I denne simuleringen har vi
       3 kokker [Anne, Erik, Knut]
       2 servitører [Mia, Per]
       Kapasiteten til brettet er 4 hamburgere.
Vi starter ...

Anne (kokk) legger på hamburger ◖1◗. Brett: [◖1◗]
Mia (servitør) tar av hamburger ◖1◗. Brett: []
Per (servitør) ønsker å ta hamburger, men brett tomt. Venter!
Anne (kokk) legger på hamburger ◖2◗. Brett: [◖2◗]
Per (servitør) tar av hamburger ◖2◗. Brett: []
Per (servitør) ønsker å ta hamburger, men brett tomt. Venter!
Knut (kokk) legger på hamburger ◖3◗. Brett: [◖3◗]
Per (servitør) tar av hamburger ◖3◗. Brett: []
Erik (kokk) legger på hamburger ◖4◗. Brett: [◖4◗]
Per (servitør) tar av hamburger ◖4◗. Brett: []
Knut (kokk) legger på hamburger ◖5◗. Brett: [◖5◗]
Erik (kokk) legger på hamburger ◖6◗. Brett: [◖5◗, ◖6◗]
Mia (servitør) tar av hamburger ◖5◗. Brett: [◖6◗]
Anne (kokk) legger på hamburger ◖7◗. Brett: [◖6◗, ◖7◗]
Knut (kokk) legger på hamburger ◖8◗. Brett: [◖6◗, ◖7◗, ◖8◗]
Per (servitør) tar av hamburger ◖6◗. Brett: [◖7◗, ◖8◗]
Anne (kokk) legger på hamburger ◖9◗. Brett: [◖7◗, ◖8◗, ◖9◗]
Mia (servitør) tar av hamburger ◖7◗. Brett: [◖8◗, ◖9◗]
Erik (kokk) legger på hamburger ◖10◗. Brett: [◖8◗, ◖9◗, ◖10◗]
Knut (kokk) legger på hamburger ◖11◗. Brett: [◖8◗, ◖9◗, ◖10◗, ◖11◗]
Per (servitør) tar av hamburger ◖8◗. Brett: [◖9◗, ◖10◗, ◖11◗]
Erik (kokk) legger på hamburger ◖12◗. Brett: [◖9◗, ◖10◗, ◖11◗, ◖12◗]
Mia (servitør) tar av hamburger ◖9◗. Brett: [◖10◗, ◖11◗, ◖12◗]
Per (servitør) tar av hamburger ◖10◗. Brett: [◖11◗, ◖12◗]
Anne (kokk) legger på hamburger ◖13◗. Brett: [◖11◗, ◖12◗, ◖13◗]
Mia (servitør) tar av hamburger ◖11◗. Brett: [◖12◗, ◖13◗]
Erik (kokk) legger på hamburger ◖14◗. Brett: [◖12◗, ◖13◗, ◖14◗]
Knut (kokk) legger på hamburger ◖15◗. Brett: [◖12◗, ◖13◗, ◖14◗, ◖15◗]
Anne (kokk) klar med hamburger, men brett fullt. Venter!
Per (servitør) tar av hamburger ◖12◗. Brett: [◖13◗, ◖14◗, ◖15◗]
Anne (kokk) legger på hamburger ◖16◗. Brett: [◖13◗, ◖14◗, ◖15◗, ◖16◗]

... (osv) ...
```

Krav til programmet

- Både **Hamburger**, **HamburgerBrett**, **Kokk** og **Servitor** skal være klasser i programmet slik det er naturlig. Kokker og servitører er tråder, mens brettet er en delt ressurs som både kokker og servitører forholder seg til.
- De ulike objektene i programmet skal samarbeide via referanser til hverandre, ikke gjennom globale variabler.
- Programmet må være trådsikkert slik at det ikke blir rot i tellingen og i køen av hamburgere. Kokker og servitører må vente hvis det er nødvendig, ikke bare kjøre videre.
- Hvis brettet er tomt, må servitører vente til det er lagt en ny hamburger på, og hvis brettet er fullt, må kokker vente til det er tatt en hamburger av.
- Det er meningen at dere skal bruke tråd-primitivene **synchronized**, **wait**, **notify** og **notifyAll** til å løse oppgaven. Dere kan ikke bruke klasser fra java.util.concurrent-API-et, f.eks. Lock og BlockingQueue.
- Tiden det tar å lage og bestille en hamburger skal være (random) mellom 2 og 6 sek.

main() kan f.eks. se slik ut:

```
public static void main(String... blablabla) {

      final String[] kokker = {"Anne", "Erik", "Knut"};
      final String[] servitorer = {"Mia", "Per"};
      final int KAPASITET = 4;

      skrivUtHeader(kokker, servitorer, KAPASITET);

      HamburgerBrett brett = new HamburgerBrett(KAPASITET);

      for (String navn : kokker) {
            new Kokk(brett, navn).start();
      }
      for (String navn : servitorer) {
            new Servitor(brett, navn).start();
      }
}
```

Oppgave 3 – Hamburgere med BlockingQueue

Samme oppgave som i Oppgave 2, men her kan/skal dere bruke java.util.concurrent.BlockingQueue i stedet for synchronized, wait, notify.

Merk: Ikke mulig med så detaljert utskrift over hva som foregår (venting og sånn). Utskriften kan f.eks. være slik:

```
I denne simuleringen har vi
       3 kokker [Anne, Erik, Knut]
       2 servitører [Mia, Per]
       Kapasiteten til brettet er 4 hamburgere.
Vi starter ...

Anne (kokk) legger på hamburger ◖1◗. Brett: [◖1◗]
Mia (servitør) tar av hamburger ◖1◗. Brett: []
Anne (kokk) legger på hamburger ◖2◗. Brett: [◖2◗]
Per (servitør) tar av hamburger ◖2◗. Brett: []
Knut (kokk) legger på hamburger ◖3◗. Brett: [◖3◗]
Per (servitør) tar av hamburger ◖3◗. Brett: []
Erik (kokk) legger på hamburger ◖4◗. Brett: [◖4◗]
Per (servitør) tar av hamburger ◖4◗. Brett: []
Knut (kokk) legger på hamburger ◖5◗. Brett: [◖5◗]
Erik (kokk) legger på hamburger ◖6◗. Brett: [◖5◗, ◖6◗]
Mia (servitør) tar av hamburger ◖5◗. Brett: [◖6◗]
Anne (kokk) legger på hamburger ◖7◗. Brett: [◖6◗, ◖7◗]
Knut (kokk) legger på hamburger ◖8◗. Brett: [◖6◗, ◖7◗, ◖8◗]

... (osv) ...
```