

Tråder - Oppgaver

Oppgave 1:

Lag en klasse som implementerer grensesnittet Runnable.

Klassen skal ta inn en id (et heltall) i konstruktøren.

Klassen sin jobb er å skrive ut alle tallene fra og med 0 til og med 10 ganget med id.

Utskriften skal være "<id> * <tall fra og med 0 til og med 10> = <resultat>"

Skrive et hovedprogram som starter opp 11 tråder, med id fra og med 0 og til og med 10.

Oppgave 2:

```
public class Main {
    public static void main(String[] args) {
        Runnable task = new MyTask();
        Thread worker1 = new Thread(task);
        Thread worker2 = new Thread(task);
        worker1.start();
        worker2.start();
    }
}

class MyTask implements Runnable {
    private final int MAX_COUNT = 10000;
    private int sharedCounter = 0;

    public void run() {
        System.out.println("Starting! Shared counter = " + sharedCounter);
        for (int i = 0; i < MAX_COUNT; i++) {
            sharedCounter = sharedCounter + 1;
        }
        System.out.println("Done! Shared counter = " + sharedCounter);
    }
}
```

A. Hva er galt med denne koden her, og hvorfor? (Se nøye på hvordan man oppretter

trådene, og hvor mange MyTask-objekter det er)

B. Prøv å fiks feilen.

Oppgave 3:

Lag et program som starter opp 8 tråder. Alle trådene skal si hei 2 ganger, men ingen av trådene har lov til å si hei den andre gangen før alle trådene er ferdig med å si hei den første gangen.

Oppgave 4:

Utvid oppgave 3, nå skal trådene skrive ut hei 3 ganger. En tråd kan ikke skrive ut hei den tredje gangen, før alle trådene er ferdig med å skrive hei ut den andre gangen.

Oppgave 5 (Stor oppgave):

SIO skal lage et nytt system for å organisere søknader ang. studentboliger, og du har blitt satt på saken.

SoknadsBehandler(e) (tråder) har som jobb å lese inn søknader, og prioritere søknadene på et sted mellom 0 (høyest prioritert), og 9 (lavest prioritert). Søknaden som Soknadsbehandleren får skal leses inn fra fil (du kan anta at filen er riktig), og kommer på formatet navn borIOslo alder rikeForeldre. Eks:

Kari ja 19 nei

Tom nei 37 ja

Hvis søkeren bor i Oslo øker prioriteten med 3, hvis søkeren har rike foreldre øker prioriteten med 2. Hvis søkeren er over 30 år skal prioriteten øke med 2 og hvis søkeren er over 35 skal den økes med ytterligere 2. Soknadsbehandlerens jobb er også å lage en **Soknad**. En søknad trenger navnet på søkeren, prioriteringen, og må i tillegg genererer en unik id kalt soknadsnr (lagd på samme måte som i oblig 2). PS: pass på at dette er trådsikkert (flere Soknadsbehandlere kan opprette Soknader samtidig, ergo må denne idTelleren beskyttes på en måte slik at kun en tråd kan bruke den om gangen). Når søknaden er laget skal den sendes

videre til en **Monitor**. Søknadsbehandlere tar en kaffepause på 10 sek mellom hver søknad (Dette kan du gjøre ved å bruke `Thread.sleep(10000)`). Søknadsbehandleren må også kommunisere til monitoren når vedkommende er ferdig med å jobbe.

Monitor har et array av `LinkedList(er)` (eller deres egne lagde lenkelister) som inneholder **Søknad** objekter. Arrayet har ti plasser, en for hver prioritering. Monitor skal kunne gjøre 2 ting; legge til en ny søknad og hente ut en søknad. Monitoren skal passe på at kun én tråd får endret på lista om gangen. `leggTilSøknad` legger til en søknad i den lenkelista som sammensvarer med prioriteten. Har den prioritet 0, skal den i lenkeliste på plass 0. Hvis søknaden har prioritet 1, skal den i lenkelisten på plass 1 osv. `taUtSøknad` skal ta ut en søknad (hvis det ikke er noen søknader der, må man vente), den skal alltid ta ut søknader med lavest prioritet først.

Boligsjef (tråder) har som jobb å hente ut søknader fra monitoren. Boligsjefen tar inn navnet på studentboligen (se listen med navn under) vedkommende er sjef for. Når boligsjefen har mottatt en søknad skal navnet på boligen skrives ut samt infoen til søknaden. Mellom hver søknad tar boligsjefen seg en pause på 5 sekunder (kan bruke `Thread.sleep(5000)`). Når det ikke er flere søknader igjen skal boligsjefen terminere.

```
static String[] boligNavn = {"Sogn", "Vestgrensa", "Kringstjø"};
```

Oppgave 6 (Utfordring):

I forelesningen (uke 11) ble det forklart et konsept, her skulle man finne det minste tallet i et int-array, ved hjelp av en `CountDownLatch` og en monitor. Kort sagt skal man:

- A. Lag en monitor som kan gjøre to ting:
 - a. Returnerer det minste tallet
 - b. Ta i mot et nytt tall fra en tråd og sjekke om dette tallet er mindre enn de andre tallene som har kommet inn til nå.
- B. Lag en tråd som tar inn en `startIndeks` og en `sluttIndeks`. Denne tråden skal finne det minste tallet fra `startIndeks` til `sluttIndeks` og sende dette til monitoren.
- C. Når alle trådene er ferdig skal monitoren returnere det minste tallet.

D. Skriv et hovedprogram som starter opp tråden og monitoren.

TIPS:

Når du skal dele opp arrayet:

- Finn hvor mange elementer hver tråd skal jobbe på: `array.length/antall tråder`.
- Ha en `startIndeks` og en `sluttIndeks`.
- Bruk en `for-loop` til å starte trådene. Husk at du må oppdatere `startIndeks` og `sluttIndeks` for hver iterasjon.
- Når du skal starte den siste tråden send alltid inn `array.length` som `sluttIndeks`.

Hvis ferdig, eller du vil trene mer på lengre oppgaver, er tråd-obligene fra våren 2018 et fint sted å starte:

<https://www.uio.no/studier/emner/matnat/ifi/IN1010/v18/obligatoriske-innleveringer/oblig-6/index.html>