Pragmatisk programmering

Mer tråder

Plan for dagen

- Har folk begynt på obligen?
- Litt om join og barrierer
- Fullføre Tråd-programmet fra sist

Fun fact: Tråd-obligen

```
lisbethyang@eduroam-193-157-172-128 Innlevering 5 (oblig) % time java IkkeTråder.java ./Data/metadata.csv
java IkkeTråder.java ./Data/metadata.csv  33.32s user 0.54s system 110% cpu 30.731 total
lisbethyang@eduroam-193-157-172-128 Innlevering 5 (oblig) % time java KlargjørData.java ./Data/metadata.csv
java KlargjørData.java ./Data/metadata.csv  95.10s user 1.56s system 703% cpu 13.736 total
lisbethyang@eduroam-193-157-172-128 Innlevering 5 (oblig) %
```

$$CPU\ \% = \frac{user + system}{real} \times 100 \qquad \frac{95.10 + 1.56}{13.736} \times 100 \approx 703\ \%$$

join()

```java
class Main {
    public static void main(String[] args) {
        int antallTråder = 8;
        for (int i = 0; i < antallTråder; i++) {
            Thread t = new Thread(new RunKlasse());
            t.start();
            t.join();
        }
    }
}
```

```java
class Main {
    public static void main(String[] args) {
        int antallTråder = 8;
        Thread[] tråder = new Thread[antallTråder];

        for (int i = 0; i < antallTråder; i++) {
            tråder[i] = new Thread(new RunKlasse());
            tråder[i].start();
        }

        for (int i = 0; i < antallTråder; i++) {
            tråder[i].join();
        }
    }
}
```

Barrierer

- En annen måte å avslutte trådene på
- `CountDownLatch` er en barriere som teller ned hver gang en tråd har utført oppgaven sin
- Main-tråden venter til nedtellingen har nådd 0 før den fortsetter

```java
import java.util.concurrent.CountDownLatch;

class Oppgave implements Runnable {
    private Monitor monitor;
    private CountDownLatch barriere;

    public Oppgave(Monitor monitor, CountDownLatch barriere) {
        this.monitor = monitor;
        this.barriere = barriere;
    }

    @Override
    public void run() {
        try {
            while (monitor.antall() > 1) {
                int tall = monitor.taUt();
                barriere.countDown();
            }
        } catch (InterruptedException e) {
            System.out.println("Jeg ble avbrutt.");
            System.exit(1);
        }
    }
}
```