Pragmatisk programmering

Tråder

Tema for uken

Tråder

Hvorfor tråder?

- Hvis vi vil at to eller flere oppgaver skal kjøre samtidig
- Når flere tråder utfører forskjellige oppgaver, får vi gjort mer på samme tid

Tråder kan …

- Vente – `wait()`
- Sove – `sleep(`*`ms`*`)`
- Avbrytes – `interrupt()`
- Dele felles data
- Kommunisere med hverandre – i IN1010 skjer ikke dette direkte, men via delte objekter (monitorer)

Runnable

```
interface Runnable {
    void run();
}
```



`Runnable`-interfacet

```
class MinOppgave implements Runnable {
    @Override
    public void run() {
        for (int i = 0; i < 5000; i++) {
            System.out.println("Jeg har gjort oppgaven min " + i + " ganger!");
        }
    }
}
```

Kjøring av flere tråder

```java
class Main {
    public static void main(String[] args) {
        Runnable oppgave = new MinOppgave();
        Thread t1 = new Thread(oppgave);
        Thread t2 = new Thread(oppgave);
        t1.start(); t2.start();
    }
}
```

```
[lisbethyang@eduroam-193-157-182-238 IN1010 % javac *.java
[lisbethyang@eduroam-193-157-182-238 IN1010 % java Main.java
Jeg har gjort oppgaven min 0 ganger!
Jeg har gjort oppgaven min 1 ganger!
Jeg har gjort oppgaven min 0 ganger!
Jeg har gjort oppgaven min 1 ganger!
Jeg har gjort oppgaven min 2 ganger!
Jeg har gjort oppgaven min 2 ganger!
Jeg har gjort oppgaven min 3 ganger!
Jeg har gjort oppgaven min 4 ganger!
Jeg har gjort oppgaven min 5 ganger!
Jeg har gjort oppgaven min 6 ganger!
Jeg har gjort oppgaven min 3 ganger!
Jeg har gjort oppgaven min 4 ganger!
Jeg har gjort oppgaven min 5 ganger!
Jeg har gjort oppgaven min 6 ganger!
Jeg har gjort oppgaven min 7 ganger!
Jeg har gjort oppgaven min 8 ganger!
Jeg har gjort oppgaven min 7 ganger!
Jeg har gjort oppgaven min 9 ganger!
Jeg har gjort oppgaven min 10 ganger!
Jeg har gjort oppgaven min 11 ganger!
Jeg har gjort oppgaven min 8 ganger!
```

Utskriften vil variere ved ulike kjøringer

Synkronisering

- Dersom to eller flere tråder deler data, kan vi få uheldige resultater om vi ikke lar bare én tråd utføre endringer på dataene av gangen

```
class MinOppgave implements Runnable {
    private final int MAX = 100000;
    private int teller = 0;

    @Override
    public void run() {
        System.out.println("Begynner å telle! Felles teller: " + teller);
        for (int i = 0; i < MAX; i++) {
            teller++;
        }
        System.out.println("Ferdig! Felles teller: " + teller);
    }
}
```

Synkronisering

```
Last login: Thu Mar 26 15:01:07 on ttys000
lisbethyang@eduroam-193-157-182-238 IN1010 % javac *.java
lisbethyang@eduroam-193-157-182-238 IN1010 % java Main.java
Begynner å telle! Felles teller: 0
Begynner å telle! Felles teller: 0
Ferdig! Felles teller: 10000
Ferdig! Felles teller: 20000
lisbethyang@eduroam-193-157-182-238 IN1010 % javac *.java
lisbethyang@eduroam-193-157-182-238 IN1010 % java Main.java
Begynner å telle! Felles teller: 0
Begynner å telle! Felles teller: 0
Ferdig! Felles teller: 112482
Ferdig! Felles teller: 109341
lisbethyang@eduroam-193-157-182-238 IN1010 %
```

Noen ganger kan det gå bra uansett

Men med flere iterasjoner av oppgaven blir det vanskeligere

Låser

- Løsning på problemet med synkronisering
- Kun én tråd kan aksessere et delt objekt på en gitt tid

```java
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;

class MinOppgave implements Runnable {
    private final Lock lås = new ReentrantLock();
    private final int MAX = 100000;
    private int teller = 0;

    @Override
    public void run() {
        System.out.println("Begynner å telle! Felles teller: " + hentTeller());
        for (int i = 0; i < MAX; i++) {
            lås.lock();
            try {
                teller++;
            } finally {
                lås.unlock();
            }
        }
        System.out.println("Ferdig! Felles teller: " + hentTeller());
    }

    private int hentTeller() {
        lås.lock();
        try {
            return teller;
        } finally {
            lås.unlock();
        }
    }
}
```

```
lisbethyang@eduroam-193-157-182-238 IN1010 % javac *.java
lisbethyang@eduroam-193-157-182-238 IN1010 % java Main.java
Begynner å telle! Felles teller: 0
Begynner å telle! Felles teller: 0
Ferdig! Felles teller: 198162
Ferdig! Felles teller: 200000
lisbethyang@eduroam-193-157-182-238 IN1010 %
```