TRÅDER

Gruppe 9:)

ragnhinw@uio.no

Runnable og Thread

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

```
class Thread{
        void start();
        void interrupt();
        static void sleep(Int mil);
        [...]
}
```

Monitor

- Kritiske regioner: mekanismer som hindrer feil ved tråder. Der man samler flere instruksjoner til en ikke-avbrytbar sekvens.
- Et objekt der metodene er kritiske regioner er en monitor
- En monitor er et trådsikkert objekt.

Monitor

- For å gjøre en metode til en kritisk region, en ikke-avbrytbar sekvens, bruker vi en lås.
- Trenger klassen ReentrantLock som implementerer interfacet Lock
- Eksempel på lock og unlock
- Og try – finally. Finally kjører uansett hva som skjer i try-blokken.

Venting - Conditions

- Aktiv vs passiv venting, vi foretrekker passiv -> conditions er løsningen.
- Lås-objekter tilbyr en metode newCondition()
- condition.await() og .signalAll()

Oppsummering

- En tråd opprettes med et runnableObjekt som argument til konstruktør.
- Tråd.start() setter i gang koden i runnableObjektet sin run()-metode.
- For å forhindre at flere tråder prøver å aksessere samme data samtidig bruker vi en et trådsikkert objekt (en monitor).
- En monitor er et objekt der alle metodene er trådsikre. Det blir de ved at de alle er ikke-avbrytbare. Det får man til vedå bruke en ReentrantLock.
- Om noen operasjoner er avhengige av en viss betingelse kan man lage en condition med lås: lås.newCondition().
- Betingelse.await() gjør at tråden ikke går videre med koden sin før en annen trår har gjort betingelse.signalAll().