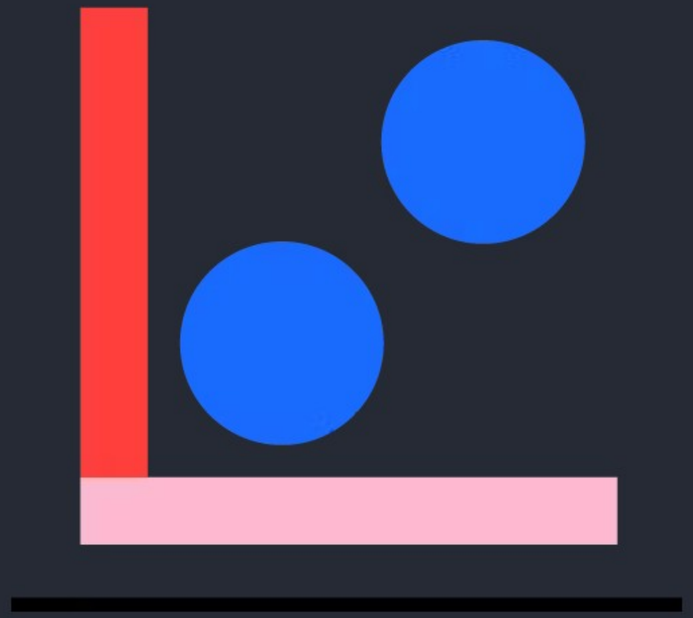
Mentimeter

Tråder del 1

IN1010 gruppetime 11

Gruppelærer: Sigmund Trutt



Mentimeter

Instructions

Go to

www.menti.com

Enter the code

6410 1877



Or use QR code

Mentimeter

Hva er en tråd

- En tråd er en sekvens av intstruksjoner i et program
- Hittil har vi skrevet vanlig, ikke parallelle programmer. Disse har hatt én sekvens av instruksjoner, atså én tråd.
- Hvis vi ønsker at to eller flere oppgaver (sett med instrukser) skal kjøre samtidig, kan vi lage flere tråder.
- Ulike tråder kan dele opp objekter (minne), og "snakke" med hverandre.
- I IN1010 skal trådene ikke kommunisere direkte, kun bruke delte objekter (monitorer)

Mentimeter

Hvorfor trenger vi tråder?

- Gjør programmet mer effektivt
- Dersom vi har mye data er det mer effektivt å ha flere tråder som jobber samtidig
- F. eks taco for 15 personer går fortere å lage om alle hjelper til med forskjellige oppgaver

Mentimeter

Thread = Worker, Runnable = task

- Vi kan tenke på en tråd som en arbeider - noe som kan utføre et sett med instrukser, en oppgave.
- Oppgavene som trådene kan utføre er objekter av grensesnittet Runnable.

Mentimeter

Runnable

→ Vi beskriver oppgaver som kan utføres ved objekter av grensesnittet Runnable

Mentimeter

Thread

- Vi lager nye arbeidere ved å lage et objekt av klassen Thread. Vi kan så gi tråden en oppgave den kan utføre, og så be den starte.
- Vi starter opp tråder med .start()
- En tråd kan sove ved å si .sleep(), men dette kan kaste InterruptedException
- Venter på andre tråder: .join()

Mentimeter

Synkronisering

- Dersom to tråder har tilgang påp delt data, kan vi få uheldige resultater dersom vi ikke passer på.
- Problemet er at trådene forsøker å oppdatere samme verdi samtidig
- Dette kalles en race condition, og er (kanskje) den største kilden til feil i parallelle programmer

Mentimeter

Løsning: Kritisk region

- Vi løser problemet ved at kun en tråd kan aksessere et delt objekt til en gitt tid
- Vi lager en såkalt mutex, en kritisk region hvor kun en tråd er om gangen.
- Andre tråder må stoppe og vente på sin tur.

Mentimeter

Locks

- Vi lager kritiske regioner med låser. Tenk at du har dører på hver side av koden du vil beskytte.
- Hvis trådene låser døren når de går inn må de andre trådene vente til tråden som låste, låser opp igjen
- Vi bruker java.util.concurrent.locks.Lock, et grensesnitt som spesifiserer blant annet to metoder, lock() og unlock()
- Den faktiske klassen vi lager låseobjekter av heter java.util.concurrent.locks.ReentrantLock