Tråder – del II

Gruppe 11 + 7

Hva skjer?

-  : gruppetimer
  lab
  forelesning

- Ordstyrer: Sindre Ask Vestaberg – sindreve@uio.no
- Seminarholder: Annika Willoch Olstad – annikaol@uio.no

- Spørsmål? → Skriv i chat
  → «Raise hand» → still det muntlig
- «Slower»/«faster»
- Vil du være anonym? → rename

Plan

- **Uke 11: Tråder – del II**
- Uke 12: Design og testbarhet
- Uke 13: Påske 🐣
- Uke 14: GUI – del I
- Uke 15: GUI – del II

Hva er tråder?

(og hva skal vi med det?)

Hvordan lager man en tråd?

1. class KlassenVaar implements Runnable
2. class KlassenVaar extends Thread

Runnable = oppgaver

```
class OppgaveSomSkalGjores implements Runnable{
    public void run() {
        System.out.println("Hello”);
    }
}
```

Thread = arbeider

```
public Main {
    public static void main(String[] args) {
        Runnable oppgave = new OppgaveSomSkalGjores();
        Thread arbeider1 = new Thread(oppgave);
        Thread arbeider2 = new Thread(oppgave);
        // kaller på oppgave.run()
        arbeider1.start();
        arbeider2.start();
    }
}
```

Hvordan lager man en tråd?

1. Opprett oppgaveklassen som implementerer Runnable
2. Implementer run()-metoden i oppgaveklassen
3. Lag et nytt Thread-objekt der parameteret er nytt objekt av oppgaveklassen
4. Kall på start()-metoden til Thread-objektet fra pkt.3.

Synkronisering

(hva gjør vi med delt data?)

```
MinMonitor monitor = new MinMonitor();
```

→ tilgang til data →

```
laas.lock();
```

```
laas.unlock();
```

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

class CountMonitor {
    private final Lock lock = new ReentrantLock();
    private int sharedCounter = 0; // The protected data.

    public void increment() {
        lock.lock();
        try {
            sharedCounter = sharedCounter + 1;
        } finally {
            lock.unlock();
        }
    }

    public int getCounter() {
         return sharedCounter;
    }
}
```

```java
class MyTask implements Runnable {
    private final int MAX_COUNT = 10000;
    private final CountMonitor monitor;

    public MyTask(CountMonitor monitor) {
         this.monitor = monitor;
    }

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

(likt hovedprogram)

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

Conditions

java.util.concurrent.locks.Condition

```
.await();
```

```
.signal();
```

```
.signalAll();
```

Barrierer

```
CountDownLatch(int count)
```

```
minBarriere.await();
minBarriere.countDown();
```

```
CyclicBarrier(int count)
```

```
minBarriere.await();
```

Eksempel 1

CountDownLatch

Eksempel 2

Fra uke 9: 

Vi skal lage et program med kaffedrikkere (tråd), en barista (tråd), og ett bord (monitor). Bordet kan ha uendelig mange kaffer på seg.

- Lag klassen Barista som er en tråd:
  - Alle har et array med ulike kaffedrikker

```
private final String[] drikker = {"Americano", "Café au lait", "Caffè latte", "Caffè mocca", "Espresso", "Cortado"};
```

  - Baristaen tar inn et Bord som vedkommende kan servere kaffen på
  - Baristaen tar inn en id.
- Implementer run()-metoden til barista:
  - Hver barista lager 10 kaffer, og disse drikkene skal velges tilfeldig fra liste drikker (hint: bruk java.util.Random for å få et tall mellom 0 og lengden til drikker).
  - Baristaen skal så skrive ut sin id og hvilken drikke som blir laget, før vedkommende serverer kaffen på bordet.
  - Når 10 kaffer er servert skal baristaen sende inn at det er “tomt”.
- Lag klassen Kaffedrikker som er en tråd:
  - Kaffedrikker skal ta inn et bord vedkommende kan hente kaffe fra
  - Kaffedrikker skal også ta inn en id.
- Implementer run()-metoden i Kaffedrikke:
  - Denne skal ha en teller som teller antall kaffer som kaffedrikkeren får drukket.
  - Så lenge kaffedrikkeren får beskjed om at det ikke er tomt skal vedkommende printe sin id og hvilken kaffe som ble drukket.
  - Når det ikke er flere kaffer for kaffedrikkeren å drikke skal vedkommende printe sin id og hvor mange kaffekopper som ble drukket.
- Lag klassen Bord:
  - Lag en metode som server kaffe (legger de til i bordet). Husk at det kan være uendelig mange kaffer på bordet om gang.
  - Lag også en metode hentKaffe som henter en kaffe fra bordet så lenge det er en kaffe der, og signaliserer til Kaffedrikker når det er tomt.
- Lag klassen Hovedprogram som lager ett Bord, to baristaer og ti kaffedrikkere. Test gjerne med ulike verdier for å se hvordan programmet oppfører seg)

Enum

«An *enum type* is a special data type that enables for a variable to be a set of predefined constants.»
https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html

```
public enum Dag {
    MANDAG, TIRSDAG, ONSDAG, TORSDAG, FREDAG, LORDAG, SONDAG
}
```

```
public class EnumTest {
    Dag dag;

    public EnumTest(Dag dag) {
        this.day = day;
    }

    public void tellItLikeItIs() {
            if (day == Dag.MANDAG){
                System.out.println("Mondays are bad.");
             }
            else if (dag == Dag.FREDAG){
                System.out.println("Fridays are better.");
            }
            else if (dag == Dag.LORDAG || dag == Dag.SONDAG){
                System.out.println("Weekends are best.");
            }
            else {
                System.out.println("Midweek days are so-so.");
            }
        }
    }

    public static void main(String[] args) {
        EnumTest firstDay = new EnumTest(Dag.MANDAG);
        firstDay.tellItLikeItIs();
        EnumTest thirdDay = new EnumTest(Dag.ONSDAG);
        thirdDay.tellItLikeItIs();
        EnumTest fifthDay = new EnumTest(Dag.FREDAG);
        fifthDay.tellItLikeItIs();
        EnumTest sixthDay = new EnumTest(Dag.LORDAG);
        sixthDay.tellItLikeItIs();
        EnumTest seventhDay = new EnumTest(Dag.SONDAG);
        seventhDay.tellItLikeItIs();
    }
}
```

...send gjerne mail med tilbakemeldinger, kommentarer, spørsmål, etc.!