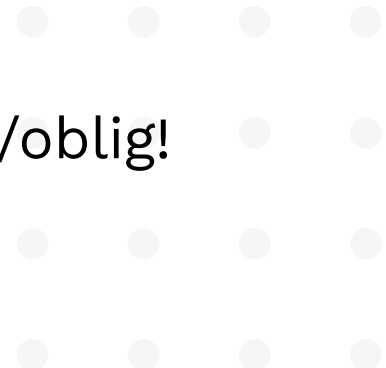




# IN1010 uke 10

Gruppe 7

# Agenda

- Litt repetisjon av tråder/monitor
  - Vente på at tråder skal bli ferdige
    - `.join()`
    - Barrierer
      - `CountDownLatch`
      - `CyclicBarrier`
  - Jobbe med oppgaver/trix/oblig!
- 

# Tråder og monitor



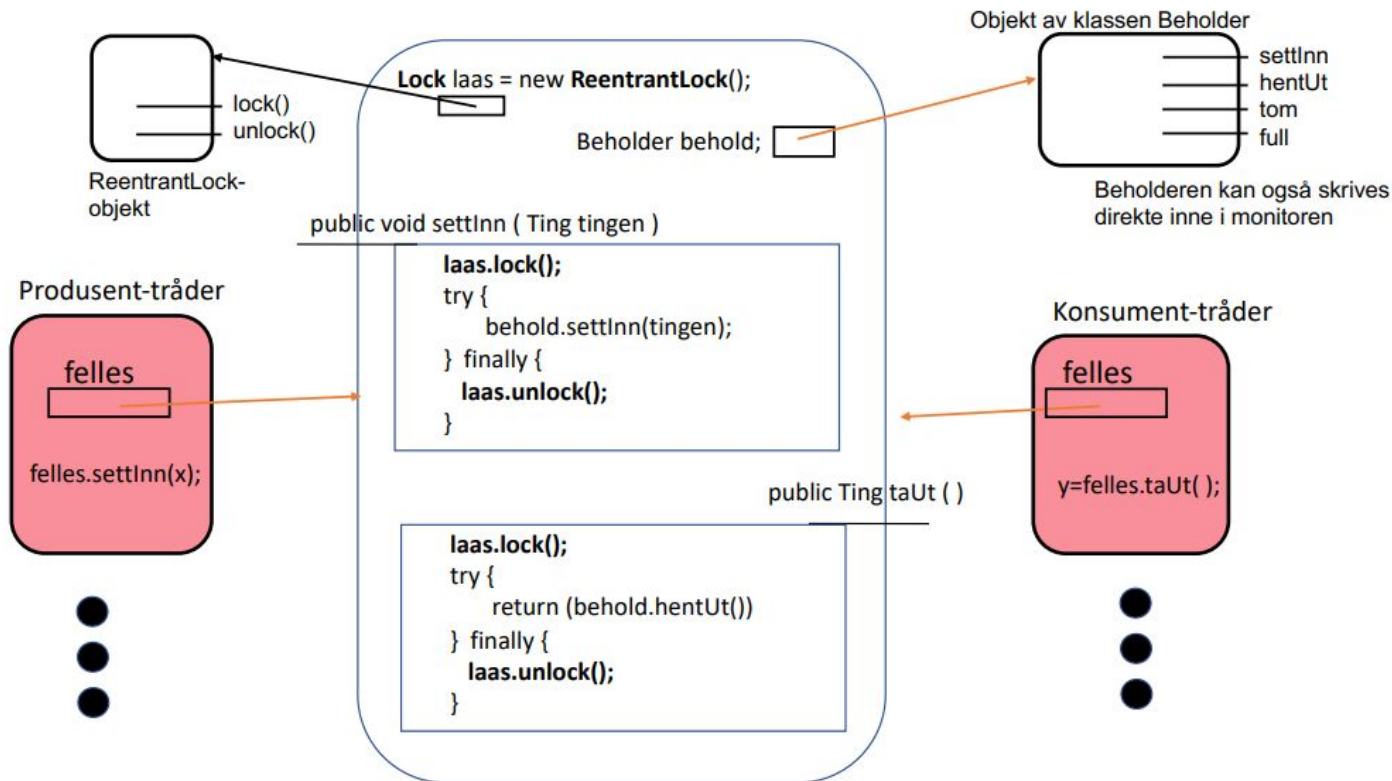
# Kritiske regioner

- Lager tråder for å kunne gjøre ting samtidig
    - Endre på samme ressurs samtidig? F.eks. oppdatere instansvariabel → Problemer  
= kritiske regioner
- Må beskytte kritiske regioner med lås
- bare en tråd av gangen kan aksessere

# Monitor

- Lager Monitor-klasse for å beskytte den delte dataen
- Har en klasse som vil bruke med tråder → lager en Monitor-klasse “rundt” den

# Monitor



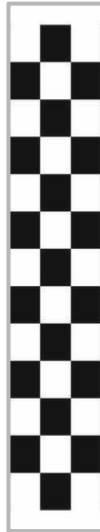
# Vente på at tråder skal bli ferdige



# Vente på tråder

- 2 måter å gjøre det på:
  - med `.join()`
  - med en barriere

*Bil-tråder*



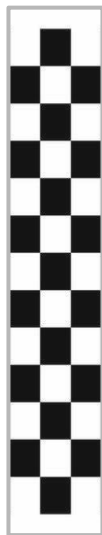
*Main-tråden*



# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



*Main-tråden*

```
bil1.start();  
bil2.start();  
bil3.start();
```

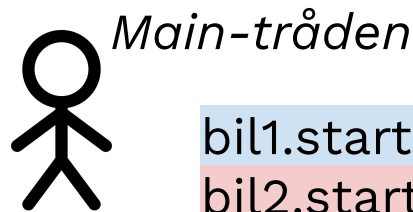
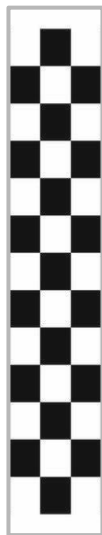
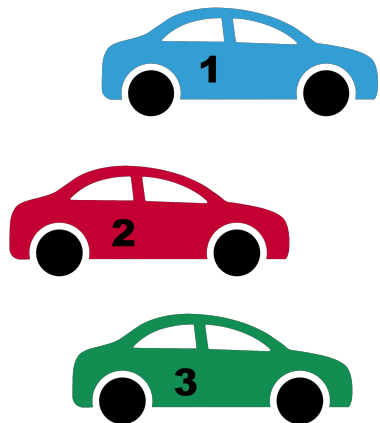
```
bil1.join();  
bil2.join();  
bil3.join();
```

```
System.out.println("Race  
ferdig!");
```

# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



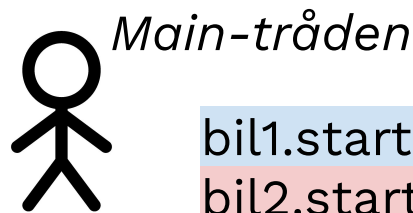
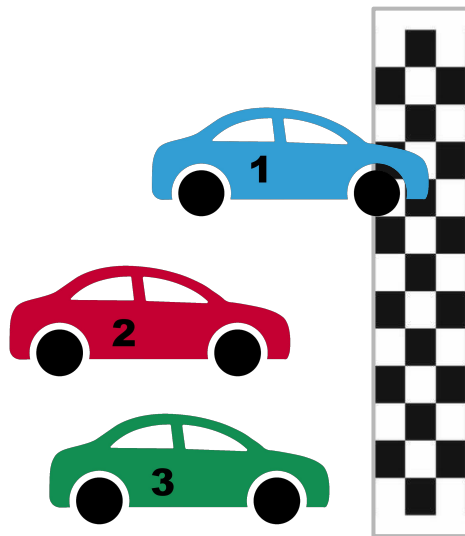
```
bil1.start();  
bil2.start();  
bil3.start();
```

```
→ bil1.join();  
   bil2.join();  
   bil3.join();  
   System.out.println("Race  
   ferdig!");
```

# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



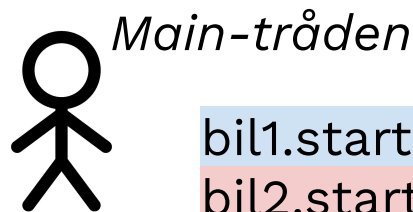
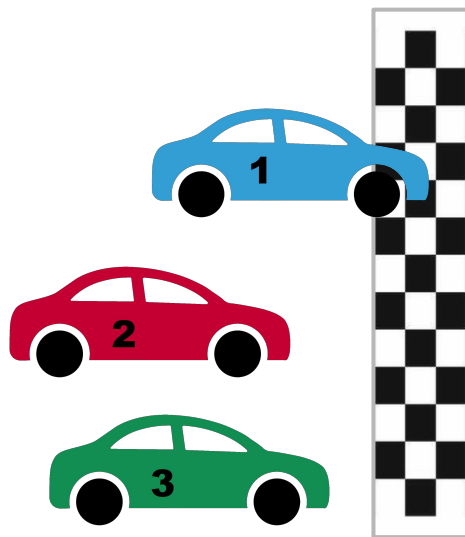
```
bil1.start();  
bil2.start();  
bil3.start();
```

```
→ bil1.join();  
   bil2.join();  
   bil3.join();  
   System.out.println("Race  
   ferdig!");
```

# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



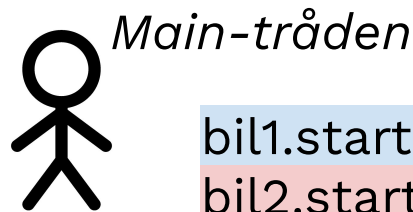
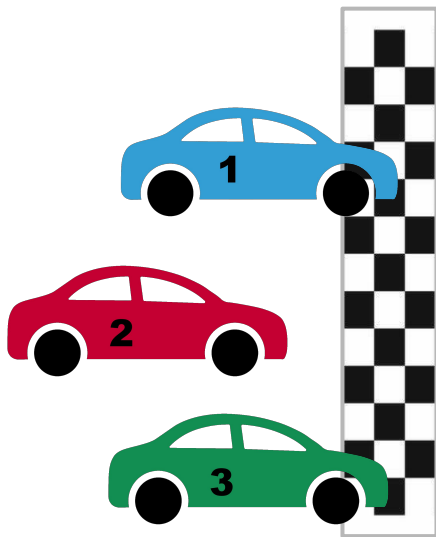
```
bil1.start();  
bil2.start();  
bil3.start();
```

```
bil1.join();  
bil2.join();  
bil3.join();  
System.out.println("Race  
ferdig!");
```

# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



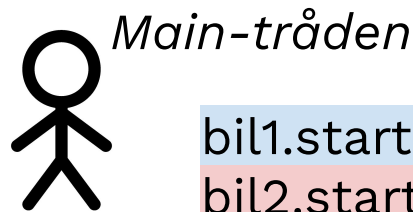
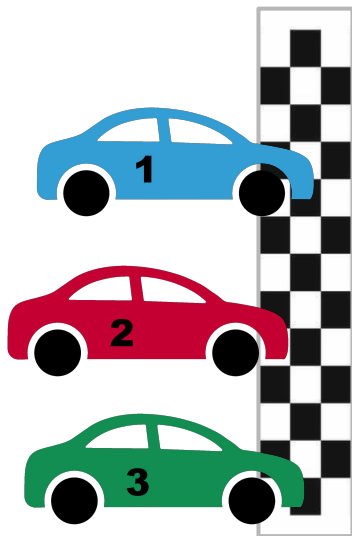
```
bil1.start();  
bil2.start();  
bil3.start();
```

```
bil1.join();  
bil2.join();  
bil3.join();  
System.out.println("Race  
ferdig!");
```

# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



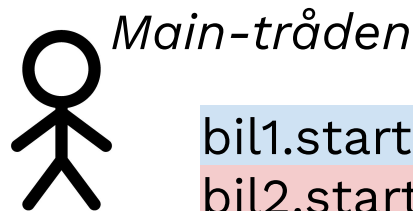
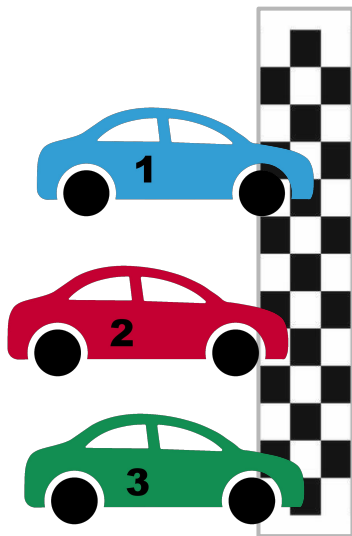
```
bil1.start();  
bil2.start();  
bil3.start();
```

```
bil1.join();  
bil2.join();  
bil3.join();  
System.out.println("Race  
ferdig!");
```

# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



```
bil1.start();  
bil2.start();  
bil3.start();
```

```
bil1.join();  
bil2.join();  
bil3.join();
```

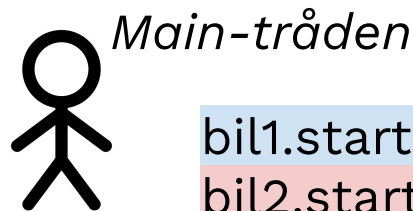
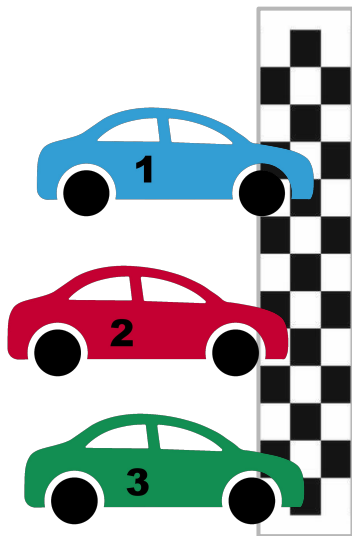


```
System.out.println("Race  
ferdig!");
```

# Join

- `traad.join()` → venter på at traad skal avslutte før går videre

*Bil-tråder*



```
bil1.start();  
bil2.start();  
bil3.start();
```

```
bil1.join();  
bil2.join();  
bil3.join();
```

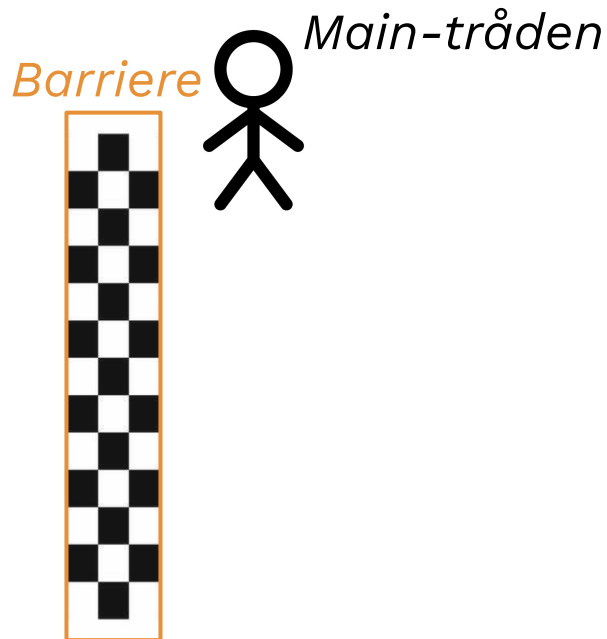
```
System.out.println("Race  
ferdig!");
```



# Barriere: CountdownLatch

- Når en tråd er ferdig teller den ned med `.countDown()`
- Kan vente på at alle tellet ned med `.await()`

*Bil-tråder*



# Barriere: CountdownLatch

countdown:

3

*Bil-tråder*



*Barriere*



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3);
```

# Barriere: CountdownLatch

countdown:

3

*Bil-tråder*



```
=new Thread(new BilTraad(barriere))
```

*Barriere*



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3);
```

# Barriere: CountdownLatch

countdown:

3

*Bil-tråder*



*Barriere*



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3);
```

# Barriere: CountdownLatch

countdown:

3

*Bil-tråder*



*Barriere*



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3);
```

```
bil1.start();  
bil2.start();  
bil3.start();
```

```
barriere.await();  
System.out.println("Race  
ferdig!");
```

# Barriere: CountdownLatch

countdown:

3

*Bil-tråder*



*Barriere*



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3);
```

```
bil1.start();  
bil2.start();  
bil3.start();
```



```
barriere.await();  
System.out.println("Race  
ferdig!");
```

# Barriere: CountdownLatch

countdown:

3

Barriere



Main-tråden

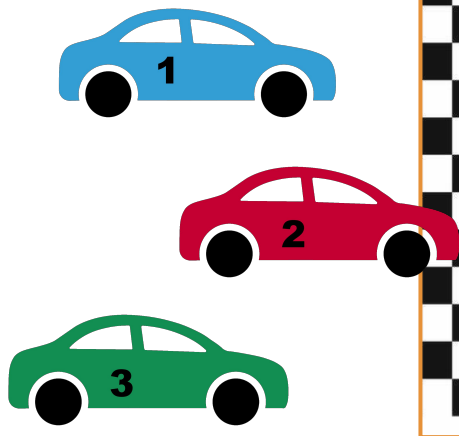
```
CountDownLatch barriere=  
new CountdownLatch(3)
```

```
bil1.start()  
bil2.start()  
bil3.start()
```

→ 

```
barriere.await()  
System.out.println("Race  
ferdig!")
```

Bil-tråder



# Barriere: CountdownLatch

countdown:

2

*Bil-tråder*



*barriere.countDown();*



*Barriere*



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3)
```

```
bil1.start()  
bil2.start()  
bil3.start()
```

```
→ barriere.await()  
System.out.println("Race  
ferdig!")
```

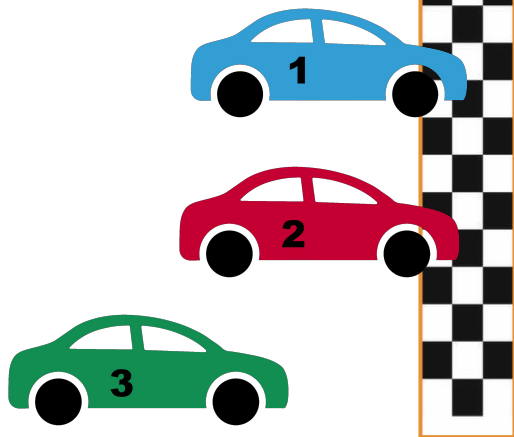


# Barriere: CountdownLatch

countdown:

2

*Bil-tråder*



*Main-tråden*

```
CountDownLatch barriere=  
new CountdownLatch(3)
```

```
bil1.start()  
bil2.start()  
bil3.start()
```

→ 

```
barriere.await()  
System.out.println("Race  
ferdig!")
```

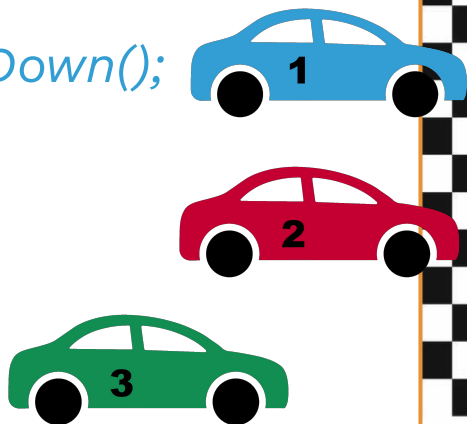
# Barriere: CountdownLatch

countdown:

1

*Bil-tråder*

`barriere.countDown();`



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3)
```

```
bil1.start()  
bil2.start()  
bil3.start()
```

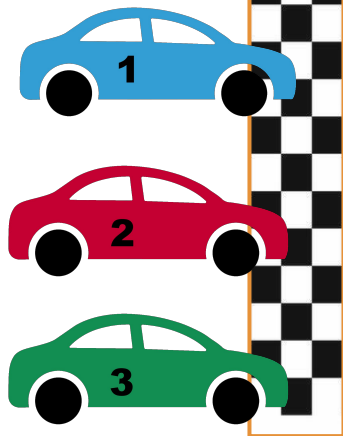
→ `barriere.await()`  
`System.out.println("Race  
ferdig!")`

# Barriere: CountdownLatch

countdown:

1

*Bil-tråder*



*Main-tråden*

```
CountDownLatch barriere=  
new CountdownLatch(3)
```

```
bil1.start()  
bil2.start()  
bil3.start()
```

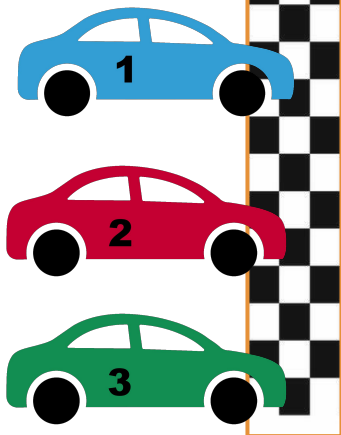
```
→ barriere.await()  
System.out.println("Race  
ferdig!")
```

# Barriere: CountdownLatch

countdown:

0

*Bil-tråder*



*barriere.countDown();*



*Main-tråden*

```
CountDownLatch barriere=  
new CountDownLatch(3)
```

```
bil1.start()  
bil2.start()  
bil3.start()
```

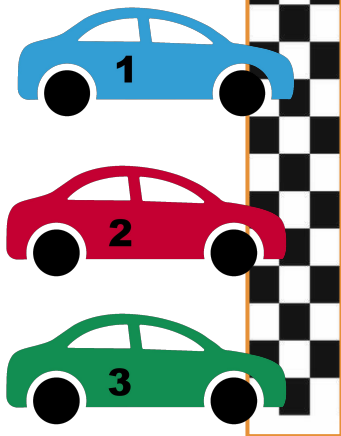
```
→ barriere.await()  
System.out.println("Race  
ferdig!")
```

# Barriere: CountdownLatch

countdown:

0

*Bil-tråder*



*barriere.countDown();*



*Main-tråden*

```
CountDownLatch barriere=  
new CountdownLatch(3)
```

```
bil1.start()  
bil2.start()  
bil3.start()
```

```
barriere.await()
```

```
→ System.out.println("Race  
ferdig!")
```

# Barriere: CyclicBarrier

- Eksempel: CyclicBarrier(3)
  - Hver tråd kaller på `.await()` når ferdig for å vente på de andre
  - Når 3 tråder har kalt `.await()` slipper alle gjennom
- Forskjell fra CountdownLatch:
  - Samme CyclicBarrier kan brukes flere ganger
  - Slipper gjennom automatisk når alle nådd barrieren
- Holder med å kunne én av dem!

# Kodeeksempel

*CountDownLatch,  
monitor*



# Jobb med oppgaver!

IN1010 Emnesiden → Grupper →  
Gruppe 7 → Uke10

