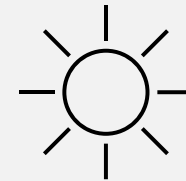


GRUPPE 8 INI010 – UKE 8

Sivert Fjeldstad Madsen

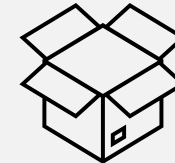
I DAG

- Repetisjon av iteratorer
- Repetisjons-Kahoot
- Oppgaver



ITERATOR

- Dersom en beholder vi designer *ikke* er **indeksert** er det ikke noen (enkel) måte å iterere gjennom den med en standard for-løkke
 - I hvert fall «utenfra»!
- Siden vi veldig ofte ønsker å kunne iterere gjennom innholdet i en beholder er det derfor som regel lurt å la beholderen implementere **Iterable<T>** - grensesnittet
- Dette gjør det enkelt for alle som bruker beholderen din å kunne iterere over den



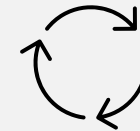
ITERABLE<T>

- Grensesnittet **Iterable<T>** krever at vi implementerer metoden **public Iterator<T> iterator()**, som skal returnere et Iterator - objekt
- Vi må da opprette en indre klasse som implementerer grensesnittet **Iterator<T>**
- Deretter kan vi la `iterator()`-metoden returnere nye instanser av den indre klassen



ITERATOR<T>

- Den indre iteratorklassen må implementere metodene
 - `public boolean hasNext()`
 - `public T next()`
- Klassen må altså holde styr på **hvor langt i iterasjonen** den har kommet
 - Dette kan implementeres akkurat hvordan du vil (så lenge det fungerer korrekt)



HVORFOR EN INDRE KLASSE?

- Kunne ikke egentlig beholderen vår bare implementert **Iterator<T>** direkte? Hvorfor må vi ha en indre klasse?
- Tenk deg at det var gyldig å kode det sånn.
 - Hva hadde skjedd i følgende tilfelle?



```
for (Element a : beholder) {  
    for (Element b : beholder) {  
        System.out.println(a + b);  
    }  
}
```

ITERATOR - KONKLUSJON

- Siden vi ønsker å begynne fra starten av beholderen hver gang vi skal iterere over den, er vi nødt til å lage et nytt iteratorobjekt for hver enkelt løkke
- Dette er grunnen til at vi trenger den indre iteratorklassen
- Det er verdt det å gjøre alt dette!
 - Det er utrolig nyttig å kunne iterere gjennom en beholder.



OPPSUMMERING

- Vi er nå halvveis i kurset!
- **Pass på at pensum frem til nå bør sitte godt i løpet av uken**
- You can do it!



- Ukas tips:
 - `switch (uttrykk) {`
 - `case verdi:`
 - `...`
 - `break;`
 - `default:`
 - `}`