

Iterator mallin toiminta tehtävä 20

Kokoelman iteroiminen kahdella säikeellä yhtä aikaan, kun molemmilla on oma iteraattori

Rakensin ohjelman joka toteuttaa edellä mainitun tilanteen ja tuloksena tuli, että kokoelma iteroitiin ainoastaan kerran.

Sama tilanne muuten, mutta säikeet käyttävät samaa iteraattoria vuorotellen

Tuloksena jokainen asia kokoelmasta iteroitiin kaksi kertaa

Kokoelman muuttaminen iteroimisen aikana

Iteraattorilla ei ole hyvää tapaa pitää merkillä missä kohdassa se on. Tähän esim on että, Iteraattori olisi vaikka kokoelman kohdassa viisi, ja poistettaisiin kokoelman kolmas kohta, niin seuraavan kerran kun kutsuttaisiin next() niin Iteraattori ei tietäisi, että mennäänkö me kohtaan kuusi? vai ollaanko me kohdassa viisi? Joten ohjelma heittäisi sitten poikkeuksen.

Jos kokoelmaa haluttaisiin muuttaa iteroimisen aikana

Kuten edellisessä testauksessa tuli ilmi, niin ongelma kokoelman muuttamisessa on se että Iteraattori ei tiedä missä kohdassa se on muuttamisen jälkeen. Kuitenkin testailtuani, kokoelmaa pystyy muuttamaan ajon aikana, jos käsitellään iteraattorin kohdalla olevaa kokoelman jäsentä, tämä ei siis sekoita Iteraattorin kyseistä kohtaa esim. tässä:

```
18 public void AddingColors() {
19
20     colorlist.add("valkoinen");
21     colorlist.add("punainen");
22     colorlist.add("sininen");
23     colorlist.add("musta");
24     colorlist.add("oranssi");
25     colorlist.add("ruskea");
26 }
27
28 synchronized void iterate() {
29
30     Iterator<String> it = colorlist.iterator();
31
32     while(it.hasNext()) {
33
34         if (it.next().equalsIgnoreCase("musta")) {
35             it.remove();
36         }
37
38     }
39     System.out.println(colorlist);
40 }
41
42
```

<

Console x Debug Project Explorer

terminated> Main (14) [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (21.11.2022 klo

[valkoinen, punainen, sininen, oranssi, ruskea]

Jokaisen kohdan testaukset löytyy:

<https://github.com/rolle324/SuunnittelumallitTehtavat/tree/master/IteratorTestingTeht20>