

UF2.NF1.Processos: concurrència

Monitors:

1.- Utilitza la classe Monitor.java que trobaràs al Moodle per implementar dos fils que s'alternin en l'accés al Monitor de lectura. Deixa dos segons d'espera entre obertura i tancament.

```
concurrència.FilX obert Monitor per lectura
concurrència.FilX tancat Monitor per lectura
concurrència.FilY obert Monitor per lectura
concurrència.FilY tancat Monitor per lectura
```

2.- Modifica l'exercici anterior per accedir també al Monitor d'escriptura. Al segon fil, fes que esperi 5 segons entre tancament de lectura i obertura d'escriptura.

3.- Fes una classe BD amb un mètode que es connecti utilitzant jdbc a una taula de mysql, esperi 3 segons, i mostri els resultats dels registres per pantalla. Assegura't que el ResultSet es tanca al final.

4.- Passant la mateixa instància de la classe BD a dos fils, executa, per cada fil, 10 vegades el mètode de l'exercici anterior. S'executen correctament totes les iteracions?

5.- Modifica l'exercici anterior per aplicar-hi un Monitor.

Semàfors:

6.- Realitza l'exercici 4 amb Semàfors (java.util.concurrent.Semaphore). Mostra els permisos disponibles del semàfor després de cada acquire/release:

```
concurrència.FilX permisos disponibles del semàfor: 0
Connexió realitzada usant DriverManager

  id descriptio  data_inici  data_final finalitzada
-----
  1  Comprar pomes 15 de febr. 2017  5 de juny 2015  true
-----

concurrència.FilX permisos disponibles del semàfor: 1
concurrència.FilY permisos disponibles del semàfor: 0
Connexió realitzada usant DriverManager
```

7.- Modifica l'exercici dels N filòsofs fet a la pràctica anterior per complir amb el següent requisit: (Pots utilitzar l'estructura que vulguis)

‘Normalment, un filòsof sap molt bé com pensar i menjar però les operacions d'agafar les forquilles i deixar-les s'han de fer de tal manera que cap filòsof mori de fam (o mori per sobrepès).’

Nota: prioritats i/o cua FIFO

8.- Llegeix l'enunciat següent:

‘El Pare Noel viu a la seva caseta de Jyëvväskjølgrø i la seva activitat preferida és fer migdiades. Dorm tot el temps, excepte quan el desperta algun grup d'elfs o un grup de rens. Hi ha 10 elfs que viuen a la zona i treballen en un taller de joguines. Produeixen joguines i, de tant en tant, es queden sense idees pel que han de demanar ajuda al Pare Noel. Només poden despertar el Pare Noel si formen un grup de 3 elfs.

Per altre banda, també hi ha 9 rens que viuen a l'estable; El Pare Noel els fa lliurar joguines als nens. Els rens són tan mandrosos com el mateix Pare Noel, així que la majoria del dia només dormen. Quan volen treballar, han de formar un grup de 9 (és a dir, tots els rens s'han d'unir) i despertar el Pare Noel. Si ho aconsegueixen, el Pare Noel reparteix els regals i després torna a dormir (els rens també).

És important tenir en compte que en una situació en què hi ha un grup de 3 elfs i el grup de 9 rens intentant despertar al Pare Noel al mateix temps, són els rens els que tenen prioritat.’



Un cop llegit, clona el següent repositori i respon/fes les modificacions proposades a continuació: <https://github.com/GeorgeKirm/Santa-Claus-Concurrency-in-Java>

8.1.- Executa el programa i després mira't i entén el codi. Quins objectes utilitza per assegurar l'exclusió mútua? Busca informació de com funcionen.

8.2.- Treu momentàniament el lock i condició de bloqueig quan afegeixes un elf al grup. Què passa si executes?

8.3.- Modifica el codi per utilitzar un semàfor amb comptador en comptes d'un Lock a l'afegir un elf al grup.

9.- (Opcional) Modifica la classe Monitor.java per permetre que varis fils puguin llegir el monitor de lectura a la vegada però que l'escriptura només pugui un. Fes-ne la prova amb dos fils.