

Activitat 03: Sleep i Coet

Introducció

Continuem practicant l'ús de Sleep en una activitat específica i després implementem una simulació de motors d'un coet espacial de forma simplista.

Enunciat

Part 1: Sleep

Crea la classe DormAleatori.java que:

- S'inicialitzi amb un nom però utilitzi la variable name de Thread
- Tingui constància de l'instant en que s'ha creat (no ficat en execució, sinó ha estat construïda → constructor)
- En l'execució repeteixi 10 vegades:
 - Mostrar → nom, num_iteració, [interval_aleatori](#), total de temps des de la construcció
 - Dormir l'[interval_aleatori](#) mostrat
- Tingui un mètode principal on:
 - Es creen dues instàncies de DormAleatori una anomenada «Joan» i l'altra «Pep» i
 - Les fiqui en marxa.
 - Mostri que s'acaba el mètode

Exemple de sortida esperada:

```
Joan(0) a dormir    74ms total      1
Pep (0) a dormir    877ms total     1
-- Fi de main -----
Joan(1) a dormir    526ms total     139
Joan(2) a dormir    214ms total     666
Joan(3) a dormir    997ms total     881
Pep (1) a dormir    467ms total     942
Pep (2) a dormir    523ms total    1410
Joan(4) a dormir    286ms total    1879
Pep (3) a dormir    932ms total    1935
Joan(5) a dormir    218ms total    2166
Joan(6) a dormir    143ms total    2385
Joan(7) a dormir    177ms total    2529
Joan(8) a dormir    360ms total    2708
```

Pep (4) a dormir	590ms	total	2869
Joan (9) a dormir	398ms	total	3070
Pep (5) a dormir	72ms	total	3460
Pep (6) a dormir	976ms	total	3535
Pep (7) a dormir	918ms	total	4512
Pep (8) a dormir	114ms	total	5432
Pep (9) a dormir	22ms	total	5548

Part 2: Coet

Introducció

Per simular processos reals, es solen construir programes informàtics que han d'emular el comportament dels sistemes que es volen simular. En aquest cas simularem el comportament d'un coet espacial amb 4 motors.

No tindrem en compte cap tipus de variable física més que el temps de resposta dels motors que serà aleatori.

Enunciat

Crea la classe `Motor.java` que simuli el comportament d'un motor real de coet de la següent manera:

- Inicialment està aturat i per tant la potència objectiu i la potència actuals són 0
- Quan se li estableix la potència objectiu amb el setter `setPotencia(int p)` ha de passar de la potència actual a la objectiu en passos de 1 (incrementant o decrementant) i per simular el comportament d'un motor real trigarà en fer-ho un interval aleatori de entre 1 i 2 segons.
- Si la potència objectiu és 0 quan arribi a la potència 0 ha de sortir

Crea també la classe `Coet.java` que tingui 4 motors i en ella:

- Crea el mètode `passaAPotencia(int p)` que comprovi que la potència demanada és correcta (està entre 0 i 10) i estableixi la potencia dels 4 motors a p o doni un error de valor
- Crea el mètode `arranca` que llanci a funcionar els 4 motors
- Fes que `Coet` llegeixi contínuament per consola la potència objectiu i immediatament la estableixi a tots els seus motors fins que la potència objectiu sigui 0

Exemple de sortida:

3

Passant a potència 3

Motor 0: Incre. Objectiu: 3 Actual: 1

Motor 0: Incre. Objectiu: 3 Actual: 2

Motor 1: Incre. Objectiu: 3 Actual: 1
Motor 2: Incre. Objectiu: 3 Actual: 1
Motor 0: FerRes Objectiu: 3 Actual: 3
Motor 3: Incre. Objectiu: 3 Actual: 1
Motor 3: Incre. Objectiu: 3 Actual: 2
Motor 3: FerRes Objectiu: 3 Actual: 3
Motor 2: Incre. Objectiu: 3 Actual: 2
Motor 1: Incre. Objectiu: 3 Actual: 2
7Motor 1: FerRes Objectiu: 3 Actual: 3
Motor 2: FerRes Objectiu: 3 Actual: 3

Passant a potència 7

Motor 0: Incre. Objectiu: 7 Actual: 4
Motor 0: Incre. Objectiu: 7 Actual: 5
Motor 2: Incre. Objectiu: 7 Actual: 4
Motor 3: Incre. Objectiu: 7 Actual: 4
Motor 1: Incre. Objectiu: 7 Actual: 4
Motor 3: Incre. Objectiu: 7 Actual: 5
Motor 0: Incre. Objectiu: 7 Actual: 6
Motor 3: Incre. Objectiu: 7 Actual: 6
Motor 2: Incre. Objectiu: 7 Actual: 5
Motor 3: FerRes Objectiu: 7 Actual: 7
Motor 0: FerRes Objectiu: 7 Actual: 7
Motor 1: Incre. Objectiu: 7 Actual: 5
Motor 1: Incre. Objectiu: 7 Actual: 6
Motor 2: Incre. Objectiu: 7 Actual: 6
Motor 1: FerRes Objectiu: 7 Actual: 7

0

Passant a potència 0

Motor 2: Decre. Objectiu: 0 Actual: 7
Motor 1: Decre. Objectiu: 0 Actual: 6
Motor 0: Decre. Objectiu: 0 Actual: 6
Motor 3: Decre. Objectiu: 0 Actual: 6
Motor 1: Decre. Objectiu: 0 Actual: 5
Motor 3: Decre. Objectiu: 0 Actual: 5
Motor 2: Decre. Objectiu: 0 Actual: 6
Motor 3: Decre. Objectiu: 0 Actual: 4
Motor 1: Decre. Objectiu: 0 Actual: 4
Motor 1: Decre. Objectiu: 0 Actual: 3
Motor 0: Decre. Objectiu: 0 Actual: 5
Motor 3: Decre. Objectiu: 0 Actual: 3
Motor 2: Decre. Objectiu: 0 Actual: 5
Motor 2: Decre. Objectiu: 0 Actual: 4
Motor 2: Decre. Objectiu: 0 Actual: 3
Motor 2: Decre. Objectiu: 0 Actual: 2
Motor 3: Decre. Objectiu: 0 Actual: 2
Motor 0: Decre. Objectiu: 0 Actual: 4
Motor 1: Decre. Objectiu: 0 Actual: 2
Motor 2: Decre. Objectiu: 0 Actual: 1
Motor 3: Decre. Objectiu: 0 Actual: 1
Motor 0: Decre. Objectiu: 0 Actual: 3

Motor 2: FerRes Objectiu: 0 Actual: 0
Motor 1: Decre. Objectiu: 0 Actual: 1
Motor 3: FerRes Objectiu: 0 Actual: 0
Motor 1: FerRes Objectiu: 0 Actual: 0
Motor 0: Decre. Objectiu: 0 Actual: 2
Motor 0: Decre. Objectiu: 0 Actual: 1
Motor 0: FerRes Objectiu: 0 Actual: 0

Lliurament

Has de lliurar el link al teu repositori de la UF2 on dins hi haurà en arrel la carpeta 03-Sleep_i_Coet/src amb el codi font demanat.

Has de fer un commit al git al acabar la classe amb la feina feta a classe i un comentari que digui «03-Sleep_i_Coet fet a classe»

Després podràs fer més commits fins la data límit de lliurament, que obligatòriament hauran de tenir el nom de l'activitat i una breu descripció del que conté el commit.