



ACTIVITAT
Objectius: <ul style="list-style-type: none">- Aprendre a programar threads amb JAVA
Instruccions: <ul style="list-style-type: none">- Es tracta d'un treball en grups de dos, no s'admet cap tipus de còpia.- Responen a l'espai de cada pregunta, si ho feu amb diapositives enganxeu la diapositiva en aquest mateix espai.- Es valorarà la cura en la presentació del document i que segueixi l'estructura indicada.
Criteris d'avaluació: <ul style="list-style-type: none">- Cada pregunta té el mateix pes
Entrega: <ul style="list-style-type: none">- Aquest document amb les explicacions i captures necessàries i els arxius adjunts necessaris del codi que es demana- El nom dels arxius adjunts a entregar seràn: nomicognom-nomicognom.zip

Noms i Cognoms:

Materials:

Necessiteu un entorn de desenvolupament en JAVA (per exemple Eclipse)
Feu servir Google per buscar els tutorials que us serveixin millor
Tens més informació sobre el mètodes d'exclusió mútua en [aquest post](#)

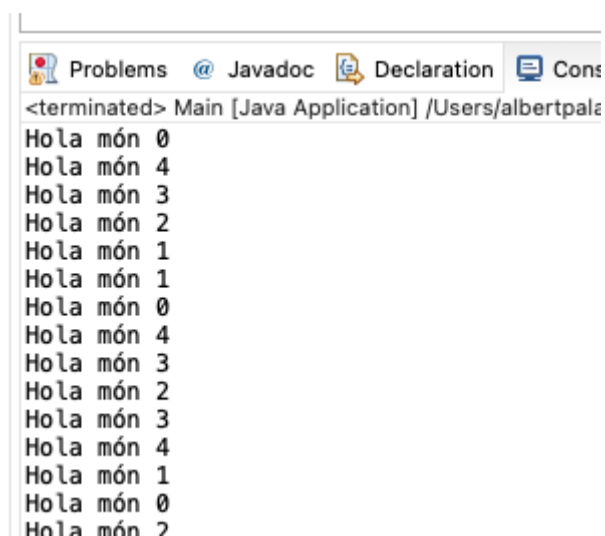


Tasques:

- **Exercici 0** - Implementa un programa en JAVA (PR210Threads.java) amb:

- 5 fils d'execució
- Que escrigui 'Hola món x' per cada fil d'execució on 'x' és el número de fil
- Cada fil espera 10 mil·lèsimes de segon abans de tornar a escriure
- El programa s'executa durant 2 segons, i es paren els fils des del main

La sortida ha de ser una llista desordenada (o més o menys ordenada) de les escriptures dels fils, tipus:



```
<terminated> Main [Java Application] /Users/albertpala  
Hola món 0  
Hola món 4  
Hola món 3  
Hola món 2  
Hola món 1  
Hola món 1  
Hola món 0  
Hola món 4  
Hola món 3  
Hola món 2  
Hola món 3  
Hola món 4  
Hola món 1  
Hola món 0  
Hola món 2
```

- **Exercici 1** - Implementa un programa en JAVA (PR211Threads.java) amb:

- 4 fils d'execució
- Cada fil incrementa un contador enter compartit per tots els fils 5.000 vegades
- El programa principal només inicia la variable compartida i la mostra al final
- Explica perquè el resultat cada vegada és diferent i menor que 20.000

- **Exercici 2** - Implementa un programa en JAVA (PR212Threads.java) amb:

- Una classe que implementa 'Runnable'
- El programa ha d'ordenar l'arxiu "llista.txt" amb la comanda:

```
Process p = Runtime.getRuntime().exec("sort -o ordenat.txt llista.txt", new  
String[0], new File("./src/"));
```

- El programa ha d'acabar abans que la ordenació, però la màquina virtual ha d'esperar a que el fil d'ordenar hagi acabat (setDaemon a false)
- Quan acaba el programa escriu "Programa acabat"

- Quan comença a ordenar escriu “Ordenant ...”
- Quan acaba d’ordenar escriu “Ordenació acabada”
- Explica la diferència entre posar ‘setDaemon’ a ‘true’ o posar-ho a ‘false’

La sortida esperada amb `setDaemon(false)` és obtenir l'arxiu 'ordenat.txt'

```
30 e.printStackTrace();
31 }
```

Problems @ Javadoc Declaration

<terminated> Threads2 [Java Application] /Users/alexis/...

Ordenant ...

Programa acaba

Ordenació acabada

La sortida esperada amb `setDaemon(true)` no obté l'arxiu 'ordenat.txt'

44

Problems @ Javadoc Declaration

<terminated> Threads2 [Java Application] /Users/ale...
Programa acaba
Ordenant ...