



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 sobre 90%- Les metodologies de treball, organització personal i participació contenen un 10%
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 serà: 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 mètode d'exclusió mútua en [aquest post](#)



Tasques:

0 - Implementa un programa en JAVA (Threads10.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
```



```
public class Main {  
  
    @ ItsIvanPsk  
    public static void main(String[] args) {  
        Main m = new Main();  
  
        Thread T0 = new Thread(new Threads10( numThread: 0), name: "Thread 0");  
        Thread T1 = new Thread(new Threads10( numThread: 0), name: "Thread 0");  
        Thread T2 = new Thread(new Threads10( numThread: 0), name: "Thread 0");  
        Thread T3 = new Thread(new Threads10( numThread: 0), name: "Thread 0");  
        Thread T4 = new Thread(new Threads10( numThread: 0), name: "Thread 0");  
        T0.start();  
        T1.start();  
        T2.start();  
        T3.start();  
        T4.start();  
  
        try{  
            Thread.sleep( millis: 2000);  
            T0.stop();  
            T1.stop();  
            T2.stop();  
            T3.stop();  
            T4.stop();  
        } catch (InterruptedException e) {  
            throw new RuntimeException(e);  
        }  
    }  
}
```



```
package Ejercicio0;  
  
/* ItsIvanPsk */  
public class Threads10 implements Runnable{  
    2 usages  
    private int numThread;  
    5 usages /* ItsIvanPsk */  
    public Threads10(int numThread) { this.numThread = numThread; }  
    /* ItsIvanPsk */  
    @Override  
    public void run() {  
        while (true){  
            try {  
                Thread.sleep(10);  
                System.out.println("Hola mon: " + numThread);  
            } catch (InterruptedException e) {  
                throw new RuntimeException(e);  
            }  
        }  
    }  
}
```

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrain  
Hola mon: 1  
Hola mon: 2  
Hola mon: 3  
Hola mon: 0  
Hola mon: 4  
Hola mon: 0  
Hola mon: 2  
Hola mon: 3  
Hola mon: 1  
Hola mon: 0
```

1 - Implementa un programa en JAVA (Threads11.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 principal i la mostra al final
- Explica perquè el resultat cada vegada és diferent i menor que 20.000

Porque los diferentes Threads se pisan y están accediendo todos a la vez a la misma variable y ajustando el valor en base a su proceso, el problema es que este puede no ser el mismo que el de los demás Threads es por ello que al sobrescribirse los valores nunca llegarán a 20k.

2 - Implementa un programa en JAVA (Threads2.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 diferencia es que si tenemos el setDaemon en false el programa no esperará a que el Thread acabe su proceso, en cambio si lo seteamos en true el programa deberá de esperar a finalizar el proceso del Thread.

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

```
30         e.printStackTrace();  
31     }  
  
<terminated> Threads2 [Java Application] /Users/al  
Ordenant ...  
Programa acabat  
Ordenació acabada
```

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



44

Problems Javadoc Declaration C
<terminated> Threads2 [Java Application] /Users/al
Programa acabat
Ordenant ...

```
1 public class Main3 {  
2     public static void main(String[] args) {  
3         System.out.println("Ordenant...");  
4         Thread T0 = new Thread(new Threads3());  
5         T0.setDaemon(false);  
6         T0.start();  
7         System.out.println("Programa acabat");  
8     }  
9 }
```

```
import java.io.File;  
import java.io.IOException;  
  
1 usage  
public class Threads3 implements Runnable {  
    @Override  
    public void run() {  
        try {  
            Process p = Runtime.getRuntime().exec("sort /o ordenat.txt llista.txt", new String[0], new File("src/"));  
            System.out.println("Ordenación acabada");  
        } catch (IOException e) {  
            System.out.println("Error exp");  
            throw new RuntimeException(e);  
        }  
    }  
}
```

```
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\...  
Ordenant...  
Programa acabat  
Ordenación acabada  
|  
Process finished with exit code 0
```