

CICLE FORMATIU: CFGS Desenvolupament Aplicacions Multiplataforma DAM
MÒDUL PROFESSIONAL: MP09 Programació de serveis i processos
UNITAT FORMATIVA: UF2. Processos i fils

=====

Pràctica 6. Sincronització multifil.

Programarem amb threads i accessos concurrents un sistema de vendes de entrades.

Per a cada espectacle es treuen a la venda 100 entrades. Utilitzarem un vector de 100 posicions per controlar-les. Valors del vector: 0 entrada no venuda, 1 entrada venuda.

El sistema comença posant a zero tot el vector de entrades.

Després crearà 10 threads que simulen 10 compradors que accedeixen desde diferents llocs. Cada thread farà el següent:

Genera un numero de 1 al 5 que son les entrades que volen.

Per cada entrada genera un numero del 1 al 100 que es la entrada que es vol. Si la entrada està venuda generem altre cop el numero de entrada. El sistema espera 1s per simular que l'usuari s'ho està rumiant.

L'usuari compra la entrada, aquesta compra cal que sigui amb accés exclusiu. La entrada es posa com venuda (+1 al valor del vector) si encara es lliure (cal tornà a comprovar, si s'ha venut es perd).

Es recullen els threads i llistem quantes entrades estan a 0, 1 y 2 o mes (si es 2 vol dir que la entrada s'ha venut dos vegades).

Lliura el codi font ben comentat , i un parell d'execucions.

```
UP2 - Practica_Threads/6/src/threads/Main.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help

Package Explorer | Main.java | Comprador.java | Entrada.java
Practica_Threads_6 | src | threads | J | J | J
Practica_Threads_4 [MR_Pro]
Practica_Threads_1 [MR_Pro]
Practica_Threads_2 [MR_Pro]
Practica_Threads_3 [MR_Pro]
Practica_Threads_5 [MR_Pro]
Practica_Threads_6 [MR_Pro]
JRE System Library [JavaS]
src
  threads
    J
      Main.java
      Comprador.java
      Entrada.java
      Main.java

50 public static void main(String[] args) throws InterruptedException {
51     // 1000 Auto-generated method stub
52     //
53     // Inicializamos la array con 100 posiciones y sus valores a 0
54     initializeArray();
55     //
56     // 10 hilos distintos de 10 compradores diferentes.
57     //
58     Comprador c = new Comprador();
59     c.start();
60     Comprador c1 = new Comprador();
61     c1.start();
62     Comprador c2 = new Comprador();
63     c2.start();
64     Comprador c3 = new Comprador();
65     c3.start();
66     Comprador c4 = new Comprador();
67     c4.start();
68     Comprador c5 = new Comprador();
69     c5.start();
70     Comprador c6 = new Comprador();
71     c6.start();
72     Comprador c7 = new Comprador();
73     c7.start();
74     Comprador c8 = new Comprador();
75     c8.start();
76     Comprador c9 = new Comprador();
77     c9.start();
78     //
79     // Estos join son para asegurar que todos los hilos acaben y a si poder imprimir
80     // un resultado final
81     //
82     c.join();
83     c1.join();
84     c2.join();
85     c3.join();
86     c4.join();
87     c5.join();
88     c6.join();
89     c7.join();
90     c8.join();
91     c9.join();
92     System.out.println(Entrada.entradas);
93 }
94
95 //
96 // Metodo para inicializar la array
97 //
98 public static void initializeArray() {
99     for (int i = 0; i < 100; i++) {
100         Entrada.entradas.add(0);
101     }
102     System.out.println(Entrada.entradas);
103 }
104 }
```

```
UP2 - Practica_Threads/6/src/threads/Comprador.java - Eclipse IDE
File Edit Refactor Source Navigate Search Project Run Window Help

Package Explorer | Main.java | Comprador.java | Entrada.java
Practica_Threads_6 | src | threads | J | J | J
Practica_Threads_4 [MR_Pro]
Practica_Threads_1 [MR_Pro]
Practica_Threads_2 [MR_Pro]
Practica_Threads_3 [MR_Pro]
Practica_Threads_5 [MR_Pro]
Practica_Threads_6 [MR_Pro]
JRE System Library [JavaS]
src
  threads
    J
      Main.java
      Comprador.java
      Entrada.java
      Main.java

1 package threads;
2
3 public class Comprador extends Thread {
4
5     //
6     // Metodo para nicar cuantas entradas va a comprar cada comprador.
7     //
8     public int inputQuantity() {
9         int result = (int) (Math.random() * 5 + 1);
10        return result;
11    }
12
13    //
14    // El run de los hilos.
15    //
16    public void run() {
17        int quantityTickets = inputQuantity();
18        //
19        // Recomendamos tantas veces como entradas vaya a comprar.
20        //
21        for (int i = 0; i < quantityTickets; i++) {
22            Entrada.buyTickets();
23        }
24    }
25
26 }
27 }
```


