

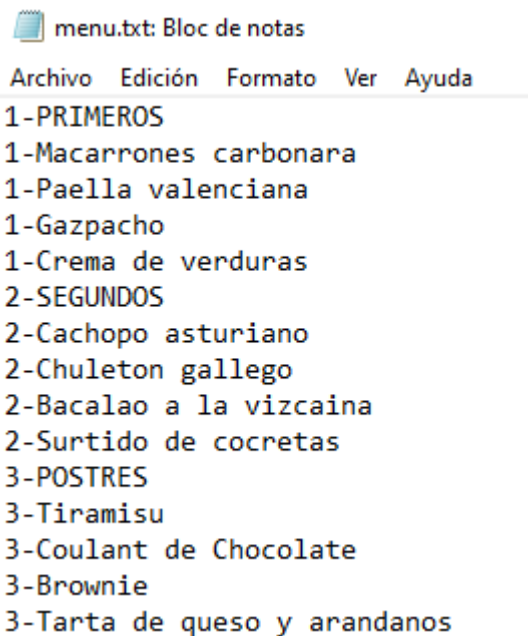
Repàs UD2. Programacio multifil

EJERCICIO 12 – Principal.java e Hilo.java

Vamos a programar en Java la lectura del menú de un restaurante (menu.txt) que ofrece varios primeros, varios segundos y varios postres. Para ello, tendremos 3 hilos que formarán parte de un grupo de hilos (Threadrgoup) llamado “Menu diario” y que trabajarán del siguiente modo:

- Un hilo que lee de menu.txt solo los primeros (que vienen precedidos de la marca 1-) y los vuelca al archivo primeros.txt
- Un hilo que lee de menu.txt solo los segundos (que vienen precedidos de la marca 2-) y los vuelca al archivo segundos.txt
- Un hilo que lee de menu.txt solo los postres (que vienen precedidos de la marca 3-) y los vuelca al archivo postres.txt

Antes de codificar la solución en Java, tómate un tiempo en diseñar tu propio menú que debería tener el siguiente formato:



```
menu.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
1-PRIMEROS
1-Macarrones carbonara
1-Paella valenciana
1-Gazpacho
1-Crema de verduras
2-SEGUNDOS
2-Cachopo asturiano
2-Chuleton gallego
2-Bacalao a la vizcaina
2-Surtido de cocretas
3-POSTRES
3-Tiramisu
3-Coulant de Chocolate
3-Brownie
3-Tarta de queso y arandanos
```

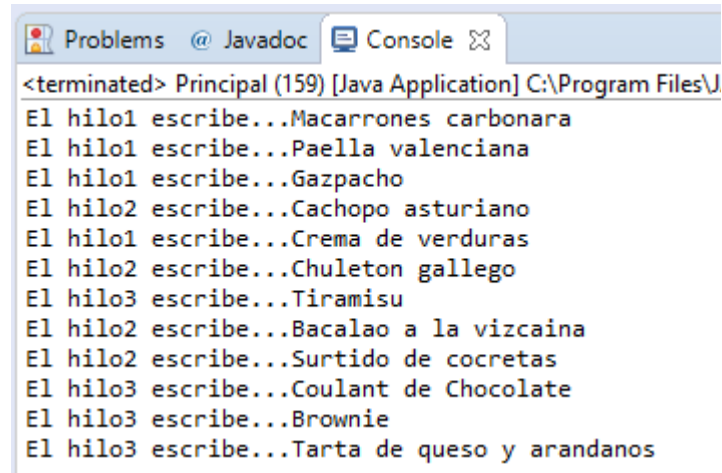
El objetivo es que cada hilo trabaje solo una parte del menú y que acabe escribiendo en su fichero pero sin el rótulo correspondiente y sin el número y el guión que precede a cada plato.

Es decir, en el fichero “primeros.txt” solo debería aparecer las siguientes líneas:

```
Macarrones carbonara
Paella valenciana
Gazpacho
Crema de verduras
```

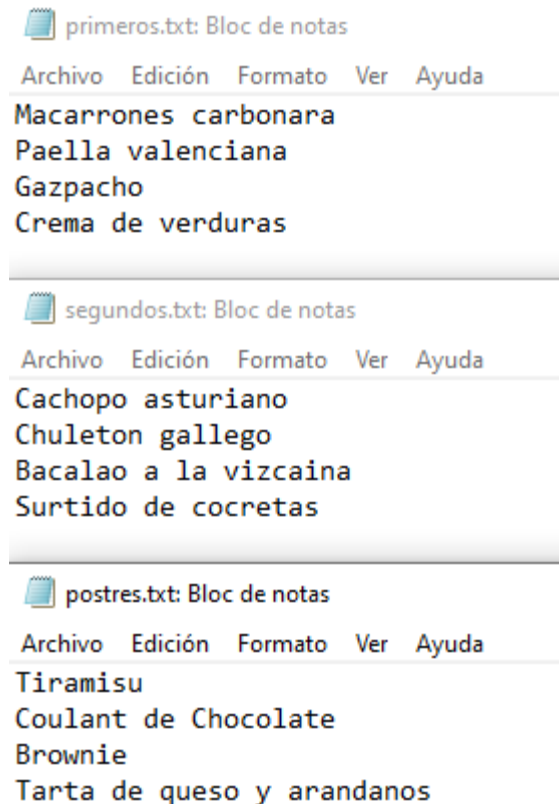
Además, el programa imprimirá en cada momento el hilo que está actuando y qué está escribiendo. Un ejemplo de ejecución sería el siguiente, que utiliza el menú.txt de arriba:

En la consola, al finalizar el programa, se mostrarán las siguientes acciones:



```
<terminated> Principal (159) [Java Application] C:\Program Files\J
El hilo1 escribe...Macarrones carbonara
El hilo1 escribe...Paella valenciana
El hilo1 escribe...Gazpacho
El hilo2 escribe...Cachopo asturiano
El hilo1 escribe...Crema de verduras
El hilo2 escribe...Chuleton gallego
El hilo3 escribe...Tiramisu
El hilo2 escribe...Bacalao a la vizcaina
El hilo2 escribe...Surtido de cocretas
El hilo3 escribe...Coulant de Chocolate
El hilo3 escribe...Brownie
El hilo3 escribe...Tarta de queso y arandanos
```

Que habrá generado los siguientes archivos:



EJERCICIO 13 – Principal.java, Casilla.java, Hilo1.java, Hilo2.java

Vamos a programar en Java la gestión de 2 hilos que accederán por turnos a un vector de 10 posiciones para ir completándolo con valores aleatorios.

Tendremos un primer hilo que comenzará a completarlo de izquierda a derecha (hilo1) y otro segundo hilo que lo completará de derecha a izquierda (hilo2). Ambos lo irán informando con valores aleatorios cuyos límites inferior y superior se solicitarán al principio del programa.

Cada vez que uno de los hilos inserte un valor, esperaremos un tiempo para de esa forma ir viendo en tiempo real como se va informando el vector.

Se mostrarán las 10 posiciones del vector, que inicialmente estarán informadas a 0 y que poco a poco se irán informando hasta que todo el vector esté completo. Cuando un hilo complete una posición, lo marcará con su identificador (1 si lo ha informado el hilo1 o 2 si lo ha informado el hilo2).

Un ejemplo de ejecución sería el siguiente:

Introduzca limite inferior:

10

Introduzca limite superior:

500

	0	1	2	3	4	5	6	7	8	9

288	0	0	0	0	0	0	0	0	0	0
1										
0	1	2	3	4	5	6	7	8	9	

288	0	0	0	0	0	0	0	0	0	464
1										2
0	1	2	3	4	5	6	7	8	9	

288	116	0	0	0	0	0	0	0	0	464
1	1									2
0	1	2	3	4	5	6	7	8	9	

288	116	0	0	0	0	0	0	0	493	464
1	1								2	2
0	1	2	3	4	5	6	7	8	9	

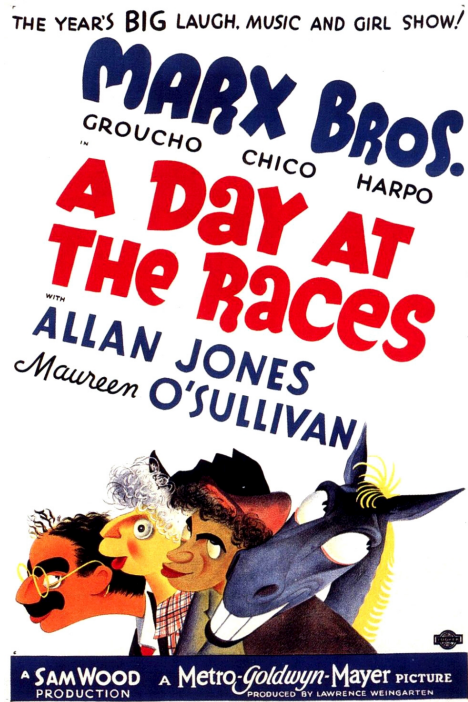
288	116	251	0	0	0	0	0	0	493	464
1	1	1							2	2
0	1	2	3	4	5	6	7	8	9	

288	116	251	0	0	0	0	103	493	464	
1	1	1					2	2	2	

0	1	2	3	4	5	6	7	8	9
288	116	251	165	0	0	0	103	493	464
1	1	1	1				2	2	2
0	1	2	3	4	5	6	7	8	9
288	116	251	165	0	0	360	103	493	464
1	1	1	1			2	2	2	2
0	1	2	3	4	5	6	7	8	9
288	116	251	165	224	0	360	103	493	464
1	1	1	1	1		2	2	2	2
0	1	2	3	4	5	6	7	8	9
288	116	251	165	224	433	360	103	493	464
1	1	1	1	1	2	2	2	2	2

EJERCICIO 14 – Coche.java y Principal.java

Vamos a realizar un programa que simule la competición de varios coches mediante la ejecución de hilos, es decir vamos a pasar...



La carrera funcionará del siguiente modo:

IES Abastos – Desenrotllament d'Aplicacions Multiplataforma – PSP

- Se solicitará al usuario el número de coches que van a competir para, a continuación, pedir la marca de cada coche. Además, a cada uno de ellos se le asignará automáticamente un número, es decir, el primer coche que demos de alta tendrá el número 1, el segundo el número 2, etc..

-Una vez tengamos los coches dados de alta, empezará la carrera lanzándose los hilos en ejecución, los cuales correrán siguiendo estos pasos:

- Un coche calculará una distancia aleatoria entre 0 y 100
- Esperará 1 segundo e imprimirá la distancia total recorrida.
- Si la distancia iguala o supera los 500 metros (1 vuelta en circuito indoor), entonces se informa que el coche ha finalizado.

Un ejemplo de ejecución sería el siguiente:

Introduzca numero de coches a competir:

3

Introduzca marca del coche 1:

Aston Martin

Coche creado: Aston Martin(1)

Introduzca marca del coche 2:

BMW

Coche creado: BMW(2)

Introduzca marca del coche 3:

Ford Fiesta

Coche creado: Ford Fiesta(3)

Que empiece la carrera!

BMW(2) lleva recorrido 0km!

Aston Martin(1) lleva recorrido 43km!

Ford Fiesta(3) lleva recorrido 25km!

Aston Martin(1) lleva recorrido 80km!

Ford Fiesta(3) lleva recorrido 62km!

BMW(2) lleva recorrido 3km!

BMW(2) lleva recorrido 58km!

Aston Martin(1) lleva recorrido 110km!

Ford Fiesta(3) lleva recorrido 78km!

Ford Fiesta(3) lleva recorrido 82km!

Aston Martin(1) lleva recorrido 164km!

BMW(2) lleva recorrido 83km!

Ford Fiesta(3) lleva recorrido 174km!

Aston Martin(1) lleva recorrido 201km!

BMW(2) lleva recorrido 142km!

BMW(2) lleva recorrido 204km!

Aston Martin(1) lleva recorrido 269km!

Ford Fiesta(3) lleva recorrido 208km!

Ford Fiesta(3) lleva recorrido 280km!

BMW(2) lleva recorrido 234km!

Aston Martin(1) lleva recorrido 349km!

BMW(2) lleva recorrido 273km!

Aston Martin(1) lleva recorrido 350km!

Ford Fiesta(3) lleva recorrido 310km!

Ford Fiesta(3) lleva recorrido 335km!

Aston Martin(1) lleva recorrido 449km!

BMW(2) lleva recorrido 315km!

BMW(2) lleva recorrido 400km!

Aston Martin(1) lleva recorrido 466km!
Ford Fiesta(3) lleva recorrido 406km!

El Aston Martin(1) ha finalizado al recorrer 519km.
Ford Fiesta(3) lleva recorrido 487km!

El Ford Fiesta(3) ha finalizado al recorrer 548km.
BMW(2) lleva recorrido 474km!

El BMW(2) ha finalizado al recorrer 518km.

EJERCICIO 15 – Hilo.java, Pasajero.java y Principal.java

Vamos a realizar una travesía en el barco más famoso de la historia, aquel que vió nacer la historia de amor que más Oscars se ha llevado en la historia... la historia del Titanic...



Para ello utilizaremos la base de datos real del Titanic que se facilita junto a esta práctica en formato csv (*titanic.csv*) y en el cual podemos encontrar todos los datos de los tripulantes. De hecho, de cada tripulante vamos a tener la siguiente información separadas por el carácter punto y coma (;):

- La clase donde viajaba (1/2/3)
- El apellido.
- El nombre.
- La edad.
- El lugar de embarque
- Si sobrevivió o no al hundimiento (S/N)

3;Abbing;Mr Anthony;42;Southampton;N

El programa se iniciará con el mensaje “BIENVENIDO AL TITANIC”, para, a continuación, hacer que 3 hilos (uno por cada clase de tripulantes que viajaban en el barco) recorran el fichero csv de tripulantes y se puedan calcular los siguientes datos:

-**Hilo1:** calculará el número de pasajeros que iban en **primera** clase, número de supervivientes y de fallecidos, junto a su porcentaje.

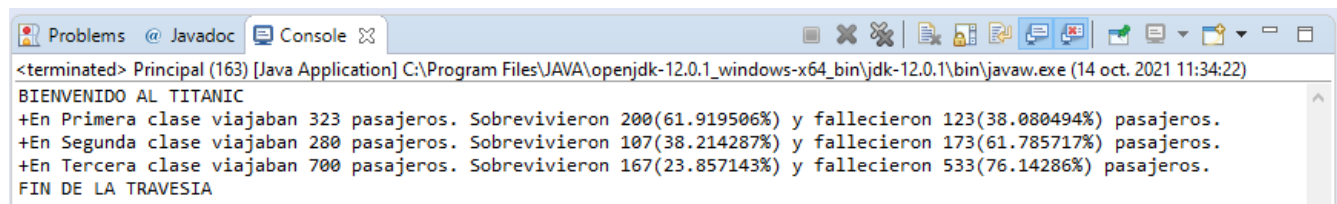
-**Hilo2:** calculará el número de pasajeros que iban en **segunda** clase, número de supervivientes y de fallecidos, junto a su porcentaje.

-**Hilo3:** calculará el número de pasajeros que iban en **tercera** clase, número de supervivientes y de fallecidos, junto a su porcentaje.

La información debería aparecer de forma que, gestionando prioridades, la mayoría de veces aparezca primero la información de la primera clase (haciendo uso de *Thread.MAX_PRIORITY*).

Una vez impresa esta información (y no antes) aparecerá el mensaje “FIN DE LA TRAVESÍA”

Un ejemplo de ejecución sería el siguiente:



```
<terminated> Principal (163) [Java Application] C:\Program Files\JAVA\openjdk-12.0.1_windows-x64_bin\jdk-12.0.1\bin\javaw.exe (14 oct. 2021 11:34:22)
BIENVENIDO AL TITANIC
+En Primera clase viajaban 323 pasajeros. Sobrevivieron 200(61.919506%) y fallecieron 123(38.080494%) pasajeros.
+En Segunda clase viajaban 280 pasajeros. Sobrevivieron 107(38.214287%) y fallecieron 173(61.785717%) pasajeros.
+En Tercera clase viajaban 700 pasajeros. Sobrevivieron 167(23.857143%) y fallecieron 533(76.14286%) pasajeros.
FIN DE LA TRAVESIA
```

EJERCICIO 16 – Principal.java, Contador.java, Hilosuma.java e Hiloresta.java

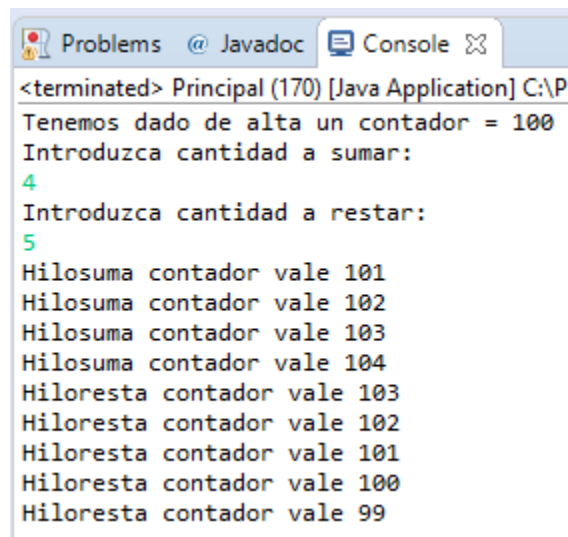
Vamos a realizar una aplicación multihilo para que 2 hilos accedan a un objeto compartido llamado “Contador”. Este contador estará inicializado a 100, tendrá un método para incrementarlo 1 a 1 y otro método para decrementarlo 1 a 1.

El contador será accedido por 2 hilos: hilosuma.java e hiloresta.java, que sumarán y restarán del contador la cantidad que se introduzca por teclado al inicio del programa.

Utiliza **bloques sincronizados** dentro del método run() de los hilos para que, segundo a segundo veamos como va variando el valor del contador (utiliza un sleep(1000)).

Como el acceso lo vamos a hacer de forma sincronizada, primero entrará un hilo (suma o resta, el que sea), finalizará y luego entrará a ejecutarse el siguiente.

Un ejemplo de ejecución sería el siguiente:



```
<terminated> Principal (170) [Java Application] C:\P
Tenemos dado de alta un contador = 100
Introduzca cantidad a sumar:
4
Introduzca cantidad a restar:
5
Hilosuma contador vale 101
Hilosuma contador vale 102
Hilosuma contador vale 103
Hilosuma contador vale 104
Hiloresta contador vale 103
Hiloresta contador vale 102
Hiloresta contador vale 101
Hiloresta contador vale 100
Hiloresta contador vale 99
```

EJERCICIO 18 – Principal.java, Autobus.java y Speed.java

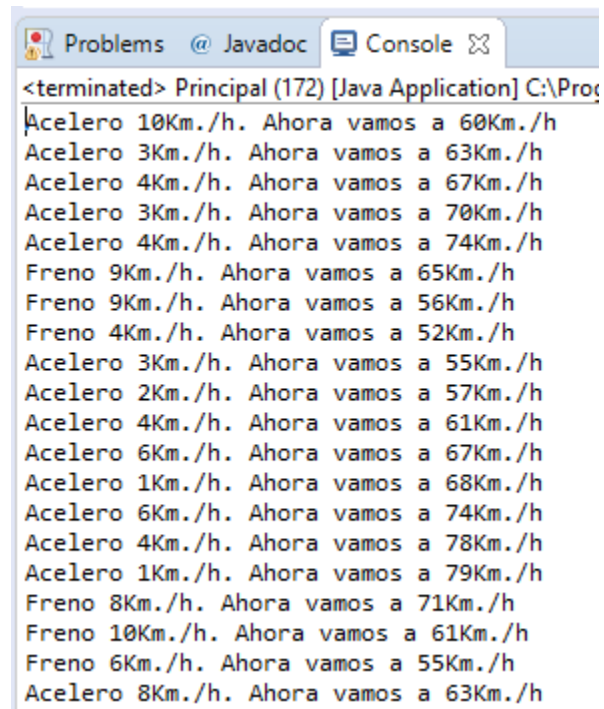
Vamos a realizar un programa en Java que simule cómo funciona el autobús de la película Speed (1994), en la cual el autobús debe circular entre 50 y 80 Km/h, de lo contrario estallará una bomba dentro del mismo. Si quieres ver como esta película hiere todas las leyes de la física, simplemente mira [esta](#) escena en la que el autobús es capaz de saltar un hueco en obras de una autopista. ¡Fascinante!

El programa en Java utilizará un objeto Autobús que tendrá una matrícula y una velocidad, que estará inicializada a 50 Km./h. Este autobús será el objeto compartido para unos hilos de ejecución que pertenecerán a la clase “Speed” y que modificarán su velocidad con un número aleatorio entre 1 y 10 Km./h.

Esta clase “Speed” comenzará acelerando el autobús (que parte de una velocidad inicial de 50 Km.h) poco a poco hasta llegar (o superar) los 80 km/h., en cuyo momento comenzará a frenar, poco a poco hasta llegar (o rebasar) a los 50 Km./h, en cuyo momento volverá a acelerar...y así cíclicamente. De hecho el programa se convertirá en un bucle infinito en el cual esperaremos 1 segundo entre cada acelerada/frenada para que se vea con más claridad cómo va avanzando el autobús.

Para poder realizar el programa, los hilos de tipo “Speed” han de “hablar” entre ellos, esperándose cuando lleguen a las velocidades límite (wait) y despertando al hilo “contrario” en ese caso (notify). Es decir, cuando se espere el de “acelerar”, notificará al de “frenar” y a la inversa.

Un ejemplo de ejecución sería el siguiente:



```
<terminated> Principal (172) [Java Application] C:\Progr
Acelero 10Km./h. Ahora vamos a 60Km./h
Acelero 3Km./h. Ahora vamos a 63Km./h
Acelero 4Km./h. Ahora vamos a 67Km./h
Acelero 3Km./h. Ahora vamos a 70Km./h
Acelero 4Km./h. Ahora vamos a 74Km./h
Freno 9Km./h. Ahora vamos a 65Km./h
Freno 9Km./h. Ahora vamos a 56Km./h
Freno 4Km./h. Ahora vamos a 52Km./h
Acelero 3Km./h. Ahora vamos a 55Km./h
Acelero 2Km./h. Ahora vamos a 57Km./h
Acelero 4Km./h. Ahora vamos a 61Km./h
Acelero 6Km./h. Ahora vamos a 67Km./h
Acelero 1Km./h. Ahora vamos a 68Km./h
Acelero 6Km./h. Ahora vamos a 74Km./h
Acelero 4Km./h. Ahora vamos a 78Km./h
Acelero 1Km./h. Ahora vamos a 79Km./h
Freno 8Km./h. Ahora vamos a 71Km./h
Freno 10Km./h. Ahora vamos a 61Km./h
Freno 6Km./h. Ahora vamos a 55Km./h
Acelero 8Km./h. Ahora vamos a 63Km./h
```