

## Ejercicios tema 2

1. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $x = y + 1$ ;

S2:  $y = x * 2$ ;

S3:  $z = a + b$ ;

2. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias..

S1:  $\text{cuad} = x * x$ ;

S2:  $m1 = a * \text{cuad}$ ;

S3:  $m2 = b * x$ ;

S4:  $z = m1 + m2$ ;

S5:  $y = z + c$ ;

3. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $a = b + c$ ;

S2:  $d = b + e$ ;

S3:  $f = c + e$ ;

S4:  $g = \text{función}(a, d, f)$ ;

S5:  $f = \text{sen}(w)$ ;

4. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

P1:  $u = a * b$ ;

P2:  $v = u + c$ ;

P3:  $w = u - d$ ;

P4:  $x = v / w$ ;

5. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $x = (A+B) * (A-B)$ ;

S2:  $y = (C-D) * (C + D)$ ;

S3:  $z = x + y$ ;

6. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $a = 2 * x$ ;

S2:  $b = y - z$ ;

S3:  $c = b / 3$ ;

S4:  $d = a + c$ ;

7. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $a = x + y$ ;

S2:  $b = z - 1$ ;

S3:  $c = a - b$ ;

S4:  $w = c + 1$ ;

8. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $a = x + y$ ;

S2:  $b = x - y$ ;

S3:  $c = x * y$ ;

S4:  $d = a + b + c$ ;

S5:  $e = a * b * c$ ;

S6:  $f = d - e$ ;

9. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $a = x + y$ ;

S2:  $b = a + 3$ ;

S3:  $c = x * y$ ;

S4:  $d = c + 8$ ;

S5:  $e = b + d$ ;

10. Realizar un programa Java utilizando hilos para programar de forma concurrente el siguiente trozo de código. Se introducirán por teclado las variables necesarias.

S1:  $a = x * x$ ;

S2:  $m = a * b$ ;

S3:  $n = c * x$ ;

S4:  $y = m + n$ ;

S5:  $z = m - n$ ;

11. Realizar un programa Java que calcule utilizando hilos la suma, la multiplicación, la diferencia y la división de la generación aleatoria de dos números enteros del 1 al 100.
12. Realizar un programa Java que calcule utilizando hilos la suma, la multiplicación, el máximo valor y el mínimo valor de la generación aleatoria de 100 números enteros del 1 al 100.
13. Vamos a simular el proceso de cobro de un supermercado; es decir, unos clientes van con un carro lleno de productos y una cajera les cobra los productos, pasándolos uno a uno por el escaner de la caja registradora. En este caso la cajera debe de procesar la compra cliente a cliente, es decir que primero le cobra al cliente 1, luego al cliente 2 y así sucesivamente. Para ello vamos a definir una clase "Cajera" y una clase "Cliente" el cual tendrá un "array de enteros" que representaran los productos que ha comprado y el tiempo que la cajera tardará en pasar el producto por el escaner; es decir, que si tenemos un array con [1,3,5] significará que el cliente ha comprado 3 productos y que la cajera tardara en procesar el producto 1 '1 segundo', el producto 2 '3 segundos' y el producto 3 en '5 segundos', con lo cual tardara en cobrar al cliente toda su compra '9 segundos'.
14. Resolver el problema del productor/consumidor.
15. Resolver el problema de los lectores y los escritores.
16. Resolver el problema de la comida de los filósofos.

17. Escribe una clase llamada SuperMarket que implemente el funcionamiento de N cajas de un supermercado. Los M clientes del supermercado estarán un tiempo aleatorio comprando y con posterioridad seleccionarán de forma aleatoria en qué caja posicionarse para situarse en su cola correspondiente. Cuando les toque el turno serán atendidos procediendo al pago correspondiente e ingresando en la variable Resultados del supermercado. Se deben crear tantos threads como clientes haya y los parámetros M y N se deben pasar como argumentos al programa. Para simplificar la implementación, el valor de pago de cada cliente puede ser aleatorio.
18. Escribe una clase llamada ModernSuperMarket que implemente el funcionamiento de N cajas de supermercado. Los mismos M clientes del supermercado realizarán el mismo proceso que en el ejercicio anterior, situándose cuando han realizado la compra, en este caso, en una única cola. Cuando cualquier caja esté disponible, el primero de la cola será atendido en la caja correspondiente. Calcula el tiempo medio de espera por cliente y compáralo con el tiempo medio que se obtendría en el ejercicio anterior. ¿Cuál de las dos alternativas es más eficiente? ¿Cuál elegirías si tú tuvieras un supermercado? Razona la respuesta.
19. Escribe una clase llamada Parking que reciba el número de plazas del parking y el número de coches existentes en el sistema. Se deben crear tantos threads como coches haya. El parking dispondrá de una única entrada y una única salida. En la entrada de vehículos habrá un dispositivo de control que permita o impida el acceso de los mismos al parking, dependiendo del estado actual del mismo (plazas de aparcamiento disponibles). Los tiempos de espera de los vehículos dentro del parking son aleatorios. En el momento en el que un vehículo sale del parking, notifica al dispositivo de control el número de la plaza que tenía asignada y se libera la plaza que estuviera ocupando, quedando así estas nuevamente disponibles. Un vehículo que ha salido del parking esperará un tiempo aleatorio para volver a entrar nuevamente en el mismo. Por tanto, los vehículos estarán entrando y saliendo indefinidamente del parking. Es importante que se diseñe el programa de tal forma que se asegure que, antes o después, un vehículo que permanece esperando a la entrada del parking entrará en el mismo (no se produzca inanición).