Programación de Servicios y Procesos

Tarea 2.02 – Gestión de datos compartidos (sencilla)

1.- Crea una aplicación, llamada psp.unidad02.tarea202. CalculaPrimosCompartidosApp que realice el cálculo de los números primos comprendidos en un rango de números determinado. La forma de funcionar de la aplicación será la siguiente:

- Recibir por línea de comandos como argumentos dos valores enteros obligatorios. El primero será el límite inferior del rango y el segundo el límite superior. Se deben proporcionar los dos parámetros, ambos deben ser números enteros positivos válidos y el límite inferior no debe ser superior al límite superior del rango. En caso de algún problema con los parámetros se debe mostrar un mensaje de error y terminar. No se deberían mostrar excepciones ni trazas.
- Crear tantos hilos workers como núcleos o procesadores haya en el sistema. Cada worker se encargará de localizar los números primos en una parte del rango (un subrango). Los límites del subrango correspondiente a cada worker lo determinará el hilo maestro usando un método que haga un reparto justo, esto es, si no puede ser que todos los rangos sean iguales, por lo menos que ninguno tenga una longitud superior a otro por más de una unidad.
- Cada worker registrará los números en un objeto **compartido entre todos**. Este objeto permitirá a los workers ir almacenando los primos que se encuentren. Este objeto será creado por el hilo maestro, que lo proporcionará a los workers.
- El hilo maestro esperará a que todos los workers acaben y mostrará por pantalla los números contenidos en el objeto compartido, ordenando los números allí contenidos.

Consideraciones de diseño:

- Habrá una clase principal (psp.unidad02.tarea202.CalculaPrimosCompartidosApp), que contendrá el punto de entrada de la aplicación (main), procesará los parámetros, creará el objeto compartido y los workers. esperará a que éstos finalicen y mostrará los resultados.
- Deberá haber una clase para el objeto compartido que almacenará los primos. Esta clase debe tener métodos para añadir números y para poder recuperar los números almacenados. Si se desea se puede añadir funcionalidad para ordenar los números o se pueden ordenar desde fuera una vez recuperados (tú decides). Ten en cuenta que los objetos de esta clase (uno sólo en realidad) se van a acceder de manera simultánea por varios hilos por lo que hay que vigilar

problemas de sincronización.

- Deberá haber una clase para los workers. Dado que todos hacen el mismo trabajo una sola clase bastará para todos
- Separa la lógica de la acción usando la interfaz runnable

Debes entregar:

- Un archivo jar ejecutable de nombre **CalculaPrimosCompartidosApp.jar**
- Un archivo comprimido con el **javadoc** generado por tu hilo
- Un archivo comprimido con el proyecto completo.

Programación de Servicios y Procesos

Tarea 2.03 – Gestión de datos compartidos (dificultad)

1. Vamos a simular la pizzería Luigi's.

En la pizzería tenemos dos tipos de "actores". Por un lado están los pizzeros y por otro los clientes.

La vida del pizzero es estar eternamente haciendo pizzas. Cada vez que hace una pizza, lo que puede tardar entre 5 y 10 segundos, de forma aleatoria, el pizzero la saca del horno, la pone en una bandeja (infinita, por lo que siempre cabrán nuevas pizzas) y pasa a preparar la siguiente.

La vida del cliente es algo más rica. Un cliente llega a la pizzería e intenta tomar una pizza de la bandeja. Si hay pizzas en la bandeja, toma una y se va (pagando, imagino). Si no hay pizzas en la bandeja, se va inmediatamente sin comer. En cualquier caso dará un paseo (en el que puede tardar entre 20 y 30 segundos si ha comido alguna pizza con anterioridad y entre 10 y 15 si no lo ha hecho) y vuelve a por pizza. El cliente se comerá 5 pizzas antes de darse por satisfecho e irse a casa.

Cuando todos los clientes estén en casa se podrá cerrar la pizzería y mandar a los pizzeros también a su casa.

Se desea crear una aplicación Java que simule esta situación usando múltiples hilos de ejecución. Para ello:

- El hilo principal creará tantos pizzeros y clientes como se indique por parámetros. Estos serán dos números enteros positivos mayores que 1. Serán opcionales y en caso de no proporcionarse se crearán 2 pizzeros y 5 clientes. El primer parámetro deberá ser el número de pizzeros y el segundo el de clientes.
- Una vez leidos los parámetros, el hilo principal creará tantos hilos pizzeros y clientes como se haya indicado en los parámetros y dará comienzo a la simulación.
- Para seguir el curso de la simulación, tanto los pizzeros como los clientes irán mostrando por pantalla lo que están haciendo en cada momento. En concreto:
    - Los pizzeros deberán mostrar un mensaje al:
        - Iniciarse
        - Comenzar a preparar una pizza
        - Terminar de preparar una pizza y colocarla en la bandeja (indicando cuantas hay en la bandeja tras dejar la nueva pizza)
        - Terminen de trabajar y se vayan a casa

- Los clientes deberán mostrar un mensaje al:
  - Iniciarse
  - Entrar en la tienda e intentar tomar una pizza
  - Tomar una pizza, indicando el número que hace de las que se lleva comidas.
  - No poder tomar una pizza
  - Comenzar un paseo
  - Terminar un paseo
  - Terminar de comer y volver a casa.

Consideraciones de diseño:

- La clase principal debe ser psp.unidad02.tarea203.PizzeriaLuigi
- Esta deberá procesar los parámetros y crear los hilos correspondientes a los pizzeros y a los clientes. Asimismo deberá crear un objeto compartido que será la bandeja que recoge las pizzas.
- Este objeto compartido deberá tener métodos para depositar una pizza, para tomar una pizza (si es que hay) y para determinar el número de pizzas de la bandeja. **NOTA: Este es un objeto compartido**.
- Habrá que crear dos clases distintas para los hilos. Una para los hilos pizzeros y otra para los hilos clientes. A todas se les deberá proporcionar un número único para identificarlos y el objeto compartido correspondiente a la bandeja de pizzas.
- Se requiere el uso del interfaz runnable para separar el código de la ejecución de la tarea.

Debes entregar:

- Un archivo jar ejecutable de nombre **PizzaLuigi.jar**
- Un archivo comprimido con el **javadoc** generado por tu hilo
- Un archivo comprimido con el proyecto completo.