

Ejercicio 1

Implementa un programa en java, lanzador java, que lance la ejecución de otro programa java, generador java.

Este último ejecutará el comando IPconfig.exe, los resultados de la ejecución los enviará al lanzador.java.

Lanzador.java enviará utilizando el operador "|" (tuberías) los datos a recogedor.java Recogedor.java

filtrará las líneas obtenidas de lanzador.java, mostrando por pantalla aquellas que tengan la palabra "IP"

Ejemplo de salida :

```
java Ejercicio1.dir_lanzador | java Ejercicio1.dir_recogedor
```

Configuración IP de Windows

Vinculo: dirección IPv6 local:...fe80::cd1a:6f11:67d2:b443%15

Dirección IPv4.....192.168.8.220

Ejercicio 2

En el proyecto base que acompaña este examen, se implementa el cocinero loco .Este va sirviendo dos platos según el número de comensales que lleguen(máximo 5, ej:3 comensales 6 platos)tipo Lizarrán (autoservicio donde según va produciendo platos el cocinero, los comensales irán consumiendo)

En el paquete Ejercicio2 tenemos:

La clase Cola: es el recurso que comparten cocinero(Productor) y comensales (Consumidor))

Esta cola que almacena los platos , nelem con el número de platos cocinados(número de elementos en la cola)

También incluye el método get, que obtiene el plato que consume. Y put que añade un plato cocinado.

La clase Consumidor(comensales) que tiene el recurso compartido cola y el numero

que identifica al comensal. Cada comensal obtiene un plato y se espera un tiempo(tiempo de consumo)

Así hasta que haya consumido dos platos, momento en el que termina de comer.

La clase productor(cocinero) que tiene el recurso compartido cola y n_comensalesnumero, que es el numero de comensales a los que cocinar.2 platos se van a cocinar (producir) por comensal.

El ejercicio consiste en implementar Ejercicio_2.java que contiene el main para instanciar los objetos:

Productor: un cocinero que produce platos

Consumidores: pudiendo ser entre 1 a 5 comensales. El número de comensales se pide por teclado.

Lanzar su ejecución y esperar a que todos hayan terminado(se cierra el bar)

Una salida por pantalla podría ser la siguiente:

Introduzca el número de comensales: 3

Ejercicio 3

Cuando se ejecuta el ejercicio anterior, se pueden observar resultados incoherentes

Esto se debe a que los comensales (consumidores) acceden a la cola a consumir un plato que aun el cocinero no ha elaborado. Como la cola esta vacía y se accede a ella, provoca una excepción `NullPointerException` (para poder ejecutar el ejercicio 2 se ha puesto , `NullPointerException`; pero se debe quitar para la solución del ejercicio 3.

Por tanto, hay que modificar el ejercicio anterior para que genere datos coherentes. Nota: La modificación del código NUNCA IMPLICA ELIMINAR NADA (salvo la excepción `NullPointerException` que se debe quitar), solo añadir lo que se estime conveniente(seguir las pistas del código)

Ejercicio 4

Implementar un programa, `Ejercicio4.java`, para un servidor TCP que escuche por el puerto 6000 y que atienda solamente a un único cliente(servidor no concurrente).El servidor recibe una cadena, una expresión de suma (ejemplo `3+2`).Y servidor devolverá al cliente la cadena con la solución de dicha expresión (ejemplo `5`). Se dispone de la función `split` que dada una cadena (ejemplo: `'3+2'`) obtiene los dos sumandos y devuelve una cadena con el resultado de una suma.

En caso de que la expresión de la suma no sea correcta, el servidor deberá enviar al cliente 'Respuesta al cliente: No expresión válida'

Un ejemplo de la ejecución del programa sería la siguiente:

EJERCICIO 4: SERVIDOR NO CONCURRENTE

Servidor se ha iniciado en el puerto 6000

Esperando al cliente...

Expresión recibida del cliente:

33+22

Solución enviada al cliente: 55

FIN DE EJECUCION DEL SERVIDOR (Ejercicio 4).

package Ejercicio4;

Ejercicio 5

Implementar un programa, `Ejercicio5.java`, para un cliente TCP que se conecte con el servidor del ejercicio4, mediante los parámetros `direccion:localhost` y `puerto:6000`.

Pida por el teclado la expresión de una suma a realizar (ejemplo `3+2`) y el servidor le envíe como respuesta el resultado de la misma.