

1.- Realiza un programa en java que simule un buzón de correo (recurso compartido), de forma que se pueda enviar un mensaje o leer un mensaje. El buzón sólo puede almacenar un mensaje, de manera que para poder escribir se debe encontrar vacío y para poder leer debe estar lleno. Crear varios hilos lectores y escritores que manejen dicho buzón de forma sincronizada.

Clases necesarias:

LectorHilo: codificamos las instrucciones que deben ejecutar los hilos encargados de leer los mensajes del recurso compartido buzón.

EscritorHilo: donde codificaremos las instrucciones que deben ejecutar los hilos encargados de escribir mensajes en el recurso compartido buzón.

BuzonCorreo: donde encapsularemos el recurso compartido buzón y haremos visibles los métodos disponibles por los lectores y escritores para poder acceder de forma sincronizada al buzón. La clase buzónCorreo así concebida posee la estructura de un monitor.

En el programa principal:

Se instancia un objeto BuzonCorreo y varios LectorHilo y EscritorHilo, respectivamente. Podéis probar con dos de cada. Se arrancan los hilos y espera a que terminen.

```
run:
Vega lee mensaje: Ana escribe mensaje numero 1
Pablo lee mensaje: Javier escribe mensaje numero 1
Pablo lee mensaje: Ana escribe mensaje numero 2
Vega lee mensaje: Ana escribe mensaje numero 3
Vega lee mensaje: Javier escribe mensaje numero 2
Pablo lee mensaje: Ana escribe mensaje numero 4
Vega lee mensaje: Javier escribe mensaje numero 3
Vega lee mensaje: Ana escribe mensaje numero 5
Pablo lee mensaje: Javier escribe mensaje numero 4
Vega lee mensaje: Ana escribe mensaje numero 6
Pablo lee mensaje: Javier escribe mensaje numero 5
Pablo lee mensaje: Ana escribe mensaje numero 7
Vega lee mensaje: Ana escribe mensaje numero 8
Vega lee mensaje: Javier escribe mensaje numero 6
Vega lee mensaje: Javier escribe mensaje numero 7
Vega lee mensaje: Ana escribe mensaje numero 9
Pablo lee mensaje: Javier escribe mensaje numero 8
Ana ha terminado de escribir
Pablo lee mensaje: Ana escribe mensaje numero 10
Vega ha terminado de leer.
Pablo lee mensaje: Javier escribe mensaje numero 9
Javier ha terminado de escribir
Pablo lee mensaje: Javier escribe mensaje numero 10
Pablo ha terminado de leer.
BUILD SUCCESSFUL (total time: 8 seconds)
```

Al final, el programa principal deberá imprimir la frase "**FIN DE LA EJECUCIÓN**".
Este mensaje es muy importante que salga el último.

Diseño del monitor con sus atributos y métodos:

Atributos: private String mensaje=""; almacenará los mensajes en una variable de tipo String. Private boolean lleno; Esta variable booleana la usaremos para indicar si existe un mensaje disponible para poder ser leído en el buzón o no.

Métodos sincronizados:

void alamacena(String men): método para almacenar un mensaje en el buzón, siempre que esté vacío. En otro caso, el proceso debe esperar a que se vacíe. Cuando almacene el mensaje, debe avisar a los hilos que estén esperando a leer.

String lee(): método para leer un mensaje del buzón siempre y cuando esté lleno. En otro caso, el proceso deberá esperar que se llene. Cuando lea el mensaje, debe avisar a los hilos que están esperando a escribir en el buzón.

Es importante subrayar que ambos métodos deben permitir el acceso exclusivo al recurso compartido (buzón).

Diseño de las clases hilo implicadas (EscritorHilo):

Atributos: BuzonCorreo bc: Definimos la variable donde guardaremos la referencia al buzón.

Métodos:

EscritorHilo(String p_nombre, BuzonCorreo p_bc): método constructor de la clase.

public void run(): El escritor va a escribir 20 mensajes. En el mensaje escribe el número de mensajes que lleva enviados y su nombre. Al finalizar, escribe en la salida estándar un mensaje indicando que ha terminado.

Diseño de las clases hilo implicadas (LectorHilo):

Atributos: BuzonCorreo bc: definimos la variable donde guardaremos la referencia al buzón.

public void run(): El lector va a leer 20 mensajes. Cada mensaje que lee, lo imprime en la salida estándar indicando el nombre del hilo que lo ha leído. Al finalizar, escribe en la salida estándar un mensaje indicando que ha terminado.

Criterios de calificación

Ejercicio 1 (10 puntos):

- Clase LectorHilo implementada correctamente (0,5)
- Clase EscritorHilo implementada correctamente (0,5)
- Sincronización correcta de la clase BuzonCorreo (3)
- Documentación del código detallada, explicando cómo se ha hecho la sincronización y se han creado los hilos (2)

- Mensajes correctos en consola (2)
- Programa principal (1,5)
- Tratamiento de las excepciones (0,5)

Es importante que los códigos entregados estén debidamente documentados explicando todas las decisiones tomadas.