Cuando tenemos varios hilos que acceden a un recurso compartido, en este caso el buzón, tenemos que sincronizar.

**Sincronizar** significa que vamos a hacer que los hilos se ejecuten en un cierto orden para acceder al recurso compartido y no vamos a dejar que el acceso sea aleatorio. Porque si dejamos que el acceso sea aleatorio, entonces el programa no funciona bien 🡪 Eso es sincronizar.

Entonces, en los lenguajes de programación, no solo en Java, existen directivas de sincronización.

Uno de los más famosos es el monitor, no es, ni más ni menos, que una clase que tiene métodos synchronized (sincronizados)

¿Qué significa que los métodos sean sincronizados? Significa que en el monitor (que es la clase que tiene métodos sincronizados) solo puede haber un hilo ejecutando al mismo tiempo un método sincronizado dentro de la clase, es decir, con otras palabras, que si hay un hilo ejecutando código de **almacena,** si otro hilo intenta ejecutar código de **almacena** al mismo tiempo, no puede ejecutarse, se queda en el wait esperando hasta que el hilo que está ejecutando almacena, termine.

Y no solo de almacena, sino que si otro hilo intenta ejecutar código de otro método como **Lee** tampoco puede hasta que el hilo que está ejecutando código de almacena termine. Esto se llama EXCLUSIÓN MUTUA y es imprescindible para cuando varios hilos acceden a un recurso compartido.

La regla dice que no puede haber dos hilos (o más) al mismo tiempo leyendo o escribiendo en un recurso compartido.

El evitar eso se llama exclusión mutua.

Región Crítica

El código que accede al recurso compartido, que en este caso sería el buzón que es el mensaje que se almacena (la variable mensaje) es lo que se conoce como región crítica.

Resumen

Trozo de código que accede a un recurso compartido 🡪 Región crítica y que como programadores debemos identificar y ese código se tiene que ejecutar en exclusión mutua o lo que es lo mismo solo un hilo se puede estar ejecutando a la vez.

Simplemente poniendo synchronized en el método (y otra forma en los objeto que está en los apuntes) conseguimos la exclusión mutua.

Y a todo esto es lo que se llama monitor.

En el buzón escriben una serie de escritores, los escritores son hilos porque derivan de la clase Thread y lo único que hacen es almacenar en el buzón un mensaje y además lo hacen 10 veces

Los lectores lo único que hacen es leer del buzón, llamando al método lee, 10 veces, lo que hay en el buzón.

Si yo sincronizo ya tengo la exclusión mutua, entonces ¿por qué necesito poner wait, notify, notifyall?

Como son dos hilos diferentes, uno lee y otro escribe, y para su correcta ejecución es importante el orden que se ejecutan, es decir, no se puede almacenar en el buzón un mensaje si previamente un hilo no lo ha vaciado leyendo o viceversa, yo no puedo leer en el buzón si previamente un hilo que almacena mensajes pone un mensaje en el buzón ¿no?

Además de que hay un recurso compartido, hay hilos que dependen de otros, HAY DOS PROBLEMAS.

* Si los hilos no dependieran unos de los otros, con poner synchronized, el problema ya estaría solucionado.
* Pero los hilos que almacenan dependen de los hilos que leen y viceversa. Por lo tanto necesitamos que unos hilos se bloqueen hasta que otros hilos hagan lo que tengan que hacer y notifiquen que lo han hecho y entonces los siguientes ejecuten su tarea.

BUZON

Conclusión de buzón:

Este código, sencillo y corto a priori, es más complejo de lo que parece, porque lo están ejecutando varios hilos lectores y varios hilos escritores. Y se van sincronizando entre sí.

Y cuando un escritor no puede escribir se bloquea, cuando se bloquea entra el lector y lo vacía. Lo notifica para que los escritores se desbloqueen y de todos los que están bloqueados, uno, aleatoriamente se desbloquea y el resto se bloquean de nuevo. El que se desbloquea escribe y notifica a los lectores para que puedan leer.

Y así sucesivamente hasta que terminen. Se están bloqueando constantemente, solo uno se puede desbloquear, solo uno puede leer, solo uno puede escribir.

MAIN

Es muy importante tener claro que el buzón solo se crea una vez. Se crea en el MAIN y se pasa al constructor de cada uno de los hilos.