

Monitores en Java

1. Monitores (repaso)
 - a. Definición.
 - b. Ejemplos. Sincronización en Java.
2. Monitores en Java
 - a. Concepto de monitor.
 - b. Ejemplos.
3. `java.util.concurrent`
 - a. Optimización de recursos en concurrencia.
 - b. También semáforos y monitores.
4. Monitores en .Net

Monitores (repaso) - Definición

Mecanismo de abstracción de datos, que:

- nos permite representar de manera abstracta un recurso compartido por medio de un conjunto de variables que definen su estado.
- nos permite realizar operaciones sobre el recurso compartido en condiciones de sincronización y exclusión mutua.

Monitores (repaso) - Composición

- Recursos compartidos y *variables de condición*: estado interno del recurso.
- Código de inicialización.
- Procedimientos internos que manipulan los recursos: delay, resume, empty.
- Procedimientos exportados que pueden ser accedidos por procesos activos que accedan al monitor.
- Conjunto de *colas*.

Monitores (repaso) - Variable de condición

Para realizar la sincronización disponemos de las **variables de condición**.

Una variable de condición es una variable que permite a un proceso liberar su cerrojo y a la vez quedarse bloqueado en espera de una notificación, para después retomar su ejecución desde el mismo punto en el que se quedó bloqueado (tras haber logrado de nuevo el cerrojo).

Monitores (repaso) - Las colas

- Cola de procesos bloqueados que quieren acceder al monitor. (1)
- Cola de procesos bloqueados por la operación *delay* asociada a la variable de condición. (2)
- Cola de cortesía para aquellos procesos bloqueados por la operación *resume*. (3)
- Prioridad a la hora de ser despertados:
 $(3) > (2) > (1)$

Monitores (repaso) - Operaciones

Delay: bloquea el proceso actual y lo manda al final de la cola asociada a la variable de condición; libera el cerrojo de la exclusión mutua para que pueda entrar el nuevo proceso.

Resume: desbloquea el primer proceso que encuentre (o en la cola de cortesía o en la cola asociada a la VC) y lo prepara su ejecución. El proceso actual es mandado a la cola de cortesía y permanecerá allí hasta que el proceso reanudado salga del monitor o se quede bloqueado.

Empty: comprueba si la cola asociada a la VC se encuentra vacía o no.

Monitores (repaso) - Consideraciones sobre JAVA

- En Java no hay cola de hilos bloqueados por VC (van al mismo *pool* que los que se encuentran en espera).
- Tampoco hay cola de cortesía.
- Existen diversas políticas sobre qué debe hacer el hilo que ejecuta la operación *resume*. En JAVA se usa *Signal & Continue*: el hilo que hace signal mantiene la exclusión mutua y el hilo despertado pasa al pool de hilos en espera por adquirir el cerrojo.

Monitores (repaso) - Ejemplos

Monitores en JAVA - Concepto

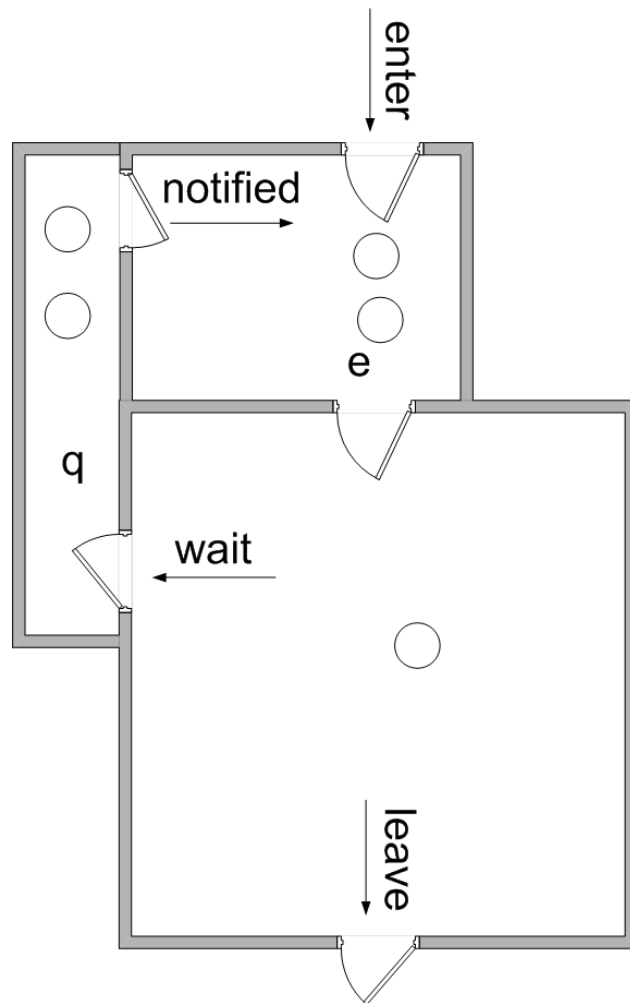
En Java, cualquier objeto puede ser usado como monitor.

Aquellos métodos que requieran hacer uso de la exclusión mutua deben de estar marcados con *synchronized*.

No existen las variables de condición como tales. Sin embargo sí existe una cola de espera asociada al objeto que se usa como monitor.

En Java a los monitores también se les conoce simplemente como cerrojos. Según wikipedia, también se les llama *monitores con variable de condición implícita*.

Monitores en JAVA - Concepto



Monitor al estilo JAVA. La figura representa un objeto con las diferentes transiciones a los estados del monitor.

Monitores en JAVA - Ejemplo