



Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro



Programación Concurrente

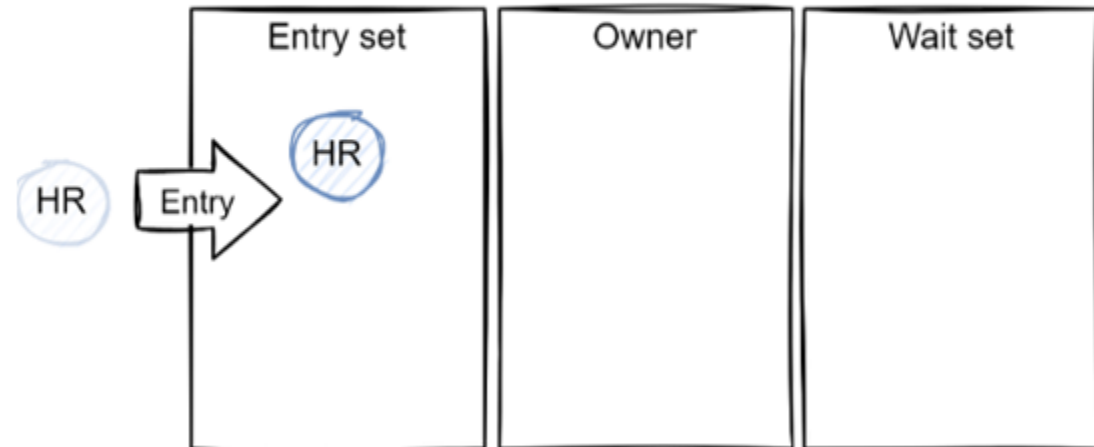
Métodos *notify* y *wait*

Métodos *notify* y *wait*

Hilo Reintegro

Comienza a ejecutarse *hilo reintegro*. Llega al bloque sincronizado, así que entra al entry set del monitor.

```
for(int i = 0; i < 1000; i++) {  
    synchronized (cuenta) {  
        while(cuenta.getSaldo() < 100) {  
            cuenta.wait();  
        }  
        cuenta.reintegrar(100);  
    }  
}
```

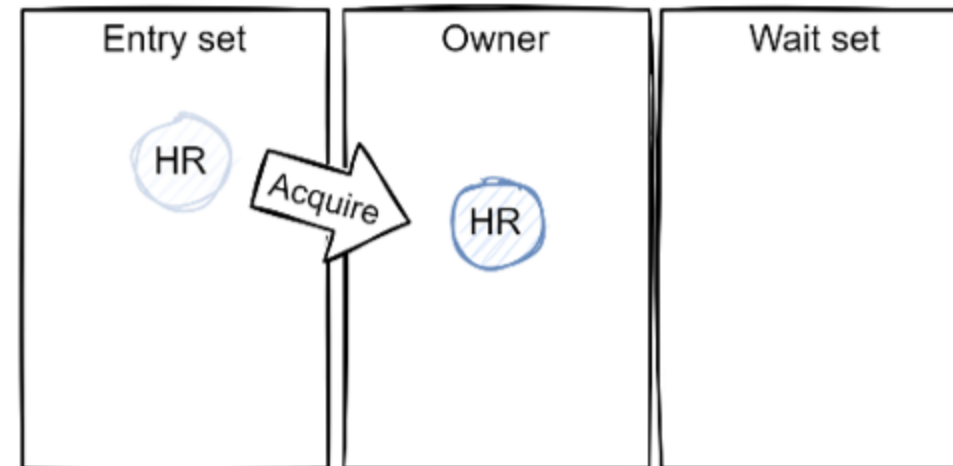


Métodos *notify* y *wait*

Hilo Reintegro

Puesto que el monitor no está bloqueado, el hilo puede entrar al bloque sincronizado, pasando a ser el propietario del monitor. Mientras el saldo sea superior o igual a 100€ realizará reintegros.

```
for(int i = 0; i < 1000; i++) {  
➤ synchronized (cuenta) {  
    while(cuenta.getSaldo() < 100) {  
        cuenta.wait();  
    }  
➤ cuenta.reintegrar(100);  
}}
```

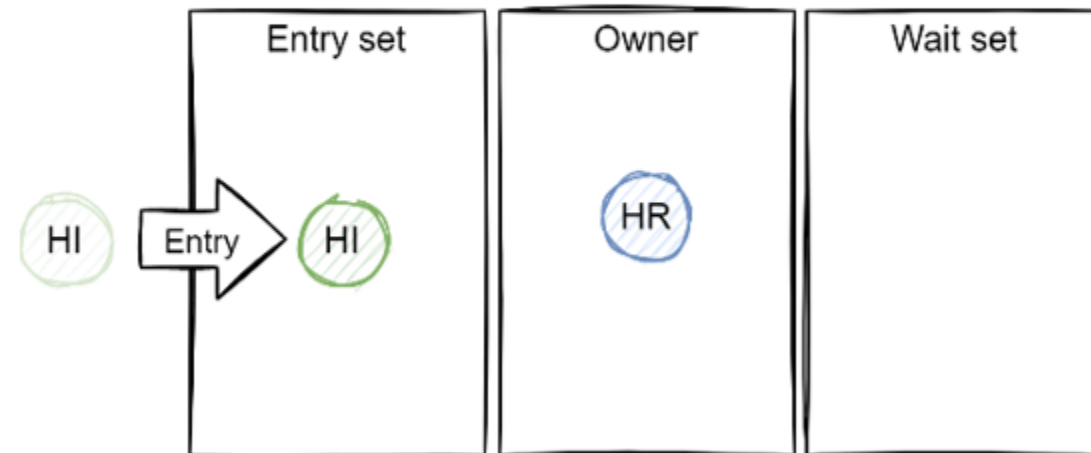


Métodos *notify* y *wait*

Hilo Ingreso

Supongamos que ahora pasa a ejecutarse *hilo ingreso*. Puesto que el monitor está bloqueado por *hilo reintegro*, quedará a la espera en entry set sin poder entrar al bloque sincronizado.

```
➤ for(int i = 0; i < 1000; i++) {  
    synchronized (cuenta) {  
        cuenta.ingresar(100);  
        cuenta.notify();  
    }  
}
```

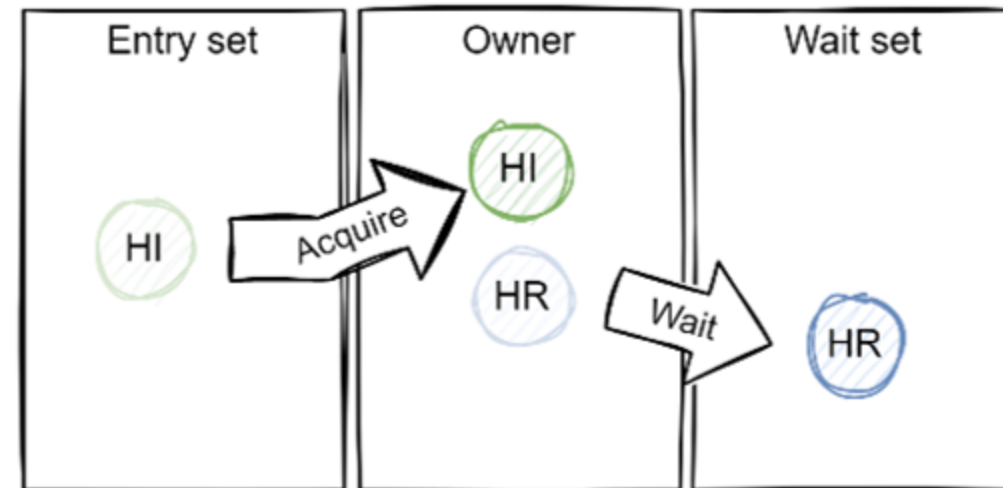


Métodos *notify* y *wait*

Hilo Reintegro

Hilo reintegro continúa su ejecución y llega un momento en el que el saldo es inferior a 100€, por lo que se cumple la condición del *while* y se ejecuta el método *wait*. En ese momento, el hilo pasará a la zona *wait set* donde permanecerá a la espera sin ejecutar ninguna línea más. Como consecuencia liberará el monitor e hilo ingreso tomará su posesión.

```
➤ synchronized (cuenta){  
➤   while(cuenta.getSaldo() < 100){  
    cuenta.wait();  
  }  
  cuenta.reintegrar(100);  
}
```

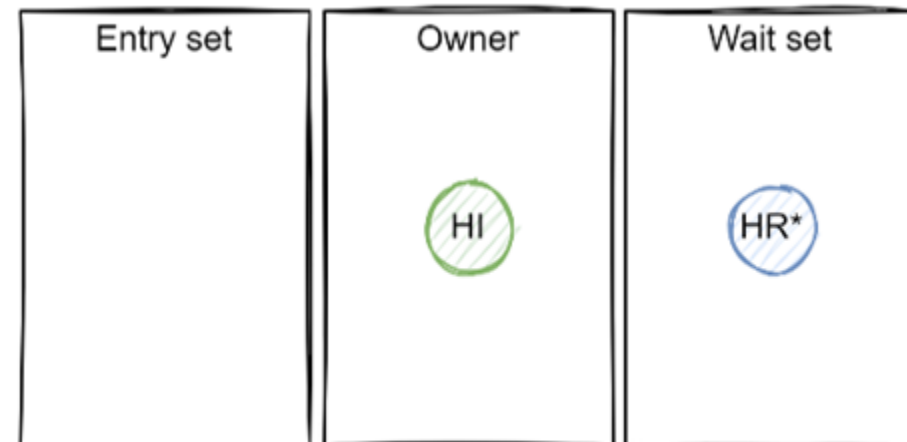


Métodos *notify* y *wait*

Hilo Ingreso

En este momento pasa a ejecutarse *hilo ingreso*. Tras realizar el ingreso, llama al método *notify* del monitor *cuenta*. Esto notificará a uno de los hilos que se encuentren en el *wait set* del monitor *cuenta*. En nuestro caso solo hay un hilo en el *wait set* por lo que será a ese al que se notificará. A partir de este momento, *hilo reintegro* permanecerá a la espera para adquirir la propiedad del monitor y así continuar su ejecución.

```
for(int i = 0; i < 1000; i++) {  
    synchronized (cuenta) {  
        ➤ cuenta.ingresar(100);  
        ➤ cuenta.notify();  
    }  
}
```



* Hilo notificado a la espera de adquirir el monitor.

Métodos *notify* y *wait*

A la hora de llamar a los métodos *notify* y *wait* hay que tener en cuenta:

- Las llamadas deben hacerse siempre desde dentro de bloques sincronizados. Si no lo hacemos, en tiempo de ejecución, cuando se llame a alguno de estos métodos, obtendremos el siguiente error:

```
Exception in thread java.lang.IllegalMonitorStateException:  
current thread is not owner
```

- Las llamadas al método *wait* las haremos siempre desde dentro de un bucle *while* para evitar un efecto llamado *spurious wakups*.