

2.7.2. METODOS SINCRONIZADOS

Se debe evitar la sincronización de bloques de código y sustituirlas siempre que sea posible por la sincronización de métodos, exclusión mutua de los procesos respecto a la variable compartida. Imaginemos la situación que dos personas comparten una cuenta y pueden sacar dinero de ella en cualquier momento; antes de retirar dinero se comprueba siempre si existe saldo. La cuenta tiene 50€, una de las personas quiere retirar 40 y la otra 30. La primera llega al cajero, revisa el saldo, comprueba que hay dinero y se prepara para retirar el dinero, pero antes de retirarlo llega la otra persona a otro cajero, comprueba el saldo que todavía muestra los 50€ y también se dispone a retirar el dinero. Las dos personas retiran el dinero, pero entonces el saldo actual será ahora de -20.

Para sincronizar un método, simplemente añadimos la palabra clave `synchronized` a su declaración. Por ejemplo, la clase `Contador` con métodos sincronizados sería así:

```
public class ContadorSincronizado {  
    private int c = 0;  
  
    public synchronized void incrementa() {  
        c++;  
    }  
  
    public synchronized void decrementa(){  
        c--;  
    }  
  
    public synchronized int valor(){  
        return c;  
    }  
}
```

El uso de métodos sincronizados implica que no es posible invocar dos métodos sincronizados del mismo objeto a la vez. Cuando un hilo está ejecutando un método sincronizado de un objeto, los demás hilos que invoquen a métodos sincronizados para el mismo objeto se bloquean hasta que el primer hilo termine con la ejecución del método.

Cuando un hilo invoca un método `synchronized`, trata de tomar el bloqueo del objeto a que pertenezca. Si esta libre, lo toma y se ejecuta. Si el bloqueo está tomado por otro hilo se suspende el que invoca hasta que aquel finalice y libere el bloqueo. La forma de declararlo es la siguiente:

```
synchronized public void metodo(){  
    //instrucciones atómicas...
```

O Bien:

```
public synchronized void metodo(){  
    //instrucciones atómicas...  
}
```

Se debe tener en cuenta que la sincronización disminuye el rendimiento de una aplicación, por tanto, debe emplearse solamente donde sea estrictamente necesario.

ACTIVIDAD 2.7.2

En una hipotética cuenta bancaria. Crea una clase de nombre Saldo, con un atributo que nos indica el saldo, el constructor que da un valor inicial al saldo. Crea varios métodos uno para obtener el saldo y otro para dar valor al saldo, este método debe informar de quien añade cantidad al saldo, la cantidad que añade, el estado inicial del saldo (antes de añadir la cantidad) y el estado final del saldo después de añadir la cantidad. Define los parámetros necesarios que debe de recibir este método y defínelo como synchronized.

Crea una clase que extienda Thread, desde el método run() hemos de usar el método de la clase Saldo que añade la cantidad al saldo. Averigua los parámetros que se necesita en el constructor. No debe visualizar nada en pantalla.

Crea en el método main() un objeto Saldo asignándole un valor inicial. Visualiza el saldo inicial. Crea varios hilos que compartan ese objeto Saldo. A cada hilo le debes dar un nombre y le asignamos una cantidad. Lanzamos los hilos y esperamos a que finalicen para visualizar el saldo final del objeto Saldo. Comprueba los resultados quitando synchronized del método de la clase