

Synchronized methods

```
public class SynchronizedMethodsOneObjectExample {
    private int count = 0;

    public synchronized void count() {
        count++;
    }

    public synchronized int getCount() {
        return count;
    }
}
```

Il lock è associato all'oggetto che invoca il metodo synchronized, quindi se 2 threads vogliono eseguire il metodo sincronizzato sullo stesso oggetto, uno dei 2 dovrà attendere che l'altro rilasci il lock

```
public class SynchronizedStaticsMethodsExample {
    private static int count = 0;

    public static synchronized void count() {
        count++;
    }

    public static synchronized int getCount() {
        return count;
    }
}
```

Se un metodo static è dichiarato synchronized, il lock è associato alla classe stessa, non a una particolare istanza della classe. Questo garantisce che solo un thread alla volta possa eseguire il metodo statico sincronizzato, indipendentemente dalle istanze della classe.

Synchronized Blocks

```
public class SynchronizedBlocksExample {
    private int count = 0;
    private final Object lock = new Object();

    public void count() {
        synchronized (lock) {
            count++;
        }
    }

    public int getCount() {
        synchronized (lock) {
            return count;
        }
    }
}
```

Riduce la sincronizzazione a una sezione specifica del codice, il lock è associato a un oggetto diverso dall'oggetto che invoca il metodo.