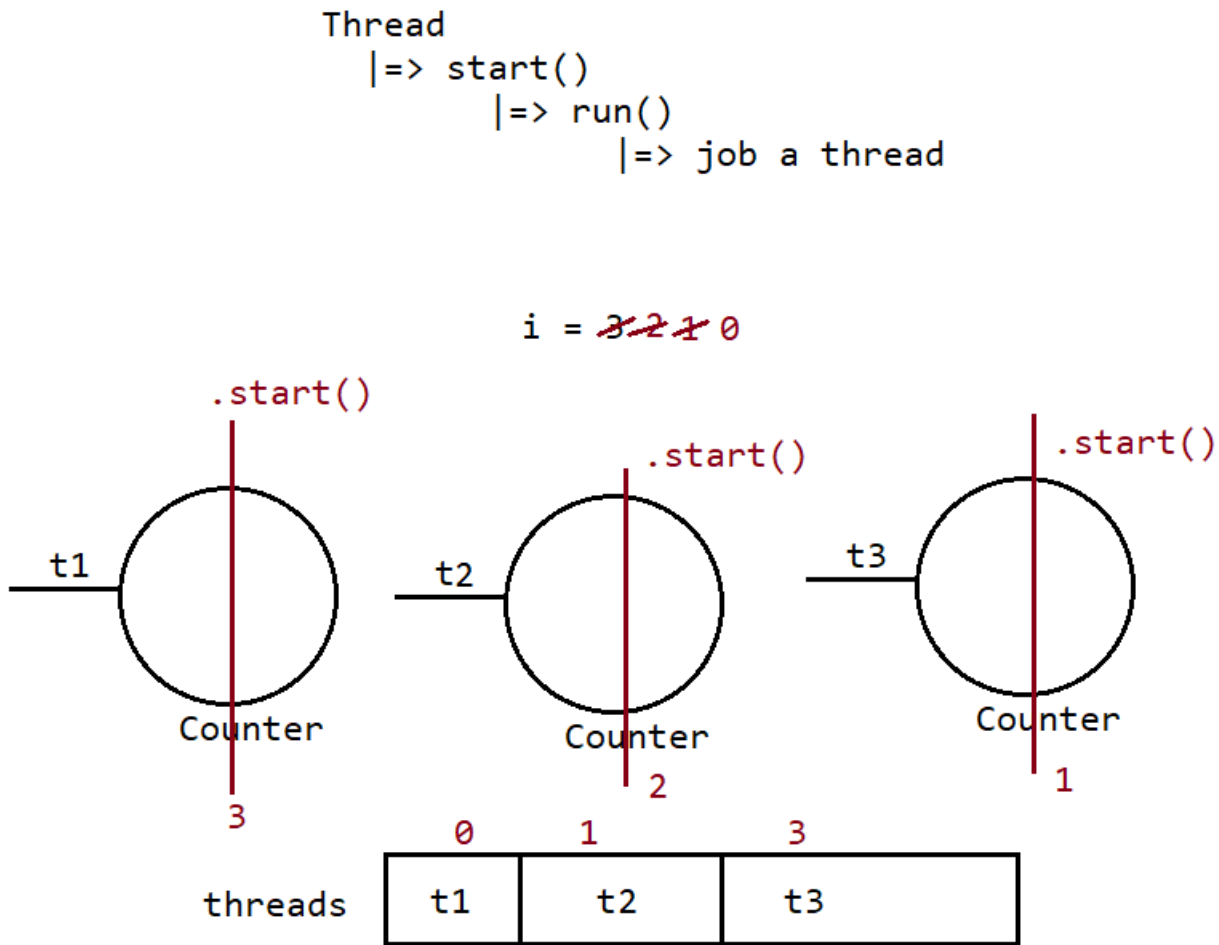```java
class Counter implements Runnable {
    private static int i = 3;
    @Override
    public void run() {
        System.out.print(i--);
    }
}

public class Test {
    public static void main(String[] args) {
        Thread t1 = new Thread(new Counter());
        Thread t2 = new Thread(new Counter());
        Thread t3 = new Thread(new Counter());
        Thread[] threads = {t1, t2, t3};
        for(Thread thread : threads) {
            thread.start();
        }
    }
}
```

Thread
|=> start()
    |=> run()
        |=> job a thread

i = ~~3 2 1~~ 0



```java
class ThreadDemo extends Thread {

    final StringBuffer sb1 = new StringBuffer();
    final StringBuffer sb2 = new StringBuffer();

    public static void main(String args[]) {
        final ThreadDemo h = new ThreadDemo();
        new Thread() {
            public void run() {
                synchronized(this) {
                    h.sb1.append("Java");
                    h.sb2.append("Thread");
                    System.out.println(h.sb1);
                    System.out.println(h.sb2);
                }
            }
        }.start();

        new Thread() {
            public void run(){
                synchronized(this) {
                    h.sb1.append("Mutithreading");
                    h.sb2.append("Example");
                    System.out.println(h.sb2);
                    System.out.println(h.sb1);
                }
            }
        }.start();
    }
```


Lock applied

output

Java Thread
ThreadExample Java MultiThreading