

## Java: Critical Code absichern

Um Race-Conditions zu vermeiden, muss dafür gesorgt werden, dass "Critical Code" nicht von mehreren Threads gleichzeitig ausgeführt wird. Ein Thread darf erst einen kritischen Codeabschnitt ausführen, wenn sich kein anderer Thread mehr im kritischen Codeabschnitt befindet.

### Synchronized Methode

```
public synchronized void methode(){  
    // CRITICAL CODE  
}
```

### Synchronized Block

```
synchronized(this){  
    // CRITICAL CODE  
}
```

### Lock

```
Lock lock = new ReentrantLock();  
lock.lock();  
  
// CRITICAL CODE  
  
lock.unlock();
```