



链滴

java 基础 | 验证 i++ 的线程不安全

作者: [xiaodaojava](#)

原文链接: <https://ld246.com/article/1572962141672>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



javaDEMO

本网站记录了最全的各种JavaDEMO,保证下载,复制就是可用的,包括基础的, 集合的, spring的, Mybatis的等等各种,助力你从菜鸟到大牛,记得收藏哦~~

<https://www.javastudy.cloud>

i++的线程不安全

i++是我们从开始学写代码时就一直在写的一个语句,也时常有前辈说过,i++是线程不安全的,我今天就来一起模拟下多个线程同时对i进行++操作

代码如下:

```
import java.util.concurrent.CountDownLatch;

public class IDemo {

    public static void main(String[] args) {
        // 用CountDownLatch来计数, 其主要有两个方法
        // 1是await, 使当前线程暂停
        // 2是countDown,使当前计数器减1, 如果计数器减为0, 则暂停的线程可以继续执行
        CountDownLatch latch = new CountDownLatch(1);
        // new了一个类, 里面持有为0的 i
        TestCounter counter = new TestCounter();
        // 定义了线程操作, 先调用latch的await方法等待计数器变成0
        // 然后执行counter.add方法, 让i++;
        Runnable runnable = () -> {
            try {
                latch.await();
                counter.add();
            }
        };
    }
}
```

```

        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    };
    // 循环1000次，新建1000个线程
    for (int i = 0; i < 1000; i++) {
        Thread t = new Thread(runnable);
        t.start();
    }
    // 这里调用latch.countDown方法，让计数器减为0，则所有暂停的线程都可以执行
    try {
        latch.countDown();
        // 主线程暂停3秒，让子线程可以执行完
        Thread.sleep(3000L);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    // 输出最后执行完i++后i的值
    System.out.println(counter.i);
}

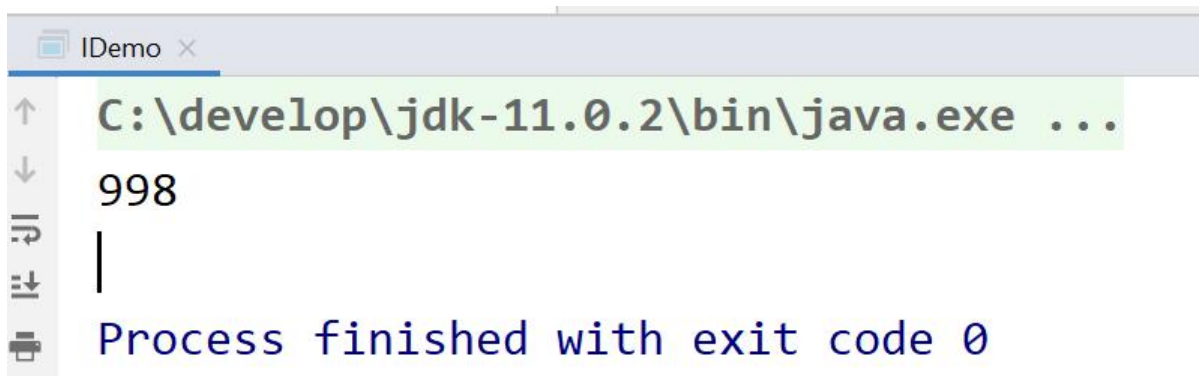
/**
 * 使用内部类的方式，封装和i++操作
 */
static class TestCounter {
    int i = 0;

    void add() {
        i++;
    }
}
}
}

```

输出结果如下：

多次运行会有不同的结果



```

IDemo x
C:\develop\jdk-11.0.2\bin\java.exe ...
998
|
Process finished with exit code 0

```

DEMO总评

i++线程不安全是一个老生常谈的问题，但真动手写起代码，还是有不少坑的，在这份dome中，有小伙伴会把countDownLatch设置为100，也是可以的，思路不一样。不管哪一种思路，能模拟出场就可以，关于CountDownLatch的用法，在其jdk源码中有示例，大家可以去看一看，加油吧，少年!