



链滴

多线程测试 n 个线程同时开始 并发辅助测试

作者: [shixiaoxiang](#)

原文链接: <https://ld246.com/article/1510908635140>

来源网站: 链滴

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

```

// 不多BB,上代码:==>
package com.baidu.test.junit;

import java.util.concurrent.CountDownLatch;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

import org.junit.runner.RunWith;
import org.springframework.boot.test.context.SpringBootTest;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

import com.taikang.Application;

/**
 * 多线程Junit单元测试
 * @author LiuDong
 *
 */
@RunWith(SpringJUnit4ClassRunner.class)
@SpringBootTest(classes = Application.class)
public class JunitThreadTest {

/**
 * 模拟100米赛跑,n名选手已经准备就绪,只等一声令下,当所有人到达终点时,比赛结束
 * @param args
 * @throws Exception
 */
public static void main(String[] args) throws Exception {
    // 选手数量
    final int num = 20;
    // 开始的倒数锁
    final CountDownLatch begin = new CountDownLatch(1);
    // 结束的倒数锁
    final CountDownLatch end = new CountDownLatch(num);
    // 十名选手
    final ExecutorService exec = Executors.newFixedThreadPool(num);

    for (int i = 0; i < num; i++) {
        final int no = i + 1;
        Runnable run = new Runnable() {
            // 重写run()方法
            @Override
            public void run() {
                try {
                    // 如果当前计数为零,则此方法立即返回
                    // 等待

```

```
        begin.await();
        // 可以写点循环控制每个运动运动员跑多少圈(长跑)
        Thread.sleep((long) (Math.random() * 10000));
        System.out.println("NO." + no + "arrived");
    } catch (InterruptedException e) {
        e.printStackTrace();
    } finally {
        // 每个选手到达终点时,end就减一
        end.countDown();
    }
}
};
exec.submit(run);
}
System.out.println("Game Start");
// begin减一,开始游戏
begin.countDown();
// 等待end变为0,即所有选手都到达终点
end.await();
System.out.println("Game Over");
exec.shutdown();
}
}
```