**班级： 软工11602 姓名： 漆鑫鑫 序号： 17**

**上机1 线程不安全**

1. **上机目的：**
2. 掌握利用eclipse了解线程应用和功能。
3. 程序调通，分析原因。
4. **主要上机仪器及材料 ：**

**Eclipse**

1. **掌握要点：**

了解线程并且实验代码异常和功能。

1. **上机内容及代码：**

**1，ThreadMain：**

**package com.captain.demo.threadNoSafe;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.concurrent.ArrayBlockingQueue;**

**import java.util.concurrent.BlockingQueue;**

**import java.util.concurrent.ExecutorService;**

**import java.util.concurrent.Executors;**

**/\*\***

**\* Created by captain on 2017/7/27.**

**\*/**

**public class ThreadMain {**

**public static void main(String[] args) {**

**//第一个的演示示例**

**/\*List<String> list = new ArrayList<>(1000);**

**for(int cnt=0;cnt<1000;cnt++){**

**list.add("abc");**

**}**

**ThreadSafeDemo1 demo1 = new ThreadSafeDemo1(list);**

**ThreadSafeDemo1 demo2 = new ThreadSafeDemo1(list);**

**Thread thread1 = new Thread(demo1);**

**Thread thread2 = new Thread(demo2);**

**thread1.start();**

**thread2.start();\*/**

**//第二个的演示示例**

**/\*List<String> list = new ArrayList<>();**

**ThreadSafeDemo2 demo1 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo2 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo3 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo4 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo5 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo6 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo7 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo8 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo2 demo9 = new ThreadSafeDemo2(list);**

**ThreadSafeDemo3 demo10 = new ThreadSafeDemo3(list);**

**new Thread(demo1).start();**

**new Thread(demo2).start();**

**new Thread(demo3).start();**

**new Thread(demo4).start();**

**new Thread(demo5).start();**

**new Thread(demo6).start();**

**new Thread(demo7).start();**

**new Thread(demo8).start();**

**new Thread(demo9).start();**

**new Thread(demo10).start();\*/**

**}**

**//使用线程池**

**ExecutorService fixedThreadPool = Executors.newFixedThreadPool(9);**

**BlockingQueue<String> list = new ArrayBlockingQueue<String>(100);**

**for (int i = 0; i < 9; i++)**

**{**

**ThreadSafeDemo2 demo = new ThreadSafeDemo2((List<String>) list);**

**fixedThreadPool.execute(demo);**

**}**

**ThreadSafeDemo3 demo10 = new ThreadSafeDemo3((List<String>) list);**

**new Thread(demo10).start();**

**fixedThreadPool.shutdown();**

**}**

**2，ThreadSafeDemo1：**

**package com.captain.demo.threadNoSafe;**

**import java.util.Iterator;**

**import java.util.List;**

**import java.util.Random;**

**import java.util.concurrent.TimeUnit;**

**/\*\***

**\* 线程安全演示类**

**\* Created by captain on 2017/7/27.**

**\*/**

**public class ThreadSafeDemo1 implements Runnable{**

**private List<String> list;**

**public ThreadSafeDemo1(List<String> list){**

**this.list = list;**

**}**

**@Override**

**public void run() {**

**Random random = new Random();**

**System.out.println("线程："+ random.nextInt());**

**synchronized(this){**

**Iterator<String> iterator = list.iterator();**

**int cnt = 0;**

**while(iterator.hasNext()){**

**iterator.next();**

**iterator.remove();**

**cnt++;**

**try {**

**TimeUnit.MILLISECONDS.sleep(10);**

**} catch (InterruptedException e) {**

**e.printStackTrace();**

**}**

**}**

**System.out.println("移除元素的数量："+cnt);**

**}**

**}**

**}**

1. **ThreadSafeDemo2：**

**package com.captain.demo.threadNoSafe;**

**import java.util.List;**

**/\*\***

**\* 线程安全演示类**

**\* Created by captain on 2017/7/27.**

**\*/**

**public class ThreadSafeDemo2 implements Runnable{**

**private List<String> list;**

**public ThreadSafeDemo2(List<String> list){**

**this.list = list;**

**}**

**@Override**

**public void run() {**

**while(true){**

**list.add("版权归作者所有，任何形式转载请联系作者。作者：KY主创们（来自豆瓣来源：https://www.douban.com/note/630084526/不过，专门研究“人猫交流（cat-human communication)”的学者认为这种印象也不完全是精确的——是的，这是一个真实存在的研究学科。研究宠物行为的学科曾经一度被取消，因为人们认为这种研究的本质是一种拟人论（anthropomorphism），但现在研究这个学科的学者已经不再这样认为了（即宠物行为研究有其自身本质的意义）。");**

**System.out.println("添加一个后尺寸："+list.size());**

**}**

**}**

**}**

1. **ThreadSafeDemo3：**

**package com.captain.demo.threadNoSafe;**

**import java.util.List;**

**/\*\***

**\* 线程安全演示类**

**\* Created by captain on 2017/7/27.**

**\*/**

**public class ThreadSafeDemo3 implements Runnable{**

**private List<String> list;**

**public ThreadSafeDemo3(List<String> list){**

**this.list = list;**

**}**

**@Override**

**public void run() {**

**while(true){**

**if(list.size()>0){**

**list.remove(0);**

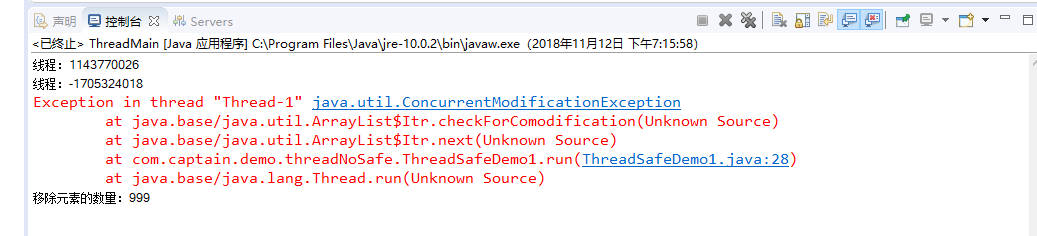
**System.out.println("删除一个后尺寸："+list.size());**

**}**

**}**

**}**

**}**



**五、心得体会：**通过上述测试，我发现，存在成员变量的类用于多线程时是不安全的，不安全体现在这个成员变量可能发生非原子性的操作，而变量定义在方法内也就是局部变量是线程安全的。多个线程之间是不能直接传递数据进行交互的，它们之间的交互只能通过共享变量来实现。多个线程执行时，CPU对线程的调度是随机的，我们不知道当前程序被执行到哪步就切换到了下一个线程。