班级：软工11602 姓名：袁孟 序号： 32

上机1 线程调试

**一、上机目的**

1、了解线程的基本过程；

2、对三种情况的线程进行各种调试；

3、调试三种线程的错误；

**二、上机准备**

软件：eclipse；

**三、上机内容**

调试代码：

**ThreadMain.java:**

package com.ym.test;

import java.util.ArrayList;

import java.util.List;

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.run();

thread2.run();

//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();\*/

//第三个的演示示例

/\* List<String> list = new ArrayList<>(10);

for(int cnt=0;cnt<10;cnt++){

list.add("abc");

}

ThreadSafeDemo3 demo1 = new ThreadSafeDemo3(list);

ThreadSafeDemo3 demo2 = new ThreadSafeDemo3(list);

Thread thread1 = new Thread(demo1);

Thread thread2 = new Thread(demo2);

thread1.start();

thread2.start();\*/

}

}

**ThreadSafeDemo1.java:**

package com.ym.test;

import java.util.Iterator;

import java.util.List;

import java.util.Random;

import java.util.concurrent.TimeUnit;

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);

}

}

}

**ThreadSafeDemo2.java:**

**package** com.ym.test;

**import** java.util.List;

**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());

}

}

}

**ThreadSafeDemo3.java:**

**package** com.ym.test;

**import** java.util.List;

**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());

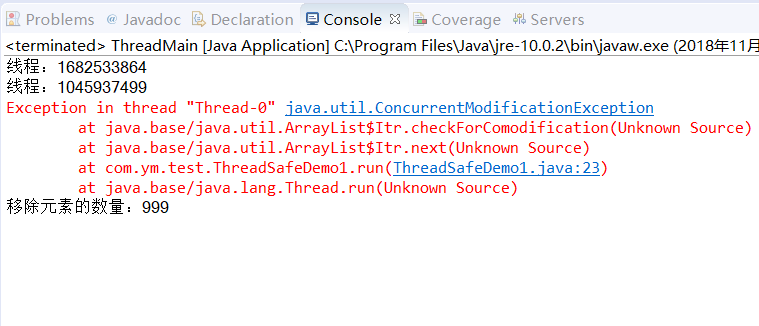
}

}

}

}

异常：



异常修改：

将ThreadSafeDemo1里的**synchronized**(this)改为**synchronized**(list)

原因：当两个并发线程访问同一个对象object中的这个synchronized(this)同步代码块时，一个时间内只能有一个线程得到执行。另一个线程必须等待当前线程执行完这个代码块以后才能执行该代码块。改为synchronized(list)使两个线程进行同步，所以不会报错。

**四、上机心得**

此次上机，我对线程的知识有了更深的了解，对多线程之间的联系，和执行顺序了解。明白了run()方法和start()方法的区别。