链滴

实现主线程与子线程交替执行

作者：Saber
原文链接：https://ld246.com/article/1517217328398
来源网站：链滴

原题

子线程循环10次，接着主线程循环100，接着又回到子线程循环10次，接着再回到主线程又循环100如此循环50次，请写出程序

实现

其实创建子线程本来只需要在Main函数内new Thread即可，但为了遵循阿里巴巴的编码规约，尝试用了线程池来创建子线程。

实现线程工厂

```
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import java.util.concurrent.ThreadFactory;

/**
 * @author Saber
 */
public class MyThreadFactory implements ThreadFactory {

    /**
     * 计数器
     */
    private int counter;
    /**
     * 线程名
     */
    private String name;
    /**
     * 状态
     */
    private List<String> stats;

    MyThreadFactory(String name) {
        counter=0;
        this.name=name;
        stats=new ArrayList<>();
    }

    @Override
    public Thread newThread(Runnable r) {
        Thread thread=new Thread(r,name+"Thread"+counter);
        counter++;
        stats.add(String.format("Created thread %d with name %s on %s \n", thread.getId(), thre
d.getName(), new Date()));
        return thread;
    }

    public String getStats(){
```

```
        StringBuilder buffer = new StringBuilder();
        for (String stat : stats) {
            buffer.append(stat);
        }
        return buffer.toString();
    }
}
```

实现主线程与子线程的循环方法

这里将主线程与子线程的循环方法拿出来放到一个类中是为了共享bShouldSub信号量，以实现交替斥执行

```
/**
 * @author Saber
 */
public class Business {
    /**这里相当于定义了控制该谁执行的一个信号灯*/
    private boolean bShouldSub = true;

    synchronized void mainThread(int i)
    {
        if(bShouldSub) {
            try {
                this.wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        for(int j=0;j<100;j++)
        {
            System.out.println(Thread.currentThread().getName() + ":i=" + i +",j=" + j);
        }
        bShouldSub = true;
        this.notify();

    }

    synchronized void subThread(int i)
    {
        if(!bShouldSub) {
            try {
                this.wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        for(int j=0;j<10;j++)
        {
            System.out.println(Thread.currentThread().getName() + ":i=" + i +",j=" + j);
```

```
        }
        bShouldSub = false;
        this.notify();
    }
}
```

实现主方法

```
import java.util.concurrent.ArrayBlockingQueue;
import java.util.concurrent.ThreadPoolExecutor;
import java.util.concurrent.TimeUnit;

/**
 * @author Saber
 */
public class ThreadTest {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        new ThreadTest().init();

    }

    private void init()
    {
        final Business business = new Business();
        MyThreadFactory myThreadFactory=new MyThreadFactory("子线程");
        ThreadPoolExecutor threadPoolExecutor=new ThreadPoolExecutor(
                4,
                10,
                200,
                TimeUnit.MILLISECONDS,
                new ArrayBlockingQueue<>(5),
                myThreadFactory);

        /*
          用线程池创建了一个子线程，来运行subThread方法，子线程的方法运行一次后把自己wait
          此时，mainThread的循环启动
          mainThread运行完一次后把自己wait
          subThread得bShouldSub锁，继续执行
          ...
         */
        threadPoolExecutor.execute(() -> {
            for(int i=0;i<50;i++)
            {
                business.subThread(i);
            }
        }
        );

        for(int i=0;i<50;i++)
        {
```

```
            business.mainThread(i);
        }
    }
}
```

原文链接：实现主线程与子线程交替执行