

题目描述

6-1 jmu-Java-07多线程-交替执行 分数 15

全屏浏览题目

作者 郑如滨 单位 集美大学

有一连串任务，需要两个线程交替执行。线程1执行完任务1后，线程2才能执行任务2，接下来线程1执行任务1，如此交替执行下去。直到所有任务执行完毕。

定义 `Repo` 类代表任务仓库，使用字符串代表任务。该类拥有：

构造方法：

```
/*将传递进来的字符串以空格分隔分解为多个不同的任务，并存储起来。如"1 2 3 4 5 6"被分解成6个任务1,2,3,4,5,6*/
public Repo(String items) {
}
```

方法：

```
int getSize(); //返回Repo包含的任务数量。注意：完成任务的时候，需要将任务删除。
//其他完成任务的方法
```

定义 `Worker1` 与 `Worker2` 类，代表两个交替完成任务的类，可以从 `Repo` 对象中获取任务。

main方法如下：

```
public class Main {
    public static void main(String[] args) throws InterruptedException {
        Scanner sc = new Scanner(System.in);
        Repo repo = new Repo(sc.nextLine());
        Thread t1 = new Thread(new Worker1(repo));
        Thread t2 = new Thread(new Worker2(repo));
        t1.start();
        Thread.yield();
        t2.start();
        sc.close();
    }
}
```

输入样例

1 2 3 4 5 6 7 8 9

输出样例

Thread-0 finish 1
Thread-1 finish 2
Thread-0 finish 3
Thread-1 finish 4
Thread-0 finish 5
Thread-1 finish 6
Thread-0 finish 7
Thread-1 finish 8
Thread-0 finish 9

裁判测试程序：

```
1 /*Repo代码*/
2
3 /*Worker1代码*/
4
5 /*Worker2代码*/
6
7 /*系统已有代码，无需关注*/
8
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

Java (javac)

   

痕 

```
1 import java.util.*;
2 class Repo {
3     ... Queue<String> queue = new LinkedList<>();
4     ... public Repo(String items) {
5         ... String[] tasks = items.split(" ");
6         ... queue.addAll(Arrays.asList(tasks));
7     }
8
9     ... int getSize(){
10        ... return queue.size();
11    }
12
13    ... public synchronized boolean getTask() throws InterruptedException {
14        ... this.notify(); // 唤醒一个线程
15        ... if(getSize() > 0){ // 如果当前还有可执行的任务
16            ... String work = queue.remove();
17            ... System.out.println(Thread.currentThread().getName() + " finish " + work);
```

< 上一题

查看上次提交

提交本题作答

下一题 >