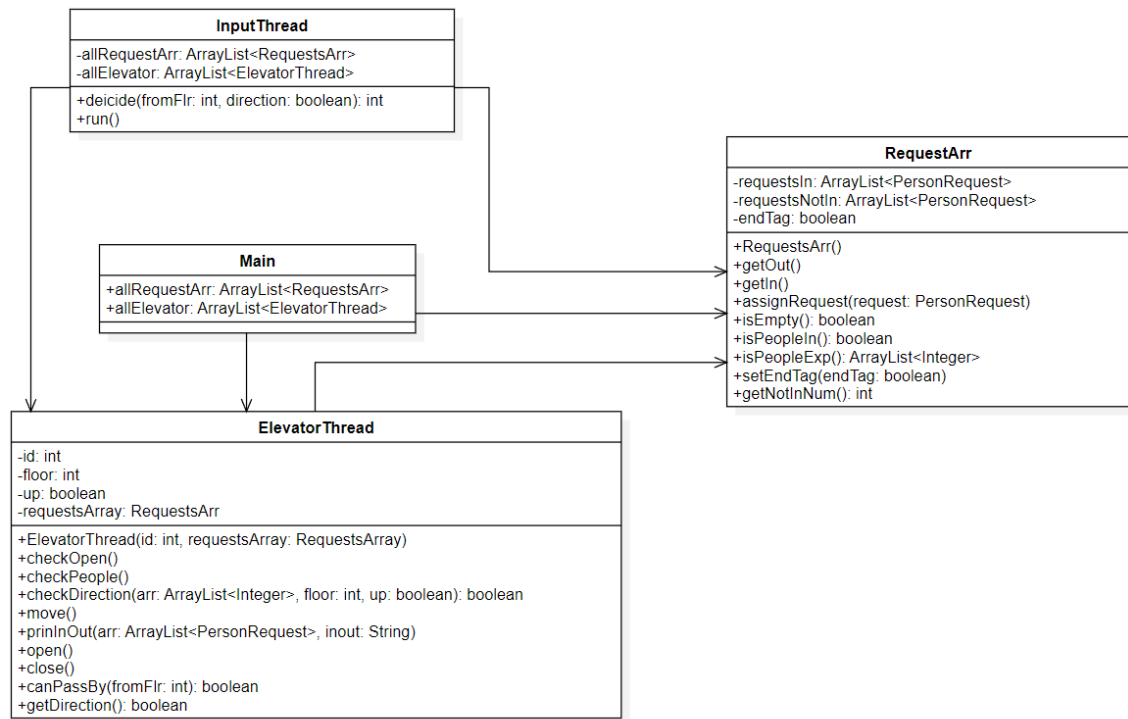


# BUAA\_OO 第二单元总结

## 第五次作业

### 架构



### 锁的设置

使用了生产者-消费者模型

共享数据为RequestArr，所以一切操作RequestArr中属性requestsIn，requestsNotIn，endTag的操作都设置为synchronized方法。

### 调度器

这次作业中并没有单独写调度器类，只是在InputThread中每得到一个请求就用decide操作分析有没有正好经过新请求fromFlr的电梯，如果能找到就把请求分配给这个电梯的requestArr。

基本实现如下：

```
PersonRequest request = eleInput.nextPersonRequest();
if (request == null) {
    for (int i = 0; i < allRequestArr.size(); i++) {
        allRequestArr.get(i).setEndTag(true);
    }
    break;
}
int findEle = decide(request.getFromFloor(),
    (request.getToFloor() - request.getFromFloor() > 0));
allRequestArr.get(findEle).assignRequest(request);
```

# BUG和分析

这次作业中有性能不佳的地方，但是没有找到bug。

第五次作业只要求实现一个拥有初始六部电梯在1楼到11楼之间运行的电梯系统，比较简单（相比后两次作业），也是我强测分数最高的一次。

## 第六次作业

### 分析

这次作业相比第一次作业增加了两个功能：

- 添加电梯请求中包含**新电梯的ID和初始楼层**，程序需要将这部电梯加入到电梯系统的调度中。
- 电梯日常维护功能，给出要维护的电梯ID，**则这部电梯需要尽快**将电梯内所有乘客放出**然后**退出电梯调度系统，**程序还需要进行必要的**重新调度来满足这些乘客的请求。

我的解决办法：

- 为了让乘客可以退回，我写了一个WaitingRequests类的waitingRequests实例，被inputThread和ElevatorThread共享。
  - 由于它是共享资源，因此也在访问的时候加锁以保证线程安全。
  - 同时，只有waitingThread不会再有输入（即不会得到InputThread的输入也不会得到ElevatorThread因为maintain退回的请求，才可以向电梯线程输入**处理完自己现有的请求就可以结束线程**的信号），也就是说，必须要调整线程结束的判断条件。
- 增加电梯比较简单，直接在new一个新电梯加入allElevator中即可。

## BUG

### bug1

之所以在架构之前新分享bug，是因为在这一单元中强测出来的bug与架构息息相关，我不得不改变架构，增加一个类RealInputThread才解决了bug。

该bug为RTLE，强测允许的最大事件为105s，我的程序用了110s才把乘客送到目的地。bug出在处理请求上。

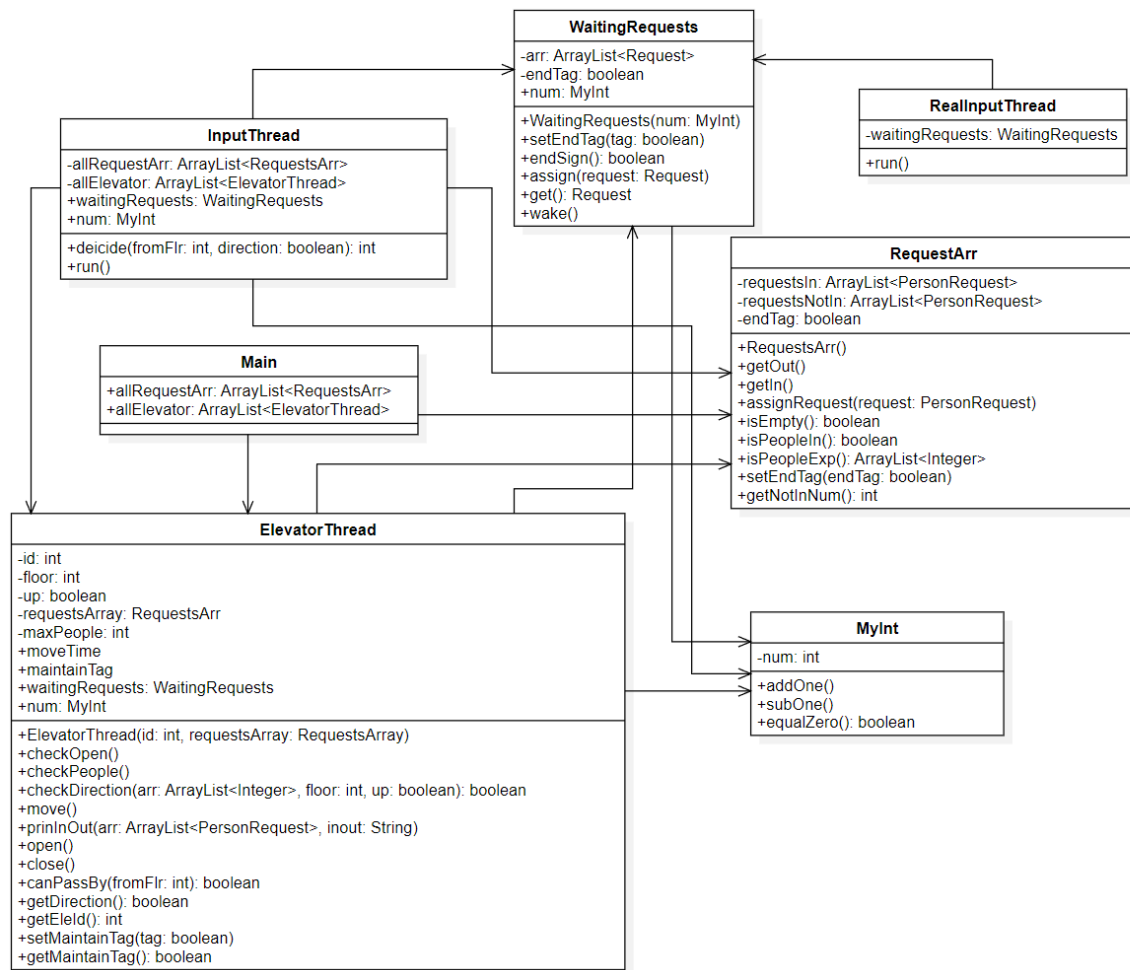
原调度代码和第一次作业一样,我将问题以注释形式标出

```
PersonRequest request = eleInput.nextPersonRequest();
//先读入下一个请求，等于是投放一个请求处理一个请求，当程序中有因为maintain返回来得请求之后，每次处理的请求并不是投放的请求，如果出现某个投放的请求和前一个投放的请求之间间隔时间非常长的情况，就会导致很早之前投放的请求延迟了很长时间才被处理，会导致性能很差，甚至会RTLE
    if (request == null) {
        for (int i = 0; i < allRequestArr.size(); i++) {
            allRequestArr.get(i).setEndTag(true);
        }
        break;
    }
    int findEle = decide(request.getFromFloor(),
        (request.getToFloor() - request.getFromFloor() > 0));
    allRequestArr.get(findEle).assignRequest(request);
```

## bug2

有些请求被maintain回去的时候该分配的线程已经结束了，因此设置了一个MyInt类处理一个线程安全的int被ElevatorThread，waitingQueue共享，表示正在Maintain的线程的数量，判断waitingQueue不会再有新的请求加入除了要使得InputThread读入Null之外还要求没有正在维护的线程。

## 架构



## 锁的设置

这次作业增加了共享资源WaitingRequests，和num表示正在maintain中的电梯数量。还是和上一次作业中一样，为共享资源加锁，以保证线程安全。

## 调度器

这次作业沿用了上次作业的调度器，唯一改动的点是，不分配给maintainTag为true的电梯。

## 第七次作业

### 分析


这次作业增加的内容有

- 电梯系统调度参数
  - 任意楼层x 处于**服务中的电梯**的最大数量  $\max x = 4$ ;
  - 任意楼层x 处于**服务中的只接人的电梯**的最大数量  $n_x = 2$
- 电梯可达性（其实就是在哪些层可以开门）

任意时刻，电梯系统**保证能完成任意请求**，即对于**任意请求**，**总能找出一个楼层序列**，**序列开头为起点层，结尾为终点层**，乘客能够通过使用电梯系统依次经过序列中各楼层直到到达终点层

我做出的增改有：

- 为电梯增加属性：access表示可达性
- 增加共享资源MyInt[] serviceNum和 MyInt[] pickNum来表示每一层处于服务中的电梯和只接人电梯的数量来满足电梯系统调度参数的要求
- 增加了一个类MyPersonRequest，这个类相比于PersonRequest的区别就是增加了属性Routes，且getFromFlr()和getToFloor()返回的是当前这一程的起始和目的楼层而不是这个乘客的整个请求，增加了属性index来表示乘客目前进行的是换乘中的哪一层。
- Routes类用floors[]和eles[]来表示构造器为这个电梯分配的请求。具体表示如下图。

 186c65d01769adb675f5b06d6cb0fbb

这个例子中乘客乘allElevator[2]从1楼到5楼，再乘allElevator[3]从5楼到7楼，并且目前正在经历从5楼到7楼。

(感觉这个架构不够优雅，但是时间紧迫，笔者当时也想不到更好的表示方法了)

- 写了一个调度器负责静态分配请求，将在调度器一节详细解释。
- 每当out了一个请求，就检查这个请求的index和Route中数组长度的关系，检查是否乘客还有下一乘，如果有的话就把下一乘的请求分配出去。

## 架构

<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>ElevatorThread</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>access</div><div>int</div></div></div></div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>maintainTag</div><div>boolean</div></div></div></div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>eled</div><div>int</div></div></div></div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>maintainTag</div><div>boolean</div></div></div></div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>direction</div><div>boolean</div></div></div></div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>access</div><div>int</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>checkPeople()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>checkDirection</div><div>(ArrayList&lt;Integer&gt;, int, boolean)</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>move()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>printlnOut</div><div>(ArrayList&lt;MyPersonRequest&gt;, String)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>checkOpen()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>close</div><div>(boolean)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>canPassBy</div><div>(int)</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>open</div><div>(boolean)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>run()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>maintain()</div><div>void</div></div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>ElevatorThread</div><div>(int, RequestsArr, WaitingRequests, MyInt, MyInt)</div><div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>ElevatorThread</div><div>(int, RequestsArr, int, int, int, WaitingRequests)</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>	<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>RequestsArr</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>endTag</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>notInNum</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>dispatchTag</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>peopleIn</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>endTag</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>empty</div><div>boolean</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>inLastRide()</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>getOut</div><div>(int)</div><div>ArrayList&lt;MyPersonRequest&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>checkDeirection</div><div>(int, int, boolean)</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>returnIn</div><div>(int)</div><div>ArrayList&lt;MyPersonRequest&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>getIn</div><div>(int, boolean, int)</div><div>ArrayList&lt;MyPersonRequest&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>wake</div><div>()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>assignRequest</div><div>(MyPersonRequest)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>returnNotIn</div><div>()</div><div>ArrayList&lt;MyPersonRequest&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>isPeopleExp</div><div>()</div><div>ArrayList&lt;Integer&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>RequestsArr</div><div>()</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>
--	---

<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Dispatcher</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>decide</div><div>(int, boolean, ArrayList&lt;Integer&gt;)</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>hasAccess</div><div>(int, int, int)</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>findRoutes</div><div>(int, int)</div><div>ArrayList&lt;Accessible&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>dispatch</div><div>(MyPersonRequest)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>findRoutesByTwoEle</div><div>(int, int)</div><div>ArrayList&lt;Accessible&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>findBestRoutes</div><div>(ArrayList&lt;Accessible&gt;, int, int)</div><div>Routes</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>findIntersectFloor</div><div>(int, int, int, int)</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>floorHasAccess</div><div>(int, int)</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>setDispatcher</div><div>(ArrayList&lt;RequestsArr&gt;, ArrayList&lt;ElevatorThread&gt;)</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>findRoutesBy3</div><div>(int, int)</div><div>ArrayList&lt;Accessible&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>decideRoutes</div><div>(int, int)</div><div>Routes</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>findRoutesByOneEle</div><div>(int, int)</div><div>ArrayList&lt;Accessible&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>getEleAccess</div><div>(int)</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>dfsFindRoutes</div><div>(int, int, int, int)</div><div>void</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Dispatcher</div><div>()</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>	<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>MyPersonRequest</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>personID</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>toFloor</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>fromFloor</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>realDst</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>personID</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>ele</div><div>int</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>addIndex</div><div>()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>toString</div><div>()</div><div>String</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>continueRequest</div><div>()</div><div>boolean</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>MyPersonRequest</div><div>(PersonRequest)</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>	<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Accessible</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>ele</div><div>ArrayList&lt;Integer&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>toString</div><div>()</div><div>String</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>firstIndex</div><div>()</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>addIndex</div><div>(int)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>lastIndex</div><div>()</div><div>int</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Accessible</div><div>(int, int, int)</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Accessible</div><div>()</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Accessible</div><div>(int, int)</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Accessible</div><div>(int)</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>	<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Routes</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>floors</div><div>ArrayList&lt;Integer&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>eles</div><div>ArrayList&lt;Integer&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>floors</div><div>ArrayList&lt;Integer&gt;</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>eles</div><div>ArrayList&lt;Integer&gt;</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>addEle</div><div>(int)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>toString</div><div>()</div><div>String</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>addFlr</div><div>(int)</div><div>void</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Routes</div><div>()</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>
---	---	---	--

<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>MyInt</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>num</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>num</div><div>int</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>subOne</div><div>()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>equalZero</div><div>()</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>toString</div><div>()</div><div>String</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>addOne</div><div>()</div><div>void</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>MyInt</div><div>(int)</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>MyInt</div><div>()</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>	<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>WaitingRequests</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>endTag</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>endTag</div><div>boolean</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>assign</div><div>(Request)</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>get</div><div>()</div><div>Request</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>wake</div><div>()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>endSign</div><div>()</div><div>boolean</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>WaitingRequests</div><div>(MyInt)</div><div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>	<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>InputThread</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>run</div><div>()</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>allInLastRide</div><div>()</div><div>boolean</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>InputThread</div><div>(ArrayList&lt;RequestsArr&gt;, ArrayList&lt;ElevatorThread&gt;)</div><div></div></div></div></div></div></div></div></div></div></div>	<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>RealInputThread</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>run</div><div>()</div><div>void</div></div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>RealInputThread</div><div>(WaitingRequests)</div><div></div></div></div></div></div></div></div></div>
--	---	---	--

<div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Main</div><div></div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>main</div><div>(String[])</div><div>void</div></div></div></div><div><div><div><div><div><div></div><div></div></div><div><div><span></span></div><div><span></span></div></div></div><div><div><div>Main</div><div>()</div><div></div></div></div></div></div></div></div></div></div></div>
--

## 调度器

我的调度器寻找路径采用dfs算法。

```

public static void dfsFindRoutes(int fromFlr, int toFlr, int curNum, int num) {
    //num代表换乘几次
    //curNum代表正在寻找第几段路的电梯
    if (findFlag == 1) {
        return;
        //已经找到一次之后就不找了
    }
    if (curNum == num) {
        //找到
        /*for(int k = 0;k < num;k++){
            system.out.println(accessing[k]);
        }*/
        findFlag = 1;
    }
}

```

```

        return;
    }
    //找第curNum段
    if (curNum == 0) {
        for (int i = 0; i < allElevator.size(); i++) {
            if (allElevator.get(i).getMaintainTag()) {
                continue;
            }
            if (floorHasAccess(allElevator.get(i).getAccess(), fromFlr)) {
                //第一段需要和可以经过初始楼层
                accessing[curNum] = i;
                dfsFindRoutes(fromFlr, toFlr, curNum + 1, num);
            }
            if (findFlag == 1) {
                return;
            }
        }
    } else if (curNum == num - 1) {
        for (int i = 0; i < allElevator.size(); i++) {
            if (allElevator.get(i).getMaintainTag()) {
                continue;
            }
            if (i == accessing[curNum - 1]) {
                //连续两座一样的电梯是没有必要的
                continue;
            }
            int intersectFlag = getEleAccess(i) &
getEleAccess(accessing[curNum - 1]);
            if (floorHasAccess(allElevator.get(i).getAccess(), toFlr) &&
(intersectFlag != 0)) {
                //最后一段需要可以经过终点楼层
                accessing[curNum] = i;
                dfsFindRoutes(fromFlr, toFlr, curNum + 1, num);
            }
            if (findFlag == 1) {
                return;
            }
        }
    } else {
        for (int i = 0; i < allElevator.size(); i++) {
            if (allElevator.get(i).getMaintainTag()) {
                continue;
            }
            if (i == accessing[curNum - 1]) {
                //连续两座一样的电梯是没有必要的
                continue;
            }
            int intersectFlag = getEleAccess(i) &
getEleAccess(accessing[curNum - 1]);
            if (intersectFlag != 0) {
                accessing[curNum] = i;
                dfsFindRoutes(fromFlr, toFlr, curNum + 1, num);
            }
            if (findFlag == 1) {
                return;
            }
        }
    }
}

```

```
}
```

## 遇到的bug

- 线程结束问题，需要防止给下一程分配请求的时候那个线程已经结束了，即必须要**增加条件判断线程的结束**
- 在我写的结构中，电梯时先放出请求再打印出门信息，并在放出请求的同时投放下一程请求，在极端的情况会导致人还没有出来就已经打印了进入下一程的消息。在debug的过程中，我改为在打印信息之后再投放请求，同时设置tag标志中间的空挡为“正在等待重新投放请求”防止线程结束。

## 心得体会

这一单元稳定与异变的内容分别有：

- 稳定的内容：
  - 电梯与对应的请求队列
  - 单个电梯对自己的队列的处理方法（用look策略在三次作业中都没有改变）
- 异变的内容：
  - 对读入请求的存储
  - 分配请求的方法去

在这一单元中我主要遇到的问题有

- 对电梯的逻辑不清晰
- 对线程结束的条件设置得不够好，要么会导致RTLE（线程一直不结束），要么会导致投放请求的时候线程已经结束（乘客无法到达目的地）

本着能少一点麻烦就少一点麻烦的原则我在第五次作业和第六次作业的时候并没有单独写出调度器，也没有考虑是否能用多个线程去提高性能，第五次作业只写了4个类，第六次作业只写了6个类，但是由于第七次作业的要求十分复杂，被迫增加到了11个类，在第七周常常十分担心写不完作业，压力很大，到作业提交的ddl还在debug。虽然没有像上个单元一样出现强测分数太差进不了互测的情况，但是第七次作业有很多bug没有被发现导致强测分数不高。

就像老师上课的时候说的那样，在得到一份需求说明之后应该花一天的时间在纸上设计好结构之后再开始写代码，可惜一方面由于时间的紧迫，另一方面由于在纸上我并不能考虑周全、不能确定某个设计是否真的可行，我只设计到了自己还不太明白的地步就开始写代码，而且在写第七次作业的时候自己写出来的东西乱到自己都不太看得懂。在第一次写完代码之后又重新看、debug的时候才理清了自己的架构。