**电梯调度**

1152703 方志晗 软工2班

**要解决的问题：**让用户等待电梯的时间最短；发生紧急情况可紧急制动电梯；可生成调度日志并可直接查看；可以手动开关电梯门；可以满足搭便车的要求；实现与图书馆电梯类似的UI和操作；

**设计思路来源：**模拟学校电梯的调度进行UI设计；主要实现外部查看的UI；因为楼层和电梯数可选，为了实现节目统一性，内部按钮集成到菜单栏（包括紧急制动，电梯门的开关）；

**调度算法思路：**

设置记录电梯当前状态（上行，停靠，下行，制动，开门）；

根据电梯状态计算时间：

1. 若电梯正在停靠——计算当前楼层与请求楼层的差值为需要等待时间；

2．若电梯正在上行——考虑两种情况，若电梯当前楼层低于请求层，直接计算差值为等待时间；若电梯当前楼层高于请求楼层，计算**到达目的层的差值**+目的层到**请求层的差值**+**开门所需时间**（开门时间为2秒，可换算成上升两层的时间） 为等待时间；

3. 若电梯正在下行——考虑两种情况，若电梯当前层高于请求层，直接计算差值为等待时间；若电梯当前楼层低于请求层，计算**到达目的层的差值**+**目的层到请求层的差值**+**开门所需时间** 为等待时间；

4. 若电梯被制动或者开门——不参与调度；

依据上述算法计算各个电梯需要等待的时间，比较各个电梯的等待时间长短。等待时间最短的电梯优先调度。

**关键API ：**

javax.swing.\*;（主要界面实现）

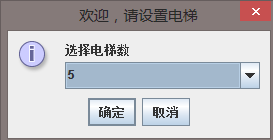
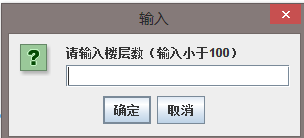
java.awt.\*;（使用其中的Layout和菜单）

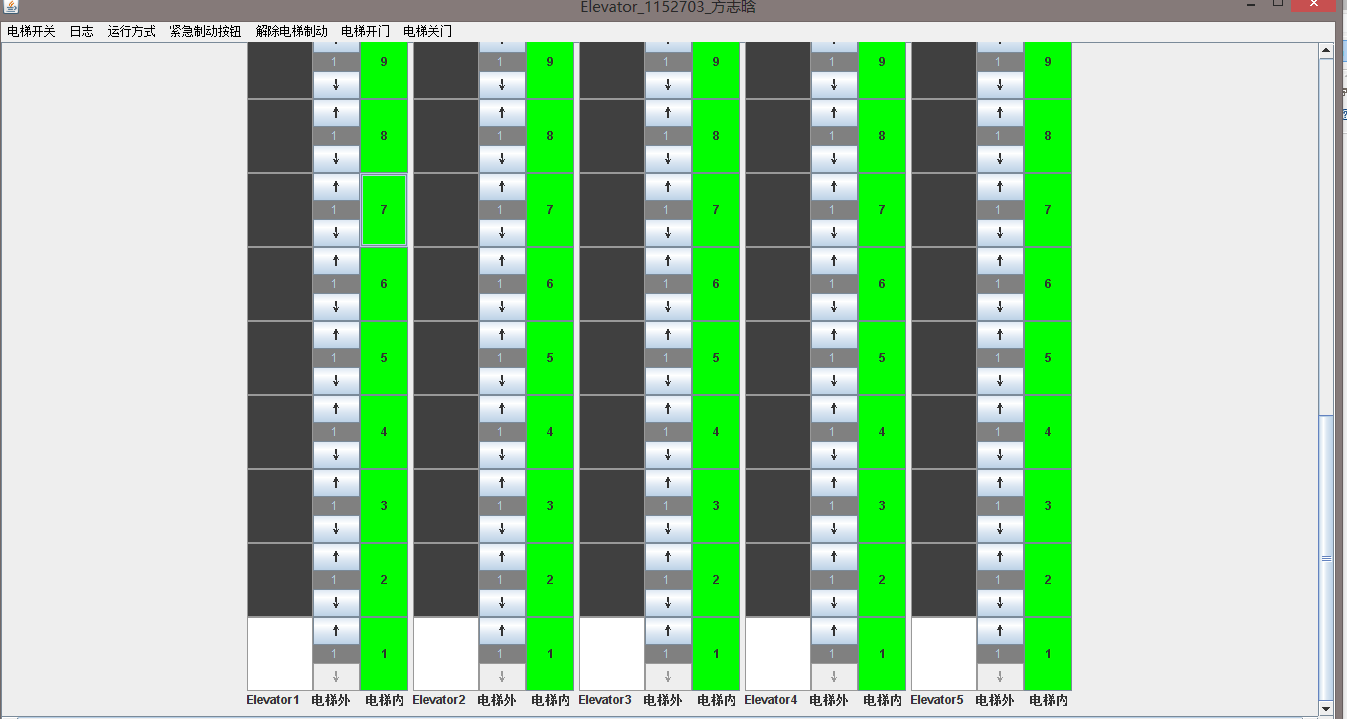
java.awt.event.ActionEvent;

java.awt.event.ActionListener;

java.io.IOException;

**界面：**



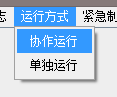
**基本功能：**

以等待时间最短为调度方式，支持紧急制动，解制动；开关门等；支持搭便车模式，即电梯状态（上升，下降）要去比请求层更高或更低层，此时若请求是和电梯状态一致，则可以在该层停。若状态相反则不停；加入滚动条方便查看电梯；内部选择按钮可实现当前电梯到达指定层；

 内部选择按钮 集成在电梯旁边。

**附加功能实现：**

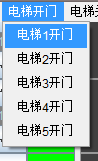
运行方式可选：协作运行与单独运行；协作运行运用调度算法进行调度，以等待时间最少为标准；考虑到了电梯开门与制动状态不可调度；单独运行遵循线程单独进行，不采用调度算法。



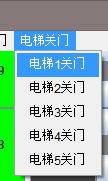
可控制电源开关：电源关闭后，电梯停止，一切请求无效，电梯停在当前楼层等待电源开启。



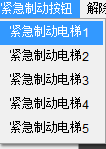
开门按钮：可实现停靠时的手动开门操作，非停靠状态不可操作，内部按钮，集成到菜单；



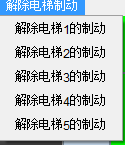
关门按钮：可实现开门后的关门，内部按钮，集成到菜单；



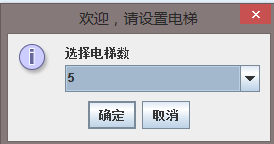
紧急制动功能：可设置某个电梯紧急停靠在当前楼层，紧急停靠后为不可工作状态，不参加协作调度和单独调度；

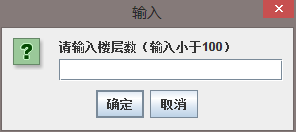


解除紧急制动：可解除制动状态，电梯重新被启用；

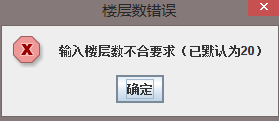


可设置电梯层数和楼层数：启动电梯开始会出现楼层设置和电梯数目设置的选择框；为了防止线程开启过多，电梯数为限定可选1-8；楼层数可最大到100；

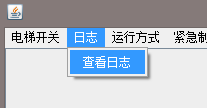
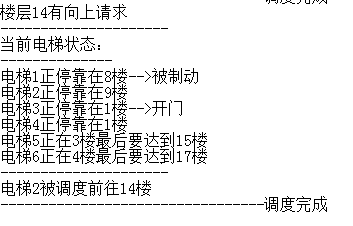




当输入错误时，自动设置默认置20层



自动生成调度日志：调度日志记录请求信息，请求时各个电梯的状态，状态细化到从当前楼层到要到楼层，可显示是否制动或门被打开，这两种状态不参加协作调度；可在当前页面查看调度日志；可通过日志查看调度的准确性；单独运行状态不生成日志；下面是自己测试的截图：

****

**使用说明：**

同调度时，所有上下请求的**外部按钮**功能相同，表示该层有请求。具体调度哪台电梯看依据调度算法实现。单独运行的模式下，**上下按钮**控制该部电梯，其他电梯不受影响，不调用调度算法。**显示窗**现实电梯当前的楼层**，内部按钮**控制该部电梯直接前往该层。**紧急制动**可是电梯马上停下来且不可调度，**解除制动**恢复电梯可调度。**开关门**可手动打开关闭电梯门；**查看日志**可看到电梯调度时各个电梯的状态和调度过程。

**个人总结：**

这次项目让我学到了很多，第一：熟悉了电梯的调度算法和调度模式，通过思考锻炼了自己设计算法的能力；第二：对操作系统的进程管理有了更深的理解；第三：熟悉了java的编程，掌握了java的图形界面，进程和线程调用。

整个项目前后用了4到5天，包括一次通宵；项目刚布置下来我就开始思考如何去写，基本思路依照图书馆的电梯方式，包括电梯的界面，协作调度时无论按下哪个请求，总是调度用户等待最短的电梯。基本功能实现较为简单，刚开始用了一个晚上写好电梯的单独调度，写协作调度的核心算法时感觉有点难度，不过因为前面的准备工作做得比较好，电梯的层次分类包括状态量设置的比较有条理和清晰。通过一天的工作，基本完成电梯调度的基本功能。

后面就是实现附加功能，依据现实为标准，我设置了电源项和运行方式选择，这些都是控制相关布尔值而实现功能；后来加了紧急制动，在电梯到达每一层都判断是否有紧急制动标志符 Urgency，也是个布尔值，若有，则停止运行，并忽略已有的所有请求，但开关门依旧可以实现。查看日志是通过一个静态函数类Record来实现的，Record的类主要包括文件的新建，清空，存入数据等函数。通过这些函数我把调度的过程存入txt文件。然后在通过“查看日志”按钮打开。

关于楼层数的设置，通过改变静态变量MAXFLOOR和MAXELEVATOR的值来实现电梯的初始化。