

电梯调度程序说明

1252960 胡圣托

一、项目简介

项目名称：电梯调度模拟程序

开发环境：Java (with JDK 7)

项目概述：某一栋楼有 20 层，有 5 部互联的电梯。基于线程的思想，编写一个电梯调度程序。这五部电梯项目联结，即当一个电梯按钮按下去时，其它电梯相应按钮同时点亮，表示也按下去了。

二、界面简介

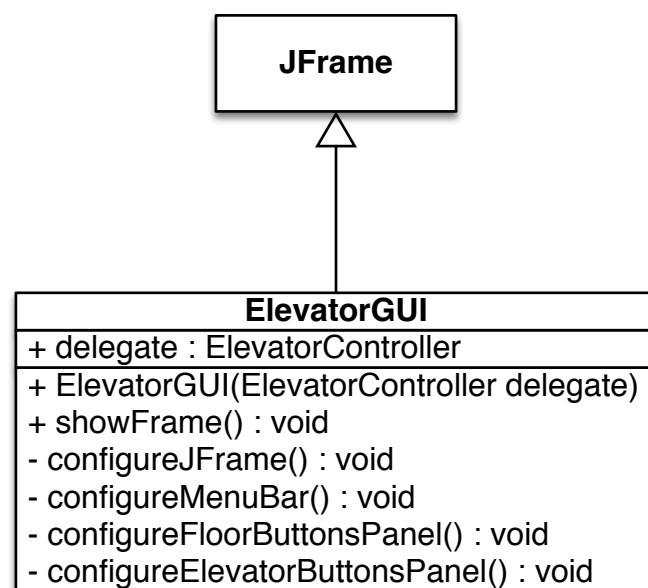


1. 右侧有为电梯内部视图，由橙色线条分开，共 5 部电梯。每部电梯都有 20 个楼层案件，黄色背景的那一列用来表示电梯楼层间的移动，其中红色表示在该层。
2. 左侧那一部分表示电梯外部视图，属于总控制台，除了第 1 层和第 20 层，每层都分别有“上”、“下”两个按钮。

三、程序结构介绍

本项目采用 MVC 的思想构建

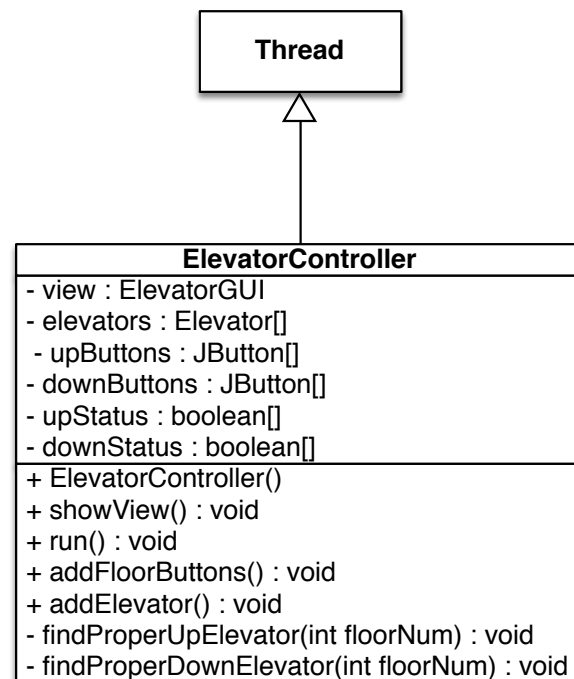
1. ElevatorGUI



```
public class ElevatorGUI extends JFrame
```

这个类主要负责主界面的布局，并将它代理给 ElevatorController，使得某些监听事件的逻辑添加可以写在 ElevatorController 中。其中的函数方法都用于 UI 设置。

2. ElevatorController



```
public class ElevatorController extends Thread
```

这个类是核心控制类(Controller) ,继承自 Thread 类 ,重写 run() ,多线程运行。

view: controller 对应的 view

elevators: 用于储存五个电梯的数组

upButtons: 储存向上按键的数组 , 便于在 controller 中添加数据

downButtons: 储存向下按键的数组

upStatus: 记录对应的向上按键是否被按下

downStatus: 记录对应的向下按键是否被按下

showView(): 显示 view

run(): 重写 Thread 中 run()

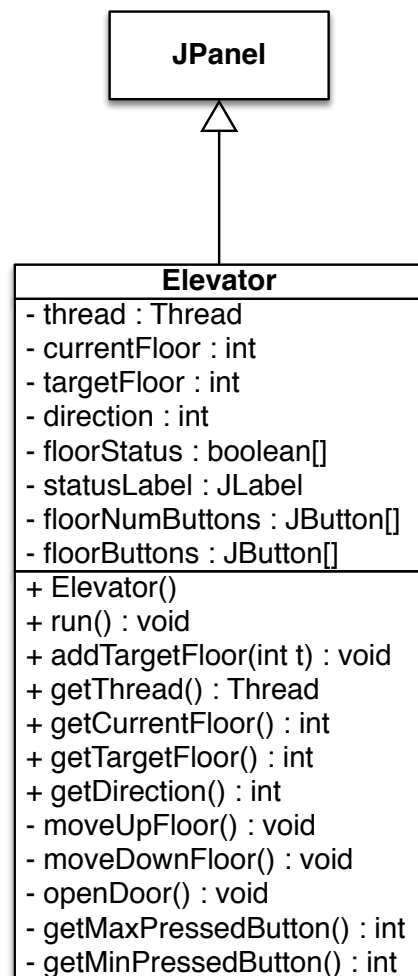
addFloorButtons(): 添加主界面左侧控制中心的所有按钮

addElevator(): 添加五部电梯

findProperUpElevator(int floorNum): 按下向上按键后寻找最优的电梯

findProperDownElevator(int floorNum): 按下向下按键后寻找最优的电梯

3. Elevator



```
public class Elevator extends JPanel implements Runnable
```

这个类表示电梯，实现了电梯内部的逻辑功能

currentFloor, targetFloor: 分别记录电梯当前楼层以及目标楼层

direction: 记录电梯的运行状态

floorStatus: 记录对应楼层的按钮是否被按下

statusLabel: 显示当前电梯的所在楼层及状态

floorNumButtons: 电梯内部的楼层按钮

floorButtons: 单个电梯界面右侧一系列的按钮，显示电梯的运动，不可点击

run(): 实现 Runnable 中的 Run()方法，以支持多线程运行

addTargetFloor(int t): 为电梯添加目标楼层

getThread(): 获取当前电梯对应的线程

getCurrentFloor(): 获取当前楼层

getTargetFloor(): 获取目标楼层

getDirection(): 获取电梯运行状态

moveUpFloor(): 电梯向上运动

moveDownFloor(): 电梯向下运动

openDoor(): 开门

getMaxPressedButton(): 获取所按按钮对应的最高楼层

getMinPressedButton(): 获取所按按钮对应的最低楼层

4. ElevatorConst

该类中定义了一系列常量

总楼层 `TOTAL_FLOOR = 20;`

电梯数量 `TOTAL_ELEVATOR = 5;`

上行状态 `STATUS_UP = 1;`

下行状态 `STATUS_DOWN = -1;`

空闲状态 `STATUS_IDEL = 0;`

四、实现思想介绍

1. 单个电梯内部

按下电梯内部的楼层按钮后，floorStatus 数组中对应的项变为 true。

若此时电梯状态为空闲(ElevatorConst.STATUS_IDEL),则将目标楼层设为当前点击楼层;若电梯为上升状态(ElevatorConst.STATUS_UP),则将目标楼层设为所有点击过的按钮中对应的最高楼层;若此时电梯为下降状态(ElevatorConst.STATUS_DOWN),则将目标楼层设置为最小楼层。

在 run()方法中比较当前楼层和目标楼层的大小关系,若当前楼层小于目标楼层,电梯向上走,并响应沿途所按下的按钮;若当前楼层大于目标楼层,电梯下行,并相应沿途所按下的按钮;若两者相等且当前楼层按钮被按下,那么开门,等待一会再关门。

2. 多部电梯统筹调度

总的思路是:在运动方向与请求方向相同的电梯和空闲的电梯中寻找离当前楼层最近的那部电梯来响应。

在 Elevator 类的 run()方法中不停地遍历 upStatus 和 downStatus 数组,并分别调用 findProperUpElevator()方法和 findProperDownElevator()方法去寻找合适的电梯,若找到合适的电梯,则调用对应电梯的 addTargetFloor()方法,并消除状态数组(upStatus 或 downStatus 数组)中的标记;若找不到合适电梯,则不消除标记,直到某轮循环找到为止。

五、改进之处

因为每次遍历寻找合适的电梯时都是按顺序遍历,所以编号越大的电梯资源利用率不高,常常处于空闲状态,这点有待改进。

六、心得体会

此次项目目的在于通过控制电梯调度来实现操作系统调度过程 ,并且学习特定环境下多线程编程的方法。在做此次项目的时候 ,我也是第一次写多线程的程序 ,最初都不怎么了解 ,查阅了一些资料才明白 ,充分利用 Thread 的 run()、sleep() 等方法 ,用多线程来模拟操作系统中的进程管理 ,加深了对进程管理知识的理解。