项目说明文档

操作系统课程设计

——电梯调度算法

作 者 姓 名： 谢宇翔

学 号： 1951708

指 导 教 师： 张惠娟

学院、 专业： 软件学院 软件工程

同济大学

Tongji University

目 录

[1 要求 1](#_Toc4650)

[2 设计 1](#_Toc19676)

[2.1 语言设计 1](#_Toc8296)

[2.2 主要类结构设计 1](#_Toc15626)

[2.2.1 Building类 1](#_Toc31422)

[2.2.2 Elevator 类 1](#_Toc14984)

[2.3 系统设计 2](#_Toc13677)

[2.3.1Building类 2](#_Toc26405)

[2.3.2Elevator 类 3](#_Toc14132)

[2.4信号量设计 3](#_Toc29203)

[3 实现 3](#_Toc28738)

[3.1 调度算法实现 3](#_Toc6523)

[3.1.1主要代码 3](#_Toc25072)

[3.1.2主要思路 4](#_Toc28128)

[3.2 单部电梯内部操作调度实现 4](#_Toc17686)

[3.2.1主要代码 4](#_Toc3474)

[3.2.2主要思路 5](#_Toc19508)

[3.3动画实现 5](#_Toc188)

# 1 要求

电梯应有一些按键，如：数字键、关门键、开门键、上行键、下行键、报警键等

有数码显示器指示当前电梯状态

每层楼、每部电梯门口，有上行、下行按钮、数码显示。

五部电梯相互联结，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。

电梯调度算法：

所有电梯初始状态都在第一层；

每个电梯没有相应请求情况下，则应该在原地保持不动；

电梯调度算法自行设计。

# 2 设计

## 2.1 语言设计

本次作业使用了java语言，使用java自带的swing库可以很好地生成界面以及设计图形，实现了算法可视化，动态化。

## 2.2 主要类结构设计

### 2.2.1 Building类

继承了Jframe类，实现整座楼的电梯调度，是其他类的基础。

public static boolean *whichFloorIsWaitUp*[]//用来存储哪些楼请求上行

public static boolean *whichFloorIsWaitDown*[]//用来存储哪些楼请求下行

Elevator elevators[]//5部电梯

public static Mybuttons *upbtn*[]//每一层楼的上行按钮

public static Mybuttons *downbtn*[]//每一层楼的下行按钮

### 2.2.2 Elevator 类

继承了Jpanel类，并实现了runnable接口，是每部电梯自身的调度操作。

2.2.2.1 内置数据

public boolean toWhichFloor[]//自身要去哪楼的数组

Mybuttons allbtn[]//楼层按钮

public int pos;//电梯当前的位置  
public int ith;//是第几部电梯  
JButton Close;//关门按钮  
JButton Open;//开门按钮  
JButton alarm;//警报按钮  
JPanel light;//装载按钮，显示灯等按钮与标签的Panel，与电梯分开  
JLabel ldoor;  
JLabel rdoor;//左右门  
JLabel floorlight;//电梯楼层的显示器  
JLabel elevator;//电梯  
Font font;//字体样式  
boolean isopen=false;  
boolean isup;  
boolean isdown;  
boolean isrun;  
boolean isalarm;//5个信号量

2.2.2.2内置函数

public int getfarestpos()//计算当前电梯要到达的最远楼层

public boolean islegal(int want,int upordown)//0为up，1为down

//判断当前电梯对要到达的楼层是否符合电梯运动的规则

public Elevator (int th)//构造函数，传入参数是第几部电梯

public void run()//电梯与灯，按钮等运行函数

void moveToFloor(int want)//电梯前往want楼层，实现电梯动态化

void Arrive()//电梯到达楼层

void Opendoor()//开门函数

void Closedoor()//关门函数

void HereWego()//电梯运行函数

void alarming()//警报发生后的警报函数

## 2.3 系统设计

### 2.3.1Building类

1.首先建立Building类为整栋建筑，设置窗口初始化。

2.建立5个电梯，分别加入Building

3.建立2排按钮，为每一层楼的上行和下行按钮

4.为上行按钮和下行按钮设置事件监听，选择合适的电梯前往

5.设置背景颜色润色界面

6.让5台电梯按线程执行起来

### 2.3.2Elevator 类

1. 初始化所有信号量与信息为正确的初始值
2. 设置初始化框架
3. 设置电梯楼层指示灯，能够按照电梯的楼层变化而变化，放在合适位置
4. 设置每一楼层的按钮，设置监听
5. 设置电梯，能够根据要求前往需要的楼层
6. 设置开门关门警报按钮，设置监听，能在触发时产生相应动作

## 2.4信号量设计

**Isrun电梯是否在运行，是等待状态还是运行状态**

**-电梯移动时设置为true**

**-电梯到达楼层停靠时设置为false**

**-电梯休息时设置为false**

**-电梯警报按钮按下后设置为false**

**-开关门按钮按下时设置为false**

**Isalarm电梯是否警报**

**-电梯正常运行时设置为false**

**-电梯警报按钮按下后设置为true**

**Isup电梯是否可上行**

**-电梯在上行时设置为true**

**-电梯下行时设置为false**

**-电梯休息时设置为true**

**-电梯警报按钮按下后设置为false**

**Isdown电梯是否可下行**

**-电梯在下行时设置为true**

**-电梯上行时设置为false**

**-电梯休息时设置为true**

**-电梯警报按钮按下后设置为false**

**Isopen电梯是否在开门状态**

**-电梯运行时设置为false**

**-电梯在楼层停靠时开门设置为true**

**-开门按钮按下时设置为true**

**-关门按钮按下时设置为false**

**-电梯警报按钮按下后设置为true//这里设置成什么都可以**

# 3 实现

## 3.1 调度算法实现

### 3.1.1主要代码

public void run() {  
 while(true)  
 {  
 if(isalarm)  
 {  
 alarming();  
 break;  
 }  
 if(isopen)  
 {  
 Opendoor();  
 try {// wait for a second  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e1) {  
 e1.printStackTrace();  
 }  
 Closedoor();  
 }  
 HereWego();  
 try {  
 Thread.*sleep*(50);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
}

### 3.1.2主要思路

通过run函数实现

1. 判断isalarm是否为true，如果是，执行alarming函数，并退出循环，使此电梯无法使用
2. 判断isopen是否为true，如果是，执行开门并关门的操作。
3. 都为false时，可以执行HereWego函数，电梯开始运动。
4. 休息一定时间间隔后重复运行

## 3.2 单部电梯内部操作调度实现

### 3.2.1主要代码

void HereWego()  
{  
 while(!isalarm&&!isopen)  
 {  
 Open.setEnabled(true);  
 Close.setEnabled(true);  
 for (int i = 0; i < 20; i++) {  
 if (toWhichFloor[i]) {  
 moveToFloor(i+1);  
 Arrive();  
 Opendoor();  
 try {// wait for a second  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e1) {  
 // *TODO Auto-generated catch block* e1.printStackTrace();  
 }  
 Closedoor();  
 try {// wait for a second  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e1) {  
 // *TODO Auto-generated catch block* e1.printStackTrace();  
 }  
 }  
 }  
 }  
}

### 3.2.2主要思路

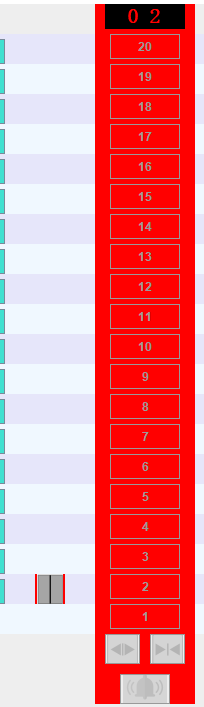
通过HereWogo函数实现

1. 先判断isopen和isalarm信号量是否为true，如果是，则直接退出
2. 对于每一层，如果toWhichFloor[i]为true，则执行以下操作：
3. 执行电梯移动函数，将电梯移动到该楼层
4. 执行Arrive函数，电梯到达楼层后，修改参数以及信号量
5. 执行opendoor函数，开门操作
6. 线程暂停1秒，表示开门时间
7. 执行关门操作，修改信号量
8. 再次暂1秒等待是否有开门按钮按下继续开门

## 3.3动画实现

1.整体实现：（由于篇幅原因只截图3部电梯）



2.警报按钮：  3.开门关门：IMG_256 IMG_256

4.指示灯：IMG_256