

**<<多电梯调度软件>>**

**设计说明书**



北京航空航天大学

2016-11

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 1.0 | 2016-11 | 岳明哲 | 张东鑫，李康宇 | 完成主体设计 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **项目组成员信息** | | |
| **小组名称** |  | |
| **学号** | **姓名** | **本文档中主要承担的工作内容** |
| 14061173 | 岳明哲 | 编写文档1-4部分并排版(1/3) |
|  | 李康宇 | 编写文档第五部分（1/3） |
|  | 张东鑫 | 编写文档第七部分(1/3) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目 录

[1. 范围 1](#_Toc307923001)

[1.1 标识 1](#_Toc307923002)

[1.2 系统概述 1](#_Toc307923003)

[1.3 文档概述 1](#_Toc307923004)

[1.4 术语和缩略词 1](#_Toc307923005)

[2. 引用文档 1](#_Toc307923006)

[3. 需求概述 2](#_Toc307923007)

[4. 体系结构设计 2](#_Toc307923008)

[4.1 总体结构 2](#_Toc307923009)

[4.2 功能分配 2](#_Toc307923010)

[4.3 关键问题及解决方案 2](#_Toc307923011)

[5. 接口设计 3](#_Toc307923012)

[5.1 用户界面设计 3](#_Toc307923013)

[5.2 外部接口设计 3](#_Toc307923014)

[5.3 内部接口设计 3](#_Toc307923015)

[6. 数据结构设计 3](#_Toc307923016)

[6.1 公共数据结构设计 3](#_Toc307923017)

[6.2 数据库设计 3](#_Toc307923018)

[7. 详细设计 4](#_Toc307923019)

# 范围

## 标识

文档标识号：A2016-01-02-00

文档标题：多电梯调度系统设计说明书

项目/产品中文全称：多部电梯集中调度系统

项目/产品英文全称： Multiple-elevator Dispatching System

项目/产品英文缩写：MEDS\_1.0

版本号：V1.0

## 系统概述

多部电梯集中调度系统可以完成多部电梯的统一调度和监控，使多部电梯能够满足用户的请求，提高电梯的使用效率。系统开发以部件单元开发，依次设计调度器模块，模拟电梯运行的模块，模拟用户发出指令的模块。

多部电梯集中调度系统的投资方、需方为使用多部电梯集中调度系统软件的电梯所有者，用户为电梯调度员以及所有电梯用户，开发方为北京航空航天大学计算机学院软件工程实践小组第27小组，支持机构为软件工程课程教师及助教。

由于软件系统仍处于设计开发过程中，因此尚无运行和维护的历史。

其它文档：第27组多电梯调度软件项目计划书v3.0

第27组多电梯调度软件需求规格说明书v1.0

## 文档概述

为明确“多部电梯集中调度系统”系统设计规格，详细内容，确保合理全面的实现系统的需求，并安排项目规划与进度、组织软件开发与测试，撰写本文档，文档供项目领导小职员、客户负责人、项目经理、设计人员、开发人员参考。

系统设计规格说明描述了“多部电梯集中调度系统（Multiple-elevator Dispatching System，MEDS\_1.0）”1.0版本的软件功能性需求和非功能性需求。这一文档计划由实现系统功能和验证系统功能正确的团队成员来使用。除非在其他地方另有说明，这里指定的所有需求都具有高优先级，而且都要在1.0版本中得以实现。

安全性要求：由开发人员负责人指定的第三方负责检查软件中是否存在漏洞。保密性要求：由于系统中不存在用户数据，因此保密性要求得到满足。

## 术语和缩略词

SDD：系统设计说明书（System Design Description）

MEDS：多部电梯集中调度系统（Multiple-elevator Dispatching System）

# 引用文档

A2010-00-01-00.SRS-系统需求规格说明书

第27组多电梯调度软件项目计划书v3.0

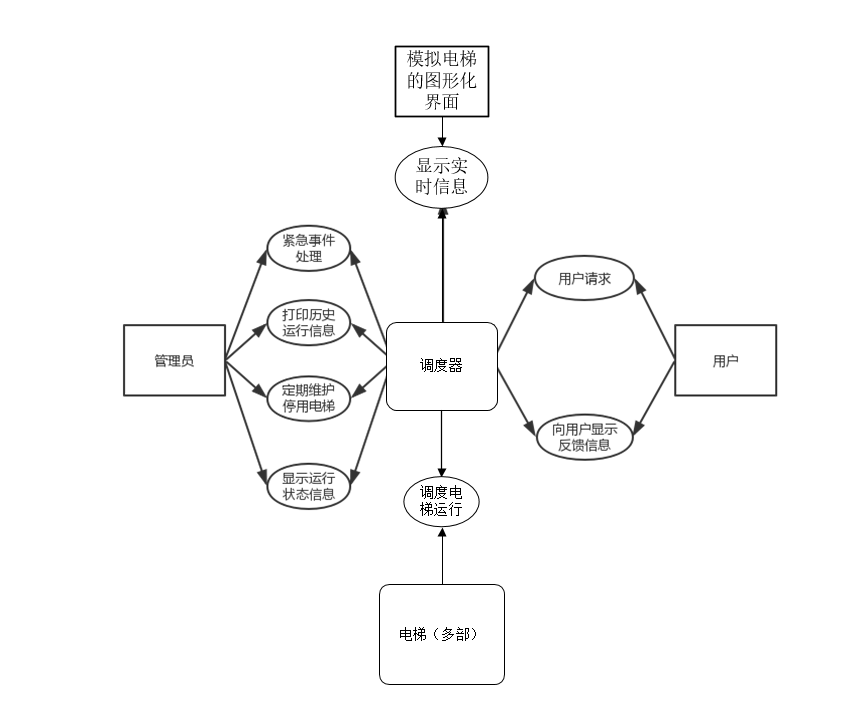
第27组多电梯调度软件需求系统需求规格说明书v1.0

# 需求概述

多电梯集中调度系统是基于现实世界的电梯系统完成的模拟调度系统，能够通过用户界面模拟对电梯发出指令，经过调度器的调度，模拟电梯运行，完成接送用户的任务。

本文档中增加了用户界面的内容，通过模拟界面的按钮实现对电梯调度器发出指令。同时界面中的电梯的运行状态显示在窗口中，方便观察运行状态。

**3.1 Usecase图**



**3.2 用例图中的参与者（新增）**

调度器：相应用户需求，接受管理员的请求，能够调度多部电梯。同时将这些信息转化成图形语言在图形化界面进行显示。

模拟电梯的图形化界面：将调度器内的参数转化成易于理解的图形信息。动态的显示实时的调度情况。

**3.3 用例说明**

（1）显示实时信息：

参与者：调度器，模拟电梯的图形化界面。

入口条件：电梯当前状态为可用。

事件流：电梯运行中将电梯的运行信息显示在图形化界面中。将用户的请求模拟成按钮显示。

出口条件：程序终止。

# 体系结构设计

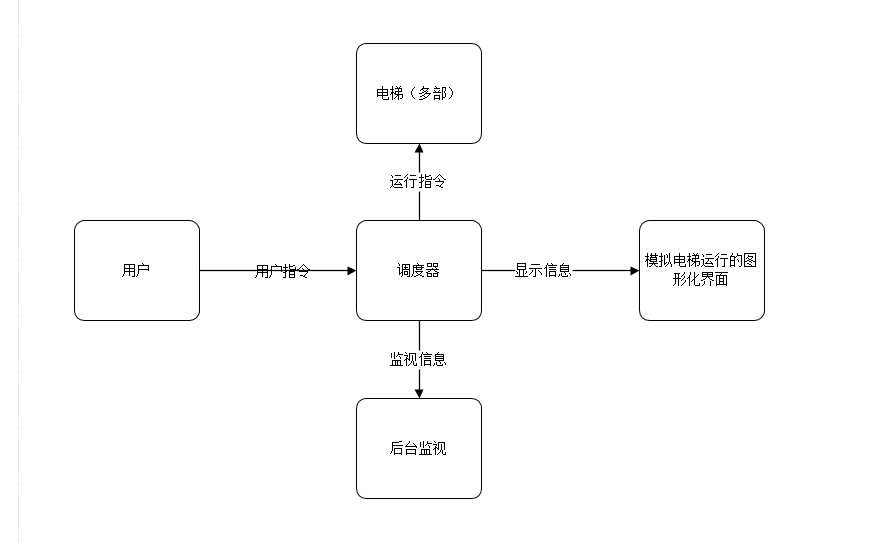
## 总体结构

用一系列图表给出系统的总体结构，并对图中的相关内容进行说明。可能包括软件体系结构、硬件体系结构、技术体系结构、支撑体系（部署和实施方案）结构等各个方面，可根据实际情况每个方面分为一小节来写。

对于软件体系结构描述，如果开发采用的是结构化分析与设计方法，建议给出模块组成及之间的调用关系、模块间的接口描述，说明:给出每个模块的具体功能；如果采用面向对象分析与设计方法，建议从UML的4+1视图描述软件结构：完善用例图、活动图，给出类图、时序图、状态图。

本系统结构设计采用数据流体系结构。

### 软件体系结构



用户发出指令给调度器，调度器根据指令的先后顺序和多部电梯的状态选择最佳调度方法，调度电梯运行用户。可以通过调度器的接口查询当前状态，同时能够通过图形化界面显示运行图像。

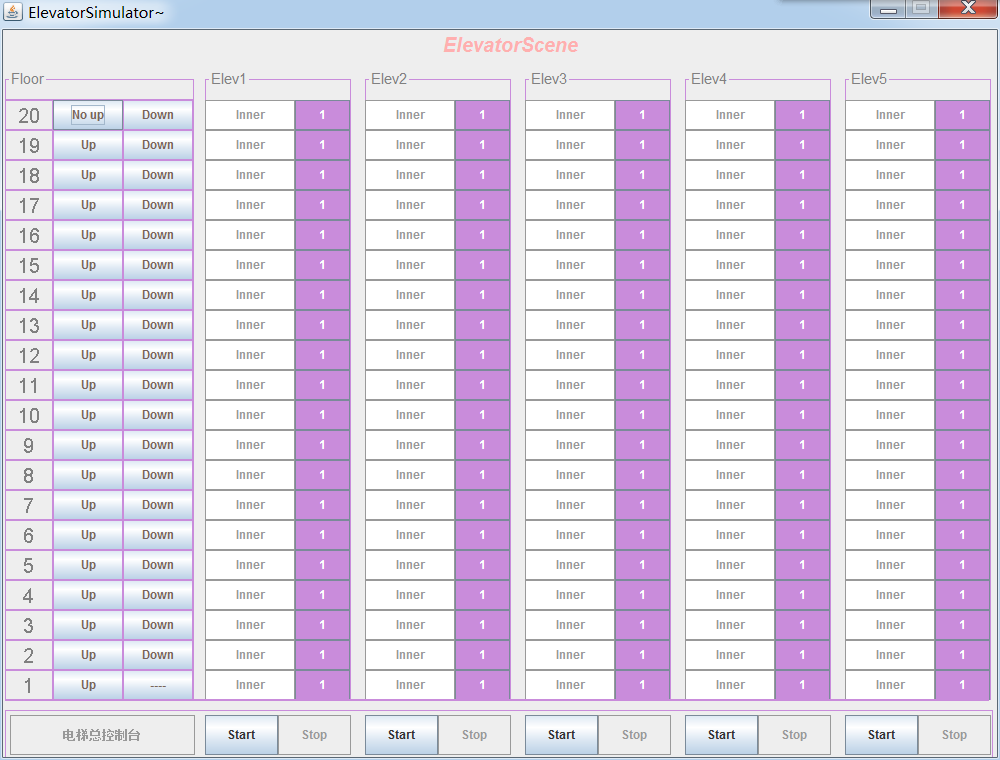
## 功能分配

本系统共设计有5项功能： 接受用户请求，确定调度顺序，调度电梯运行，提供查询服务，模拟系统运行。

# 接口设计

## 用户界面设计

用户主界面设计如下：



界面总体设计分为四部分，由上至下分别是标题区、楼层按钮区和电梯运行区以及电梯总控制台区。

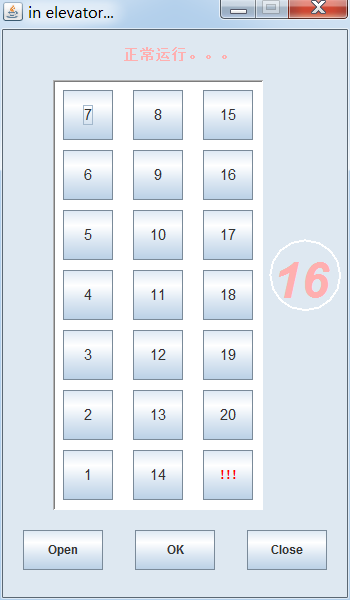
标题区包括程序名称（ElevatorScene）、楼层名称(Floor)和电梯名称标题(Elev1-5)。

楼层按钮区由楼层按钮组成，用于生成楼层指令。每一层都有up按钮和down按钮，用于发出楼层指令。

电梯运行区由5列虚拟电梯组成，直观地展示电梯的运行效果。

电梯总控制台区由用于启动和关闭电梯的按钮组成。

电梯内部分界面设计如下：



包括电梯内部楼层按钮、电梯当前楼层号，电梯运行状态的展示，电梯门开关按钮。

## 外部接口设计

无

## 内部接口设计

1. 修改电梯线程状态的接口

public void normalStop()

//将电梯状态置为暂停：

public void normalSuspend()

//将电梯状态置为继续：

public void normalResume()

//结束电梯线程：

public void normalExit()

//得到线程状态

public int getThreadState()

1. 改变电梯内部视窗的接口

public void onChangeFloor()

1. button类的接口

public void setBtnIsResponded(boolean state)

public void setAnsweredElevId(int answeredElevId)

# 数据结构设计

## 公共数据结构设计

电梯状态类ElevState()中：

int direction;//保存电梯方向

int position;//保存电梯的位置

int lastFloor;//保存电梯上次的位置

boolean isMoving;//保存电梯是否正在移动

boolean isClose;//保存电梯是否已关门

TreeSet<Integer> desFloor;//电梯目标位置

主面板类ElevMainFrame()中：

IdentifiedButton pushBt[]=new IdentifiedButton[40]//Button定义类的实例化

IdentifiedButton switchBt[]=newIdentifiedButton[10]//Button定义类的实例化

JButton elevBt[]=new JButton[100]//电梯按钮

JButton stateBt[]=new JButton[100]//状态按钮

ElevatorThread elevThr[]=new ElevatorThread[5];//电梯线程的实例化

ArrayList unAnsweredBut=new ArrayList();//存储暂未被响应请求按钮

SpareElevatorListener selistener;//电梯响应调度类的实例化

电梯线程类ElevatorThread()中：

ElevMainFrame sceneMaintainer;//主面板类的实例化

ElevState statHolder;//电梯状态类的实例化

InElevator mySpace;//电梯内部面板类的实例化

int myId;//保存电梯ID

JButton myImage;//电梯线程监听的按钮对象

public static int RUNNING=0;//线程的一个状态

public static int SUSPENDED=1;//线程的一个状态

public static int STOP=2;//线程的一个状态

private int threadState;//保存线程状态

private boolean isThreadDead;//保存线程是否活跃

电梯响应调度类SpareElevatorListener()中：

public static int RUNNING=0; //线程的一个状态

public static int SUSPENDED=1; //线程的一个状态

private int threadState;//保存线程状态

ElevMainFrame emframe;//主面板类的实例化

private boolean isThreadDead;//保存线程是否活跃

Button定义类IdentifiedButton()中：

int btnId;//保存按钮ID

int btnFunc;//保存按钮作用

boolean isResponded;//保存按钮等待响应状态

int answeredElevId;//保存响应的电梯ID

电梯内部面板类InElevator()中：

JButton numberBut[]//作为数字按钮

JButton openBut//作为开门按钮

JButton closeBut//作为关门按钮

JButton alertBut//作为警报按钮

JLabel posShowBut//显示电梯当前楼层的标签

JLabel elevState//显示电梯当前运行状态的标签

JButton okBut//界面结束按钮

ElevatorThread etBelonged;//实例化的电梯线程

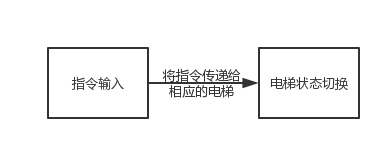
## 数据库设计

无

# 详细设计

## 电梯总控制模块

### 功能流程图



需要说明的问题：

1. 指令输入使用的是在图形化界面按相应的按钮的方式。

### 功能描述

功能类型：状态控制

功能描述：控制电梯的运行状态

前提业务：无

后继业务：04 （模拟电梯运行模块）

功能约束：

约束描述：

操作权限：系统使用者

### 动作说明

|  |  |  |
| --- | --- | --- |
| 动作编号 | 动作名称 | 动作描述 |
| 1 | 修改电梯状态 | 点击start或stop按钮，修改电梯的运行状态。 |

### 7.1.4 数据要求

无

### 内部逻辑

1. 电梯初始状态设置：

for(int i=0;i<5;i++)

{

switchBt[2\*i]=new IdentifiedButton("Start",i+1,1);

switchBt[2\*i].setBounds(200+i\*160,5,73,40);

switchBt[2\*i+1]=new IdentifiedButton("Stop",i+1,-1);

switchBt[2\*i+1].setBounds(273+i\*160,5,73,40);

switchBt[2\*i+1].setEnabled(false);// 所有电梯初始状态均为未启动

switchBt[2\*i].addActionListener(this);

switchBt[2\*i+1].addActionListener(this);

controlPanel.add(switchBt[2\*i]);

controlPanel.add(switchBt[2\*i+1]);

}

1. 电梯状态修改。点击按钮后修改相应电梯的状态。

for(int i=0;i<10;i++)

{

if(ae.getSource()==switchBt[i])

{

switchBt[i].setEnabled(false);

/\*\*

\* 因为开关控件的的btnFunc用1和-1表示，所以要改变和它对应的按钮的状态

\* 只需这一行代码

\*/

switchBt[i+switchBt[i].getBtnFunc()].setEnabled(true);

if(switchBt[i].getBtnFunc()==1)//如果是启动按钮

{

//这里因为电梯线程作为数组元素被存储，所以其索引应该要比它的ID号少1

elevThr[switchBt[i].getBtnId()-1].normalStart();

}

else

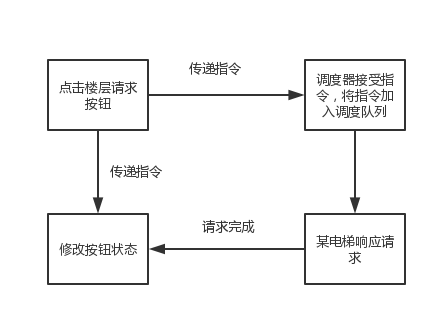
elevThr[switchBt[i].getBtnId()-1].normalStop();

}

}

## 楼层请求模块

### 功能流程图



### 功能描述

功能类型：请求处理

功能描述：处理楼层请求

前提业务：无

后继业务：05（调度器模块）

功能约束：

约束描述：

操作权限：使用者

### 动作说明

|  |  |  |
| --- | --- | --- |
| 动作编号 | 动作名称 | 动作描述 |
| 1 | 输入楼层请求 | 点击某楼层的UP或DOWN按钮，生成楼层请求，传递给调度器。 |

### 数据要求

无

### 内部逻辑

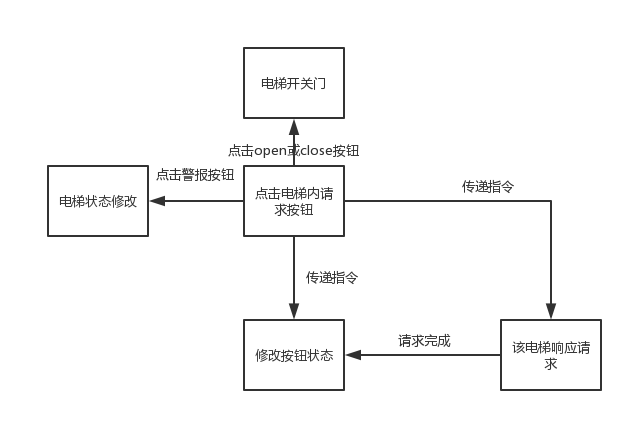
1. 楼层按钮初始化，在主屏幕上生成模拟楼层请求的按钮。

2.点击某楼层的UP或DOWN按钮，修改该按钮的状态，在请求被响应前处于不可按状态，同时生成请求加入调度器队列中。

1. 当请求被完成时，恢复按钮状态，重新变成可被按下的状态

## 电梯内请求模块

### 功能流程图



### 功能描述

功能类型：请求处理

功能描述：处理电梯内部请求，包括前往的楼层请求，开关门请求，以及警报。

前提业务：01（电梯总控制台）

后继业务：04（电梯模块）

功能约束：

约束描述：

操作权限：使用者

### 动作说明

|  |  |  |
| --- | --- | --- |
| 动作编号 | 动作名称 | 动作描述 |
| 1 | 输入楼层请求 | 点击代表楼层的数字按钮，生成楼层请求，加入电梯的请求队列。 |
| 2 | 开关门请求 | 点击open或close，使电梯开关门，只有在电梯在某层停靠时可用 |
| 3 | 警报 | 点击警报按钮，发出警报，修改当前电梯的状态。 |

### 数据要求

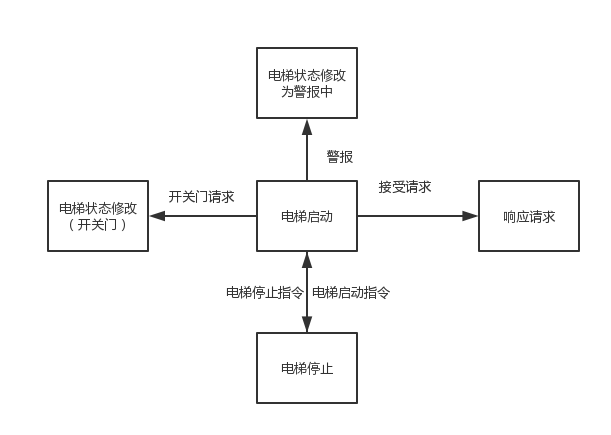
无

### 内部逻辑

1. 电梯内按钮初始化，生成模拟电梯内部请求的按钮。点击相应的电梯可以看到这些按钮。
2. 点击代表楼层的数字按钮，修改该按钮的状态，在请求被响应前处于不可按状态，将请求加入电梯的响应队列。
3. 点击close或open按钮，关闭电梯状态，使电梯开门或关门，只有在电梯在某层停靠时可用。
4. 点击警报按钮，更改电梯状态。

## 电梯模块

### 功能流程图



需要说明的问题：每一部电梯是一个线程，共有5部电梯。在总控制台启动电梯后，电梯的状态变为运行中，此时，这部电梯可以接受请求。电梯会按顺序执行调度器分配的请求，到达目标楼层后，电梯会开门，并停靠3秒，在这期间，可以按电梯内的开关门按钮，使电梯开关门。在电梯内部可以按警报按钮，使电梯状态修改。

### 功能描述

功能类型：线程控制

功能描述：

前提业务：01（电梯总控制台）02（楼层请求模块） 03（电梯内请求模块）05（调度器模块）

后继业务：

功能约束：

约束描述：

操作权限：使用者

### 动作说明

|  |  |  |
| --- | --- | --- |
| 动作编号 | 动作名称 | 动作描述 |
| 1 | 启动或停止 | 电梯总控制台发出指令，启动电梯，电梯状态修改为正在运行；停止电梯，电梯状态修改为停止。 |
| 2 | 开关门请求 | 接受开关门请求，开门或关门 |
| 3 | 警报 | 接受警报，修改当前电梯的状态为警报中。 |
| 4 | 目标楼层请求 | 接受请求，响应请求。 |

### 数据要求

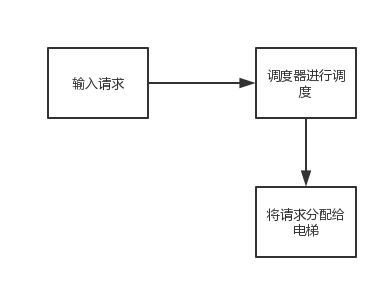
无

### 内部逻辑

1. 电梯线程
2. 外部接口 用于修改电梯线程的状态使线程停止

## 调度器模块

### 功能流程图



### 功能描述

功能类型：请求调度

功能描述：

前提业务：02（楼层请求模块）

后继业务：04（电梯模块）

功能约束：

约束描述：

操作权限：

### 动作说明

|  |  |  |
| --- | --- | --- |
| 动作编号 | 动作名称 | 动作描述 |
| 1 | 请求调度 | 使用者发出请求后，请求被调度器接受，调度器根据一定的算法，选择合适的电梯响应请求。 |

### 数据要求

无

### 内部逻辑

请求调度，调度器从正在运行的电梯中选择距离请求发出楼层最近的一部电梯响应请求，如果有多部电梯距离相同，则优先选择正在运动的电梯，如果电梯都处于停靠状态，优先选择不在一层的电梯，如果各种情况都相同，则随机选择一部电梯响应请求。如果当前没有电梯处于运行状态，则把请求加入等待队列，等到有电梯启动后再进行分配。