**电梯调度系统**

操作系统第一次作业

指导老师：张慧娟

姓名:李若冰

学号：2251657

学院/专业：软件学院/软件工程

# 1项目分析

## 1.1项目名称

电梯调度系统

## 1.2基本任务

某一层楼 20 层，有五部互联的电梯。基于线程思想，编写一个电梯调度程序。

## 1.3功能需求

1、每个电梯里面设置必要功能键：如数字键、当前电梯的楼层数、上升及下降状态等。

2、每层楼的每部电梯门口，应该有上行和下行按钮。

3、五部电梯门口的按钮是互联结的，即一个楼层外部按钮被按下后，五台电梯都有可能来响应该请求。

4、所有电梯初始状态都在第一层。每个电梯如果在它的上层或者下层没有相应请求情况下，则应该在原地保持不动。

## 1.4开发环境

**○操作系统：Windows10 家庭中文版**

**○编程语言：python**

**○开发工具：vscode**

**○GUI库：qt**

# 2.项目设计

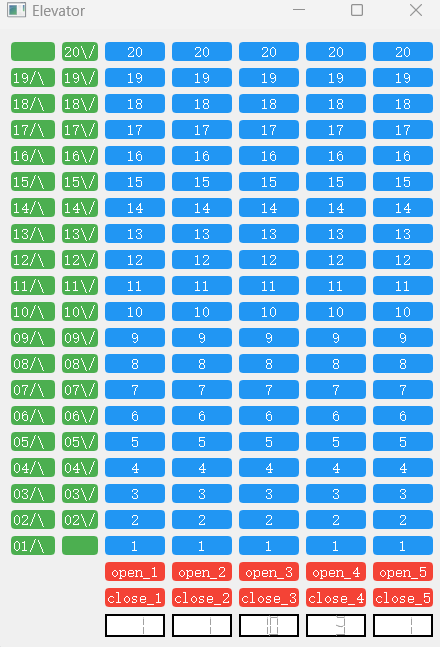
## 2.1界面设计

界面蓝色主页部分是五部电梯，每部电梯下方有两个数码管，来实时显示电梯当前楼层，每部电梯上是1-20层的显示，如果有人在电梯内部进行楼层呼叫，电梯运行到底指定楼层后，变成绿色表示开门进行中，关门后重新变回蓝色，属于电梯的内部交互。

界面左侧绿色部分是外部交互部分，在每一个楼层都会有人叫梯进行的模拟。

界面右侧下方有open和close按钮，电梯内部的人员可以自行选择开门/关门，属于电梯的内部交互。

**界面截图：**



## 2.2调度算法设计

#### 2.2.1、总体调度概述

本项目的调度主要分为两个部分：外部调度、内部调度。

外部调度的主要作用是：有乘客进行外部请求电梯的过程中，五部电梯可以相应请求并且到请求的楼层。

内部调度的主要作用是：有乘客进行内部请求电梯的过程中，乘客所在的电梯可以进行正常相应。

1. 定义了一些常量和全局变量，包括电梯状态、楼层数、电梯方向等。
2. 创建了一个连接到Qt的信号机制类Link\_to\_qt，用于发送消息。
3. 定义了一个消息类Msg，用于表示消息的类型和值。
4. 创建了一个电梯类Elevator，其中包含了电梯的状态、楼层、方向等属性，以及上升、下降等方法。
5. 创建了一个UI类Example，用于显示电梯按钮和LCD数字显示屏，并实现了按钮点击事件的处理函数。
6. 定义了一些线程函数，包括开门、关门等。
7. 定义了一个状态机函数statemachine，用于模拟电梯状态的变化。

在if \_\_name\_\_ == '\_\_main\_\_':中，创建了Qt应用程序实例，并启动了UI界面和状态机线程。

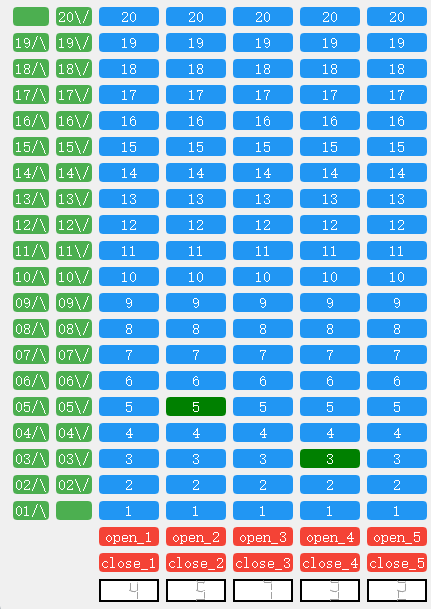
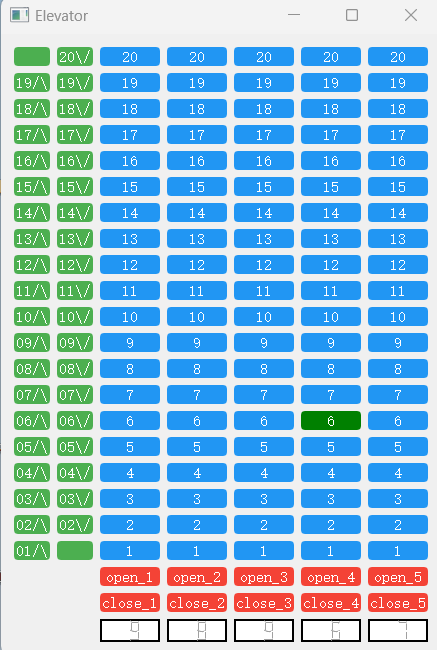
整体运行逻辑是，用户通过UI界面点击按钮模拟乘客请求，这些请求被发送到消息队列中。然后，状态机函数根据电梯当前的状态和消息队列中的请求，模拟电梯的运行状态，包括开门、关门、上升、下降等。同时，状态机函数也会发送消息到UI界面，更新LCD数字显示屏的值和按钮的状态。

#### 2.2.2电梯内部调度

当电梯内的乘客点击对应楼层的时候，电梯会进行到相应楼层，并且相应楼层的按钮变绿表示正在开门。

如下图，是分别把几个不同的电梯叫到不同的楼层。

具体实现



## 2.2.3外部调度

对于每一个外部请求（即楼层的上下按钮）而言，当这种请求出现时，不应该对应于某一个特定的电梯，而是应该由联合调度系统通过计算来规划出更可能是最好选择的电梯来执行外部请求。

当点击绿色的外部调度按钮，如果每个电梯完全没有其他任务进程，则根据平均载客原则，全部向点击楼层前进，并且到达楼层后进行开门动作。

**具体实现**

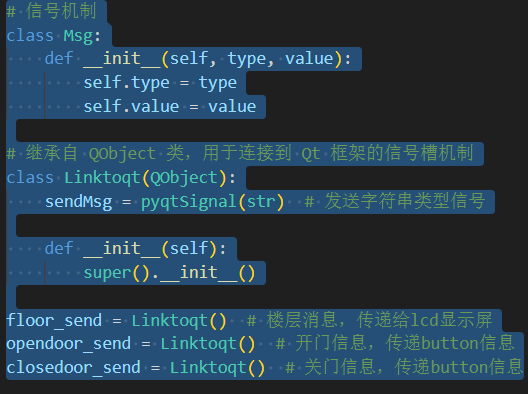


# 3.代码实现

## 3.1信号机制（example.py）

在 PyQt 中，信号用于对象之间的通信，特别是在使用 Qt 框架时，通过信号可以实现对象之间的事件触发和响应。

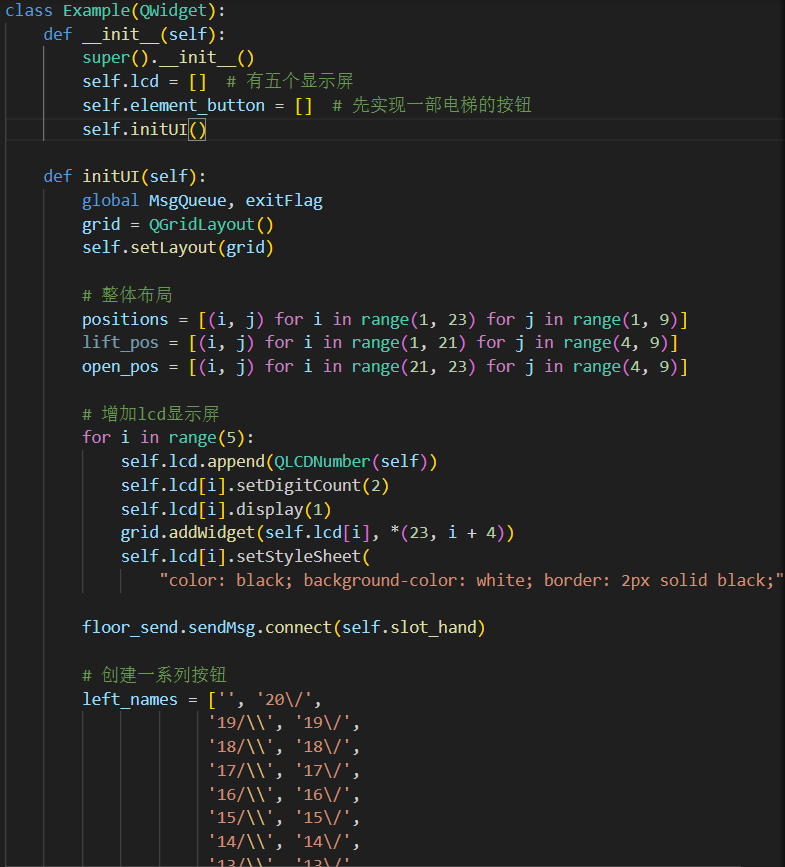
* Msg 类用于定义消息的结构，包括消息的类型和值。
* Linktoqt 类继承自 QObject，这使得它能够连接到 Qt 框架的信号槽机制。在该类中，定义了一个名为 sendMsg 的信号，它的类型是 pyqtSignal(str)，表示该信号将传递字符串类型的参数。
* floor\_send、opendoor\_send 和 closedoor\_send 是 Linktoqt 类的实例，它们用于传递不同类型的消息，如楼层消息、开门消息和关门消息。



## 3.2交互界面设计(example.py)

Example 类实现了一个简单的电梯控制界面，用户可以通过界面上的按钮操作电梯的上下行和开关门等功能。

* \_\_init\_\_ 方法用于初始化界面的各个部件，包括五个显示屏（LCD）和一系列按钮。
* initUI 方法用于初始化界面布局，包括设置显示屏和按钮的样式、连接信号和槽等。
* opendoor 方法用于处理开门信号，当接收到开门消息时，将对应楼层的按钮背景色设置为绿色。
* closedoor 方法用于处理关门信号，当接收到关门消息时，将对应楼层的按钮背景色恢复为原来的颜色。
* slot\_hand 方法是一个回调函数，用于处理接收到的消息，将消息中的楼层信息显示在对应的 LCD 上。
* up\_down\_button\_clicked 方法是上下楼按钮的槽函数，用于处理上下楼按钮的点击事件，根据按钮的文本内容创建并发送消息到消息队列中。
* floor\_clicked 方法是楼层按钮的槽函数，用于处理楼层按钮的点击事件，获取按钮对应的楼层信息并发送到消息队列中。



(部分代码展示)

## 3.3电梯类设计（elevator.py）

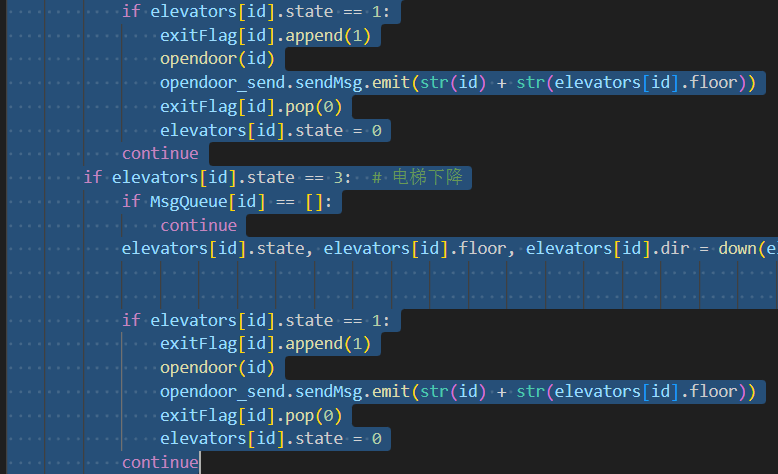
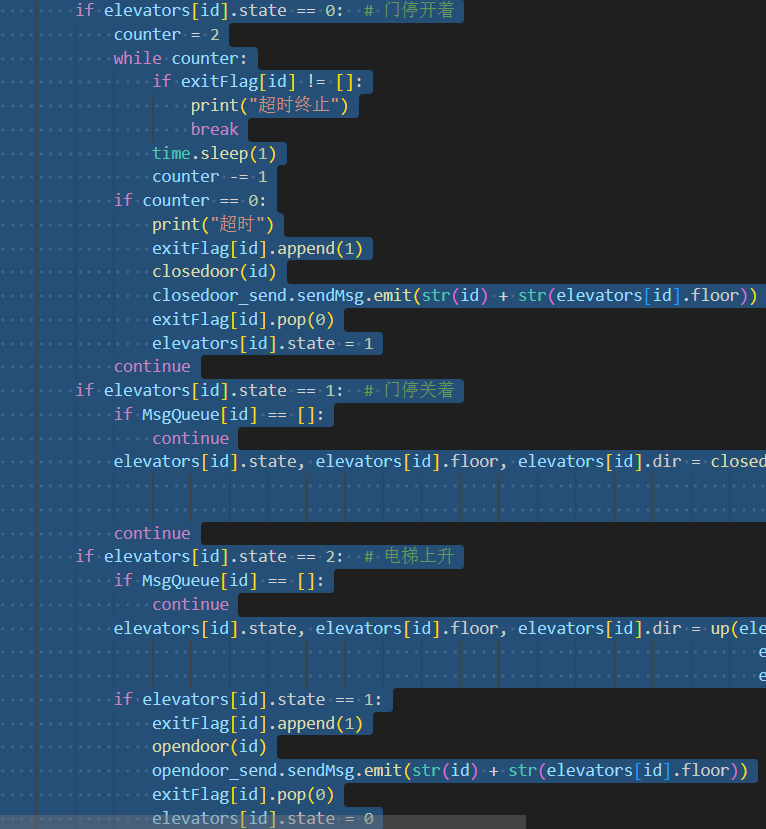
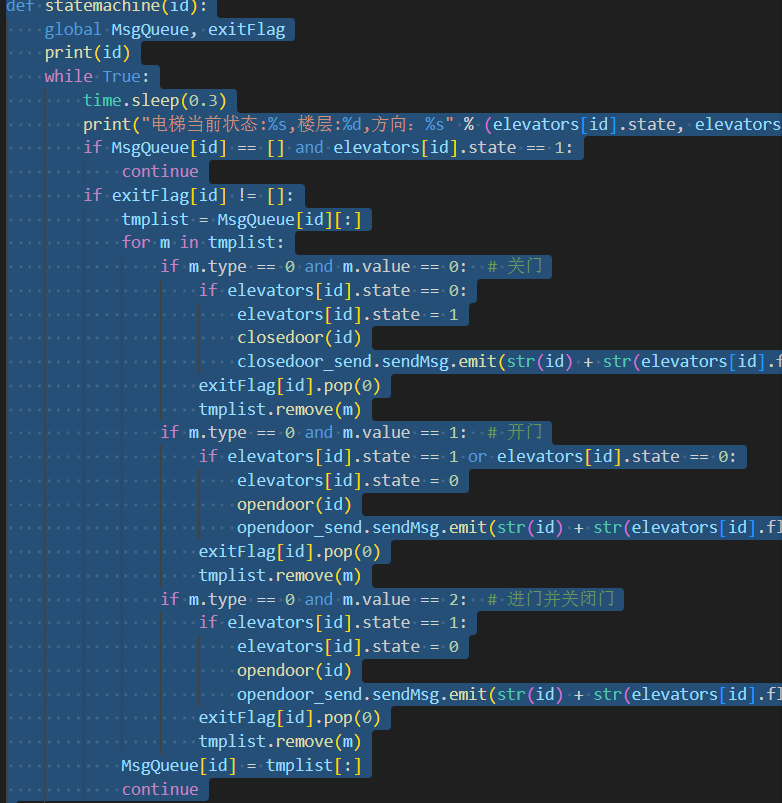
Elevator 类可以实例化一个电梯对象，并通过调用其方法来模拟电梯的上下行动作，并获取电梯当前所在的楼层。

* state：表示电梯门的状态，初始值为 1，表示门是关闭的。
* floor：表示电梯所在的楼层，初始值为 1。
* dir：表示电梯的运行方向，初始值为 0，表示静止状态。
* \_\_init\_\_ 方法用于初始化电梯对象，接受一个参数 id，表示电梯的编号。
* up 方法用于使电梯上升一层，如果当前楼层小于 20，则楼层加一；否则打印提示信息，表示已经到达顶层，无法继续上升。
* down 方法用于使电梯下降一层，如果当前楼层大于 1，则楼层减一；否则打印提示信息，表示已经到达底层，无法继续下降。
* get\_floor 方法用于打印当前电梯所在的楼层。

## 3.4电梯调度设计

statemachine 函数负责模拟电梯的运行过程，根据电梯的状态和外部指令执行相应的动作，并不断更新电梯的状态和楼层信息。

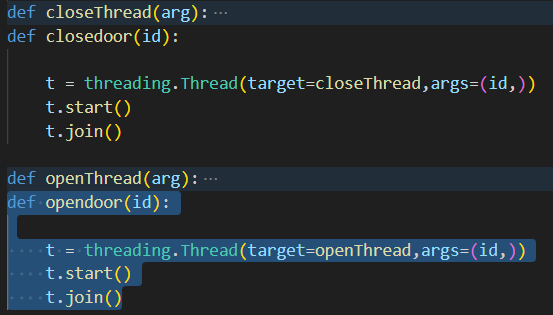
* 该函数通过不断循环，模拟了电梯的运行状态。
* 在每次循环中，首先打印当前电梯的状态、楼层和方向信息。
* 然后根据电梯的状态和消息队列中的指令执行相应的操作，包括开门、关门、上升、下降等。
* 在处理完指令后，更新电梯的状态和楼层信息，并根据需要发送相应的信号给界面显示。



## 3.5线程设计

以opendoor为例，用于模拟电梯开门的操作。

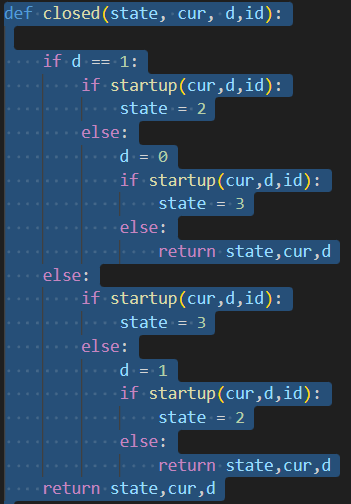
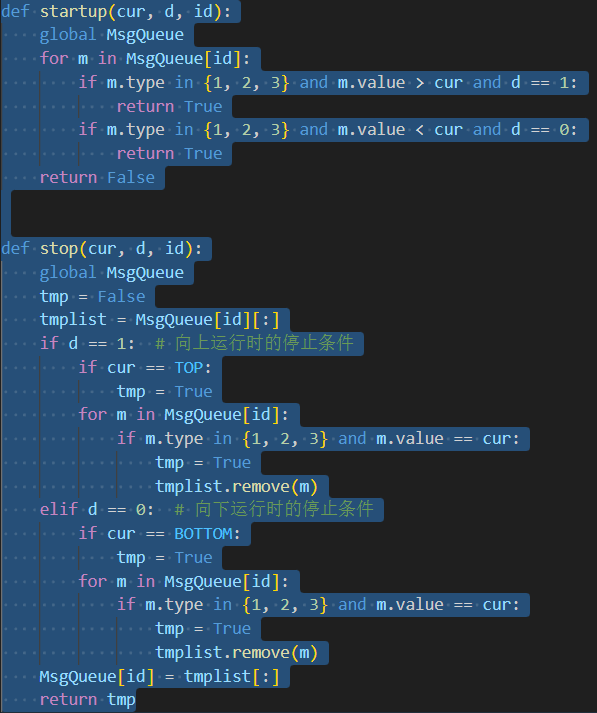
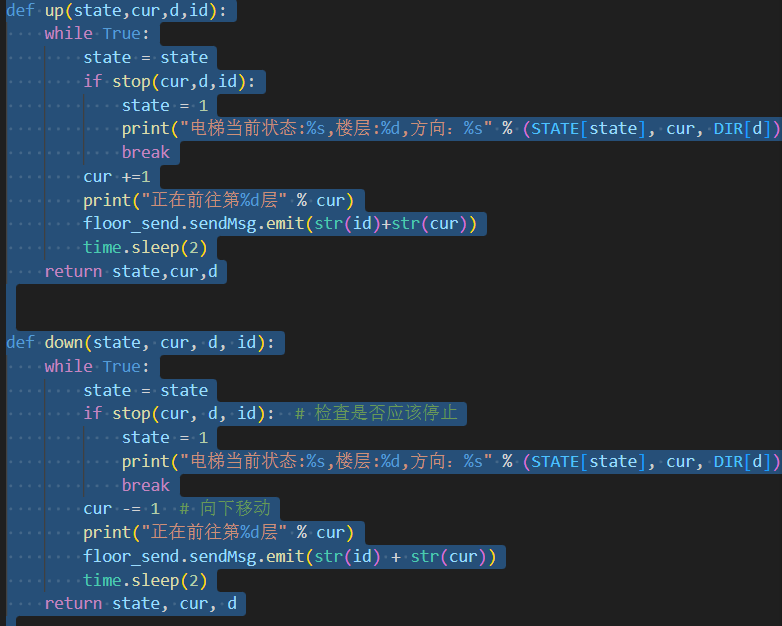
* opendoor 函数会创建一个新的线程 t，并将其绑定到 openThread 函数上。
* openThread 函数用于模拟电梯开门的实际操作，可能包括改变电梯按钮的颜色、发出开门声音等。
* 通过创建新线程的方式，可以使得电梯的开门操作与其他操作并发执行，提高了程序的效率和响应速度。
* t.join() 语句用于等待线程 t 执行完成，确保开门操作完成后才继续执行主线程的后续操作。



## 3.6运行模拟（qt,py）

5个函数共同构成电梯状态机的运行逻辑，通过不断地检查当前状态和消息队列中的任务，模拟了电梯的运行过程。

* closed(state, cur, d, id): 该函数用于模拟电梯门关上后的状态变化。根据当前状态、楼层和方向，决定下一步的动作。如果在当前方向上没有目标楼层，则改变方向；如果在当前方向上有目标楼层，则继续朝该方向运行。
* up(state, cur, d, id): 该函数模拟电梯向上运行的过程。在循环中，先检查是否需要停止，如果需要停止则改变状态为停止状态；否则继续向上运行，并发送当前楼层信息给显示屏。
* down(state, cur, d, id): 该函数模拟电梯向下运行的过程。在循环中，同样先检查是否需要停止，如果需要停止则改变状态为停止状态；否则继续向下运行，并发送当前楼层信息给显示屏。
* startup(cur, d, id): 该函数用于检查是否有目标楼层需要电梯前往，即在当前方向上是否有等待服务的乘客。如果有，则返回True，否则返回False。
* stop(cur, d, id): 该函数用于检查电梯是否需要停止。根据当前楼层和方向，判断是否到达目标楼层或者到达顶层/底层。如果需要停止，则返回True，否则返回False，并且更新消息队列中的状态。



# 4、总结

* 通过本次实验，一方面，通过本次实验，我深刻体会到了多线程调度算法的设计与实现。在操作系统中，每个电梯就好像一个线程，它们都需要申请资源，并且电梯的调度过程就是在模拟线程及其资源的调度。
* 在本次实验的过程中，我使用了一个我完全没有接触过的GUI工具pyqt5，在整个配置过程当中也遇到了一些问题，比如说运用清华源的时候忘记关闭vpn，还有在进行配置的时候，了解了虚拟环境的两种格式.venv和conda，使用的指令不同，我在最初配置一个之后忘记使用的哪个，导致指令一直出错。在进行打包exe的过程当中，报了几个pathlib error，这个过程我发现是需要重新下载，但是清理的过程非常缓慢，排查原因之后进行了环境配置，才能够正常的删除和打包exe。
* 在本次实验的过程当中，我也遇到了许多界面实现方面的问题，但是通过自己摸索和互相交流，我都顺利的解决了这些问题，这让我对于代码的运用能力也是有一定程度的提高。
* 综上所述，在电梯调度系统的开发当中，我了解了开发工具的使用方法，相关环境的配置方法，和线程调度的相关知识。

# 5.提交说明及注意事项

* 本实验的python项目文件夹主要部分分为三个子文件,qt.py,elevator.py,example.py，还有虚拟环境文件等。
* 本实验的说明文档为.docx格式
* 本实验提供可运行文件.exe格式，可直接运行/可以在vscode打开文件后配置虚拟环境后进行直接按钮运行。