褚老师好，我之前的版本没有保留分离编译的文件，只有写在一块的 cpp，就直接发这个了，不好意思= =
另外我会附上第 4 次的分离编译式项目文件，供参考，谢谢!

总体设计思路:

分四个大块：电梯类，乘客类，Settings 部分（开头的宏定义）以及打印输出信息（main 函数），
其中，电梯类和乘客类相互设置为友元，可访问对方的数据成员。
乘客类管理什么时候上电梯，电梯类管理什么时候下乘客，
定义 pas_remained 来表明当前时刻还剩下多少乘客没有完成目标，清零时退出主循环。

定义全局变量 vector<Passenger*> pas_wait_list 表示还没有结束的乘客的等待名单
注意传入的是指针，不然 vector 容器使用的是值拷贝

main 函数中设置主循环与表示当前时间的计数器 counter，一次循环表示 1 秒，可以通过 Settings 中的 MODE 来调节为自动播放（含倍速）和手动播放。

主循环中，一个循环会使所有的电梯和乘客执行一次自己的 run 函数，从而决定这一时刻该对象的状态，实现重点具体见下:

通过向 run()中传递当前时间的计数器

a.电梯类:

重点:

1. 利用类的 static 成员 ele_code 来使得每次初始化的电梯对象的编号不同（构造函数中++）
2. 电梯的状态参数:
   isselected 表示是否被选中，如果被选中，direction_up 表示电梯运行方向朝上还是朝下
3. pop_pasgr()表示电梯判断该乘客已到达目标楼层，使其下电梯后记录下电梯人数，并设置下电梯乘客的下一次状态
4. push_pasgr()表示乘客进入电梯，当电梯只有一个乘客时，运行方向变为乘客的方向，记录上电梯人数
5. run()首先检测电梯是否有第一个乘客上，不然的话等待;
   然后每一时刻检验，是否有乘客需要自己来接（乘客按下按钮），从而设置自己的 isselected 状态
   之后通过设置状态量分为两个部分：等待部分（上下梯）以及上升下降部分，利用 count 分别处理各自的时间，并改变状态
   其中，通过每一时刻新增的乘客数和下去的乘客数来刷新等待时间
6. check()作用于 run()的开头，判断当电梯没有乘客时是否有乘客需要去接，并改变自身状态（如果有）

（实现细节见程序，注释较为详细）

b.乘客类:

重点:

1. 利用类的 static 成员 pas_code 来使得每次初始化的乘客对象的编号不同（构造函数中++）
2. 乘客类的构造函数中，随机生成乘客的目标楼层、到达时间、选择的电梯编号，并将其加入等待名单
3. run()在每一时刻判断当前目标电梯是否到达自己的楼层，是否应该上这个电梯(通过排除不该上的情况)
   如果上了，则设置其状态，调用电梯的 push_pasgr()将其加入电梯的乘客 vector

4. choose_elevator()表示乘客如何选择下一次要乘坐的电梯

（实现细节见程序，注释较为详细）

c.Settings

1. 可通过这里的宏定义来修改程序的各项可变参数

d.main 函数（打印部分）

第 4 次上机中打印的各名称含义：

Elevator	编号 状态（ - / ↑ / ↓ ）
first_pas_code:	要去接的第一个乘客的编号， a--b 表示编号为 a 的乘客要去 b 楼层
wait_time:	电梯当前需要等待的时间（包括等待乘客上下梯和楼层间运行）
Passengers:	当前电梯上的乘客
isselected:	当前电梯是否被选中
waiting_pas:	是否处于等待乘客上下梯状态
between_floor:	是否处于楼层间（正在上升或下降）
set:	是否有第一个乘客按下按钮选择该电梯（其实是一个锁）
Remained Passengers List	剩余未完成的乘客名单
arriveds at a	表示在 a 时刻到达
selects elevator a	表示当乘客在楼层间休息好时按下按钮后，前来接他的电梯
current_floor 和 dest_floor	表示当前楼层和目标楼层
wait_time	表示乘客随机生成的下电梯后在该楼层的等待时间
Total_took_time	表示乘客共计要坐多少次电梯 - 1 （最后回到 1 层）
took_time	表示乘客已经坐了多少次电梯
is_ready	表示乘客当前是否已经按下按钮
current time	表示主循环中的时间计数

*注：第四次最后输出的状态中各电梯的运行时间+休息时间=current time，证明结果无误

==========

第 2 次上机修改部分：

1. 电梯类新增判断是否为第一个按下电梯的乘客(set 锁)，如果是，则赋予其最高优先级
2. 乘客类新增 L(乘坐电梯次数，随机生成), took_time, wait_time, counter(用于乘客等待时间计数)，改变构造函数内容
3. pop_pasgr()更新了对乘客下电梯时状态设置，并判断乘客是否已经完成自己的目标，修改 pas_remained(如果是的话)
4. 结合 set 锁，修改了电梯类的 check()

5. 更新乘客的 run()方法

具体结果见文件夹 第二次上机

==========

第 3 次上机修改部分：

1. 电梯类新增 enabled_floor()函数，用于判断传入的楼层是否对于当前电梯可到达
2. 电梯类新增 frame_chose 成员，用于区分同一个主循环中先后按下按钮的乘客（优化部分）
3. 乘客类新增 chosen_ele 成员，表示是否已经确定下次乘坐的电梯编号
4. 新增 choose_elevator_first()函数，用于生成每一乘客第一次要乘坐的电梯编号

   具体方法：遍历所有电梯，生成一个可乘坐的电梯的集合，然后取随机一个电梯
5. 新增 choose_elevator()函数，用于生成乘客按下按钮后前来接他的电梯

   具体方法：遍历所有的电梯，如果有电梯正处于闲置状态且暂未被帧选中(frame_chose)，加入集合

   继续遍历，找到所有闲置的电梯，并在其中找到离自己最近的一部，记录到乘客的 ele_code
6. 更新乘客的 run()方法，用于模拟等待结束后按下按钮
7. 电梯初始化放入全局，方便程序进行

具体结果见文件夹 第三次上机

==========

第 4 次上机修改部分：

1. 电梯类新增 waiting_time 和 resting_time 成员
2. 乘客类新增按下电梯后等待时间的集合 vector<size_t>
3. 主程序循环结束后新的输出

具体结果见文件夹 第四次上机