**数据结构与算法课程实验报告**

|  |  |
| --- | --- |
| **实验名称：** | **模拟银行** |

|  |  |
| --- | --- |
| **学 员 姓 名：** |  |
| **学 号：** |  |
| **年 级：** | **大二** |
| **学 员 队：** | **计算机五队** |

摘　要

本实验对银行排队系统进行了三种不同的 VIP 策略的增添和修改，以优化服务效果。具体包括：

增设 VIP 窗口： 在银行中增加了 VIP 窗口，专门为 VIP 用户提供服务。VIP 用户可以按照空 VIP 窗口、空普通窗口、非空 VIP 窗口、非空普通窗口的顺序选择排队窗口，以减小 VIP 的等待时间。1号窗口被指定为 VIP 窗口，只有 VIP 用户可以选择该窗口，非 VIP 用户无法选择，以提高 VIP 用户的优先级。

VIP 插队： 允许 VIP 用户在普通窗口队列中插队，按照到达顺序排列。VIP 用户选择排队的优先级为：空 VIP 窗口 > 空普通窗口 > 在普通窗口插队。通过调整出队和入队的操作，实现了 VIP 用户在普通窗口排队时的灵活选择。

VIP 限时等待： 在没有设置 VIP 窗口的情况下，所有出纳柜员在每一次办理完业务后检查是否有等待时间已满的 VIP 客户。对于等待时间满的 VIP 用户，按照等待时间由长到短的顺序选择进行服务。VIP 用户在等待时间达到最大等待时间后可以选择更换柜台服务，打破了最初不能更换队伍的约定。

第1章　实验原理

1.1　实验内容

在原有代码的基础上增添修改。实现了三种vip策略。

在银行增加vip窗口。Vip窗口只服务vip，vip可以按照空vip窗口，空普通窗口，非空vip窗口，非空非vip窗口的顺序来选择排队的窗口。以此减小vip的等待时间。

Vip插队。Vip用户可以插普通用户的队，对于vip则按照时间顺序排列。Vip用户选择的优先级顺序为空vip，空普通窗口，在普通窗口插队。

Vip限时等待。在不设置vip窗口的情况下，所有出纳柜员将在每一次办理完业务时检查有没有等待时间已满的vip客户，对于等待时间满的vip用户，则按照等待时间顺序来选择进行服务。

最后对三种排队机制进行评估。根据实验数据来说明在不同的情况下银行将如何选择排队策略。

1.2　算法思路

1.2.1　针对问题1

1.2.1.1

第一问在已有的代码框架基础上增加了对顾客身份的描述isVip，顾客身份采用随机数生成，为了数值较为合理，vip客户的概率设置为25%。

将1号窗口设置为vip窗口，只有vip能够选择该窗口,非vip用户无法选择。以此达到使vip等待时间更短的效果。

1.2.1.2

第二问主要是在RunSimulation函数里修改。根据已有代码流程。可知，是以时间为唯一排序标准对优先级队列进行出列操作的。在代码中，处理优先级队列中的arrival事件的同时，将生成下一个arrival事件（在模拟时间内），处理departure事件时将计算等待时间，这是通过finishServive的增减来实现的。

为了达到vip插队的效果，我们应该改变满足条件的vip的离开的发生时间，并且相应地改变被插入队列中的其他离开事件的发生时间和等待时间。由题给条件可知，插队要满足：是vip用户，是普通窗口，窗口前面有未开始服务的非vip用户这几个条件。而且插队不会改变到来事件，对于没有完成的业务，会继续处理。

我们可以简单地将其概括为，当处理可插队的vip用户的arrival事件时，生成该事件对应的departure事件，和对其他departure事件进行调整。可以用临时的优先级队列来进行管理。

1. 先将队列中的事件全部出列进入临时队列里，寻找是否有普通窗口前未开始服务的普通用户，若有则插队成功。再把临时队列里的恢复到原优先级队列中。
2. 再判断是否插队成功了，插队成功则再次遍历，对优先级队列中的非vip的离开事件进行时间和等待时间的调整（加上插队的vip的服务时间）。
3. 插队不成功或者不能插队，都进行一般的离开事件的生成。

1.2.2　针对问题2

对于vip限时等待，由题意我们可以知道当某一出纳员服务完当前用户后，将会重新计算所有排队的用户中的VIP用户的等待时间，如果等待时间已到达最长等待时间，将优先服务等待时间长的VIP用户。

由题意可知，将在某一出纳员处理离开事件的时候，要判断当前优先级队列中所有vip的已经等待了的时间。

这里可以通过优先级队列中vip的离开事件减去vip服务时间再减去等待时间得到vip到达事件的时间。如果该等待时间已经超过了容忍程度，则为此vip用户分配该出纳员立即服务。这里需要注意：

1. 改变vip的离开事件的生成事件和等待时间
2. 对于被插队窗口，此后的离开事件的发生和等待时间都增加
3. 对于vip离开的窗口，此后的离开事件的发生和等待时间都将减小

同样的我们可以通过对优先级队列的出列入列来实现。

1.2.3 针对问题三

问题 3 为开放性思考题，即综合考虑各类角色的等待时间，被插队的普通用户越多，普通用户越不满；VIP 用户等待时间越长，VIP 用户越不满；银行同时也不希望窗口的空闲。

针对普通用户：不满意度被定义为设置某vip等待策略后多等待的时间与原时间的比值。

针对vip用户：不满意度被定义为实际等待时间与预计等待时间之间的差值和预计等待时间的比值。

针对银行：如果银行有窗口空闲，可以看作银行的损失，作为银行不满意度的评价标准，因此可以通过窗口的平均繁忙程度来反映方案优劣。

对于以上三种策略，还有不设置vip窗口的情况，将对同样的模拟时间，窗口数，到达时间，时间服务时间（对最后一种策略还有等待时间）进行十次模拟。可以得到近似的解。

第2章　实验结果

2.1　实验数据及实验设计

Vip用户数量不能过多，不然不合理，故设置为25%。

认为vip用户的容忍等待时间是服务时间的平均值。比如对于服务时间[10,20]，那么vip所容忍的等待时间就是15。

由于为随机事件，能设置的只有模拟时间，出纳员的数量，到达时间和服务时间。下面将对每一问分开讲。

对于第一题的第一问，题目给定了用例，为了更好的模拟真实情况，在后面增加了模拟时间更长的用例来看vip的优越性。

对于第一题的第二问，实现vip插队，选择了一个较长的时间500，有4个窗口，其中1号为vip窗口，到达时间为[3,6]，而所需服务时间为[25,30]。这是为了在此时间内能够发生vip需要插队的情况，以及模拟真实银行行为（窗口不会太少，时间不会太短）。

对于第二题，实现vip限时等待，选择了一个较长的时间500，有4个窗口，没有vip窗口，到达时间为[3,6]，而所需服务时间为[25,30]，而vip的所容忍的等待时间设置为所需服务时间的最大值，符合常理。

对于第三题，对三种策略和不设置vip窗口进行横向对比，由于数据具有随机性，故对每种策略以相同的初始条件重复五次，然后统计vip等待时间，非vip等待时间，窗口忙碌程度。

2.2　实验结果

2.2.1　实验结果展示与分析

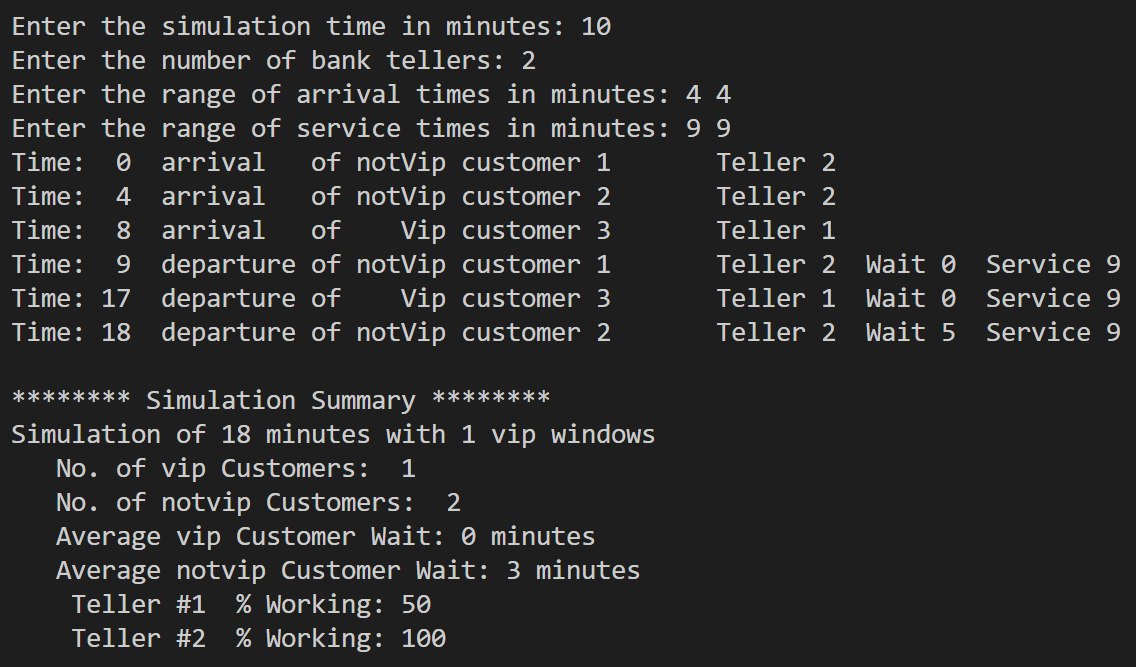
第一题第一问

Num\_vip\_windows=0

图形用户界面, 文本

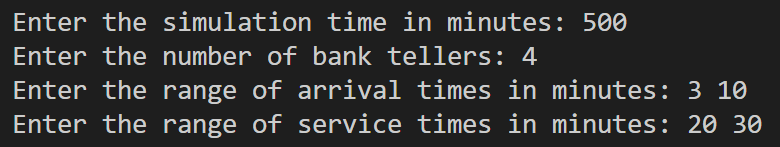
描述已自动生成

Num\_vip\_windows=1

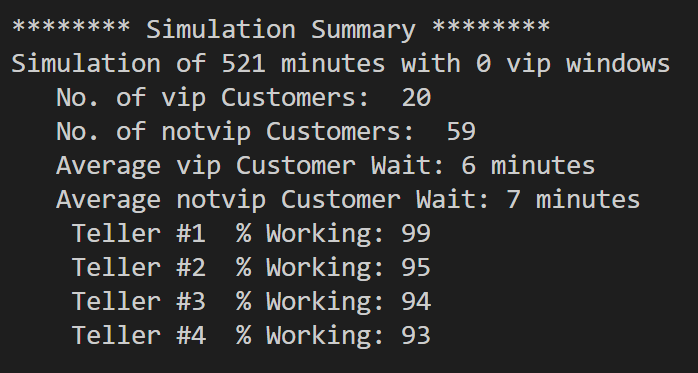


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 到达队列情况 | 总用时 | 繁忙程度1 | 繁忙程度2 | Vip | 非Vip |
| 增设前（NNV） | 18 | 100 | 50 | 1 | 0 |
| 增设后（NNV） | 18 | 50 | 100 | 0 | 3 |

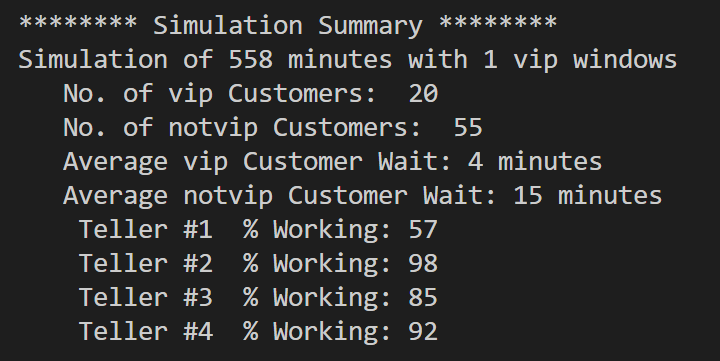
对于同样的，模拟时间较长的，初始条件如下的：



在Vip窗口数量为0的情况下



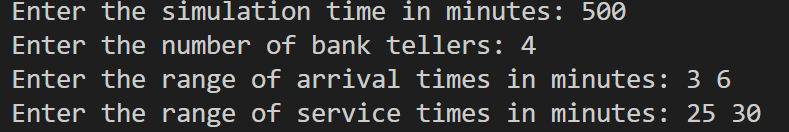
在Vip窗口数为1的情况下：



可以见得加了vip窗口后，vip等待时间减少了许多。

第一题第二问

数据如下



由下可见64号vip插队成功，来得比非vip58号晚，但是比他离开的早。

日历

中度可信度描述已自动生成



下为此次模拟的汇总，可见vip等待时间为非vip的63%，vip等待时间比非vip小许多。

文本

描述已自动生成

第二题

文本

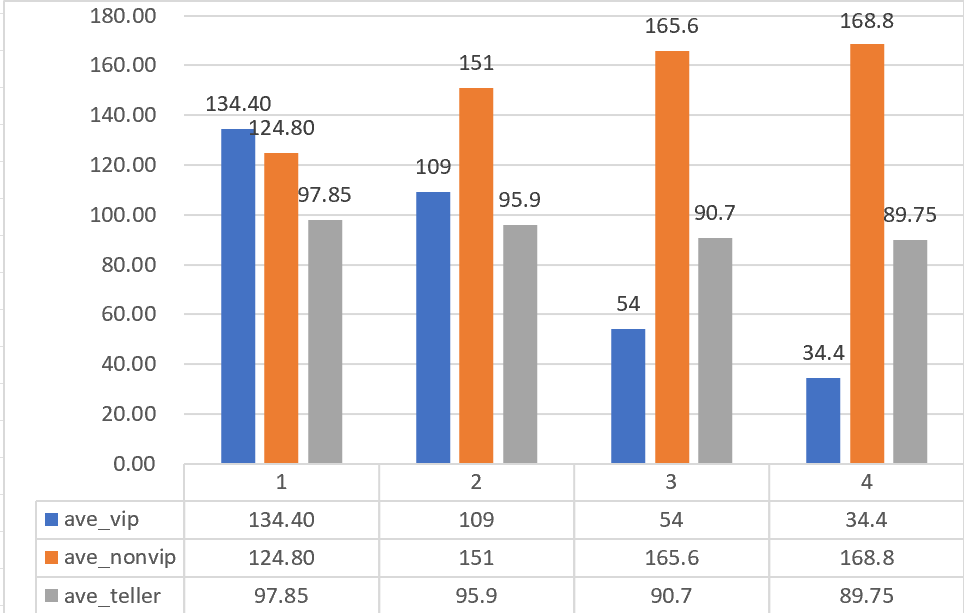
描述已自动生成

由下可见vip限时等待插队成功

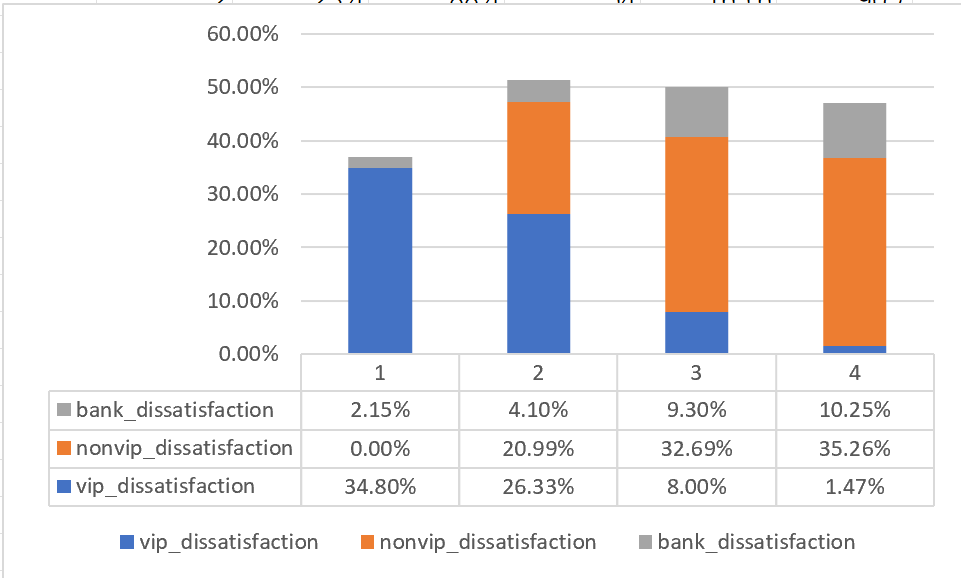
图形用户界面, 文本

描述已自动生成

第三题



1为未设置vip策略，2为设置vip窗口，3为vip插队，4为vip限时等待。



可见对于vip，vip限时等待最有利；对于非vip则是不设置vip窗口不满意度最低；对于银行，不设置vip窗口对于银行的时间利用率最有益。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | vip\_dissatisfaction | nonvip\_dissatisfaction | bank\_dissatisfaction | total |
| 1 | 100.00% | 0.00% | 20.98% | 42.10% |
| 2 | 75.67% | 0.60% | 40.00% | 34.57% |
| 3 | 22.99% | 92.72% | 90.73% | 64.63% |
| 4 | 4.21% | 99.99% | 100.00% | 61.68% |

计算方式：

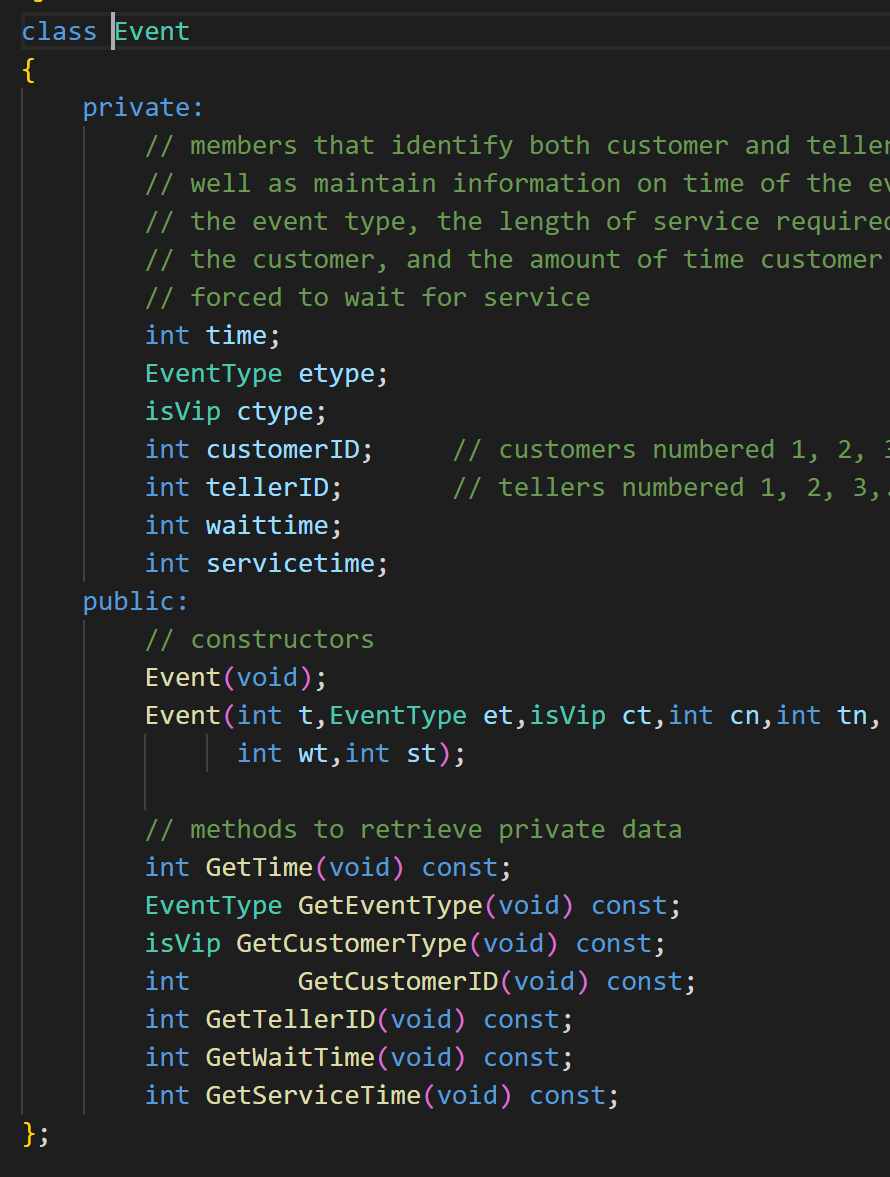
对于银行，如果更在意vip的感受，那么选择vip限时等待最有利,但是这会造成非vip用户的极大不满；如果选择不增加vip窗口，会造成vip的不满；综合考虑，可能增加vip窗口是比较合适的做法。

第3章　源代码解析

3.1.1第一题第一问

在Event类里，增加了对vip身份进行判断的私有成员。





在Simulation类中增加了对于顾客身份生成的函数。为了合理化，在这里设定vip的概率为25%。

文本

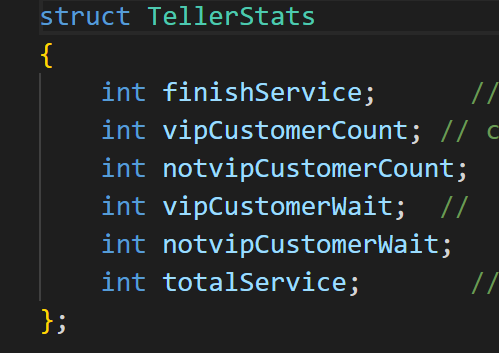
描述已自动生成

在simulation类里修改了对于下一个可用窗口的选择，对于vip可以选择所有窗口，而对于非vip，则只能选普通窗口。

文本

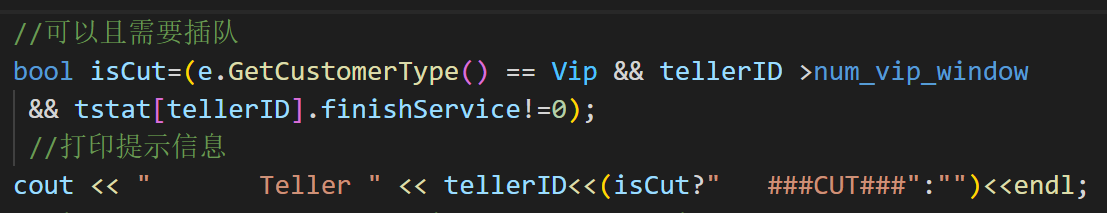
描述已自动生成

为方便打印输出，还修改了TellerStats,其中对vip和非vip进行了分开处理。

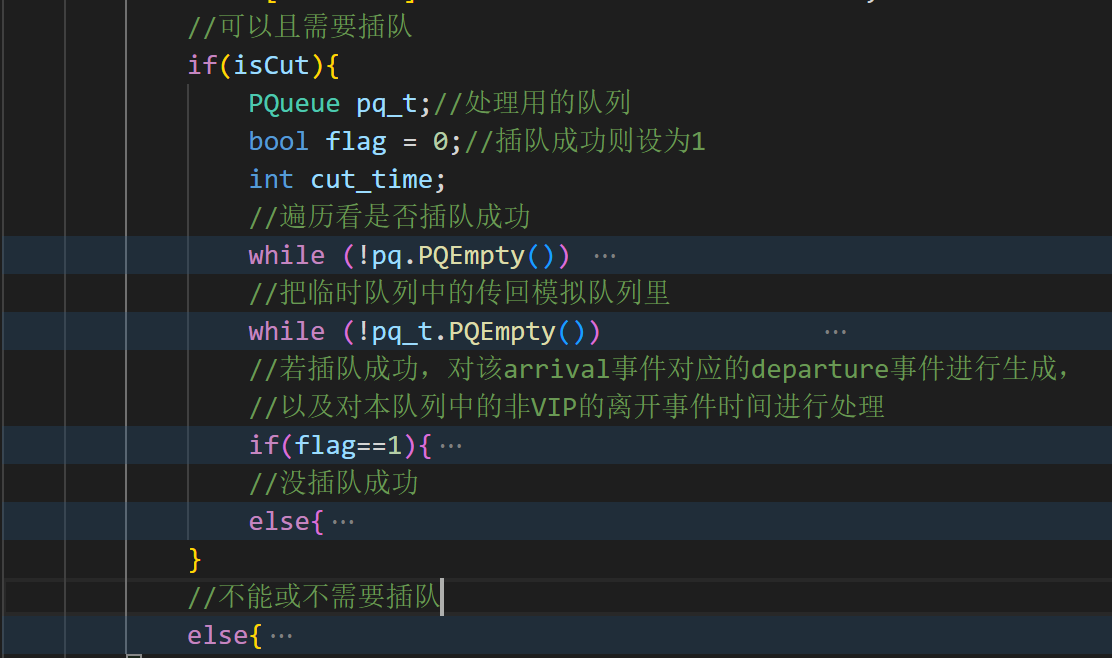


3.1.2第一题第二问

是在上一问的基础上，对RunSumulation函数进行了修改

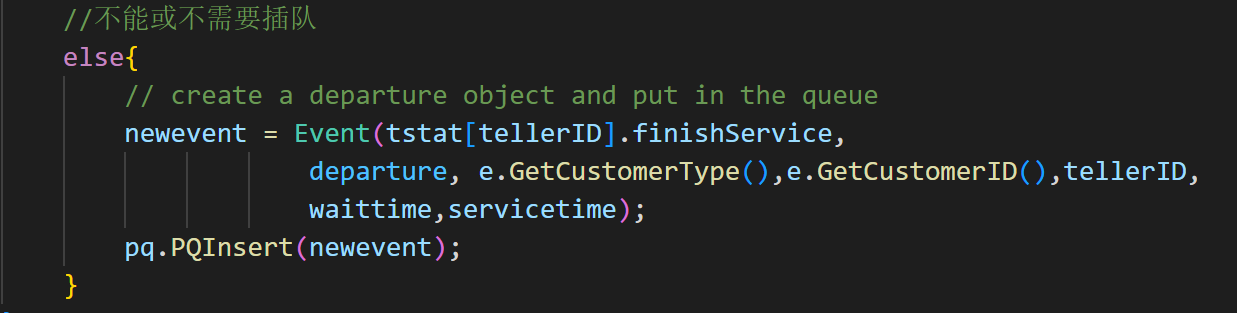


框架如下：



具体实现如下，主要思想是通过模拟队列和辅助队列之间的出列入列，对是否满足插队条件进行判断，并且实现对于插队后相关事件的时间改变。若不满足插队条件则与原处理相同，也就是生成相应的departure事件。





3.2第二题

思想与上一问一样，也是通过辅助的临时队列来对符合条件的元素进行修改。

先遍历一遍找到等待时间最长的vip。

电脑萤幕的截图

描述已自动生成

再把tmp里的Event元素放回pq中，同时对于等待时间最长的vip进行离开事件的改变处理。把flag即本次是否有vip插队，置为1。电脑屏幕的截图

描述已自动生成

若有VIP插队，则对vip插队的队列和vip离开的队列的离开事件进行更新。然后在放回优先级队列pq里。文本

描述已自动生成