**数据结构与算法课程实验报告**

|  |  |
| --- | --- |
| **实验名称：** | **银行VIP窗口设计** |

|  |  |
| --- | --- |
| **学员姓名：** | **李玉良** |
| **学 号：** | **202102001025** |
| **年 级：** | **2021级** |
| **学 员 队：** | **计算机学院四队** |

摘　要

问题一的设计要求是增设VIP窗口，根据约定设置1号出纳员的柜台为VIP柜台，只有VIP用户可以在VIP柜台办理业务，用户一旦选择排队不再更换队伍。所以为一个用户的到达和离开事件增加一个VIP属性，然后再更改根据事件获得的窗口序号（加入VIP属性对其进行处理），最后在打印时，打印出顾客是否是VIP，实现上述功能。

问题二要求是实现VIP用户对普通用户的插队，插队的VIP用户也需要遵循先后顺序，可以更换柜台插队，VIP用户存在一个最大等待时间。核心思想是，对于普通用户不作特殊处理，选择结束时间最早的队伍正常排队。对于VIP用户，首先为其选择最短的队列，然后计算VIP的等待时间，如果不超过最长等待时间则正常排队；如果超过最长等待时间，则将该VIP放在一个待插队的队列中进行等待。当有其他用户离开时，对该用户所离开的队列进行检查，如果不存在VIP，则将待插队的VIP队列中的一名最早到达的VIP取出，将其插入到该队中，将该队中后续的用户的等待时间与离开时间进行更改，加上该VIP用户的服务时间。 问题三要求最佳的策略，银行不希望有空闲的窗口但是其他窗口人在排队；普通用户不希望被一直插队；银行的VIP用户不希望等待太久；为了平衡这三者，在基于第二问的基础上，做出如下改进：首先对于窗口选择，普通用户在特殊情况下也可以选择VIP队列，即VIP窗口没有人排队的情况和普通队列人排满但是VIP窗口未排满的情况。其次对于普通用户来说，为了衡量其不满意程度，为用户增加一个愤怒程度，当用户被插队的次数过多时，愤怒值高于阈值的队列禁止插队。同时，如果所有队列愤怒值都高于设定的一个阈值，那么就等待走了一定的顾客，使得愤怒值降低之后才允许插队。

关键词：优先级队列，银行窗口，VIP窗口，VIP插队

第1章　实验原理

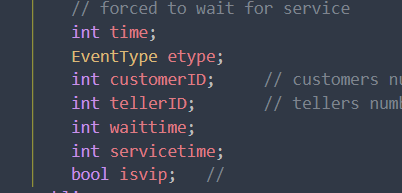
1.1　实验内容

实验总体是在一个已有的框架上，在问题一、二、三规定的情景下去模拟用户的排队和插队的过程。问题一实现了事件的VIP属性，增设了VIP窗口。问题二，为了使得VIP可以插队，按VIP的等待时间大小进行了分类，会超时的VIP被分配到一个等待插队的队列中，待到寻找到合适的可以被插队的队列时，在上一个用户办理完之后，插队进去。问题三是寻找一个最优的策略，在银行方面，我考虑的是窗口不能闲置且有人排队，同时也需要保证普通队伍满了但是VIP队列没有满时允许普通用户去排VIP队伍。

1.2　算法思路

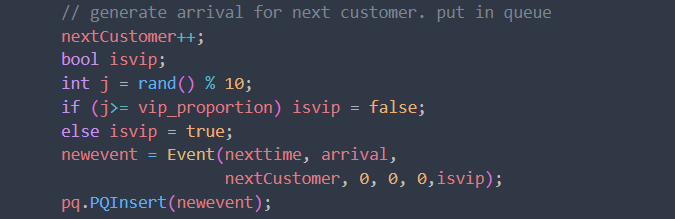
1.2.1　针对问题1

（1）. 增加VIP属性



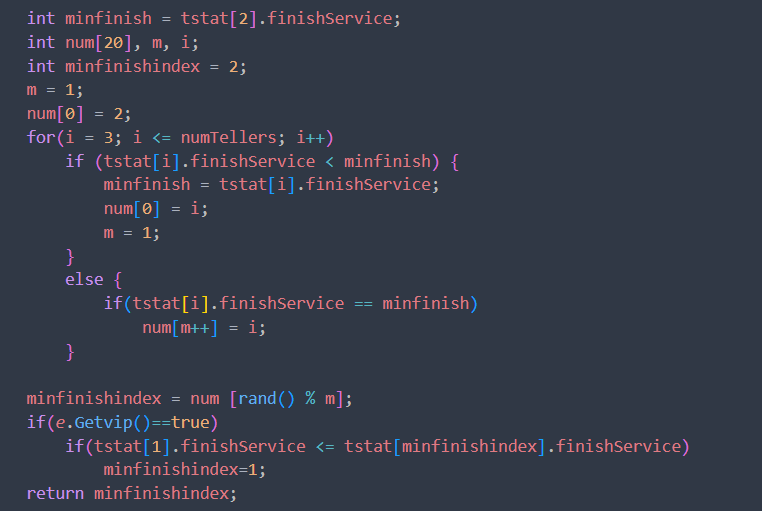
（2）. 设定VIP比例j / 10，随机生成VIP

：生成用户到来的新事件时，采取rand()函数决定是否是VIP



（3） 根据用户到来事件，生成用户所去到的窗口（NextAvailableteller()函数）

通过一个比较选择最快结束的一个teller，如果有多个最短一样的，那么就随机选择一个窗口。最后，比较：如果用户是VIP，并且此时VIP队伍的排队时间比原来选择的队伍短的话，那么就将选择的柜台号改为1号VIP窗口。



（4）. 打印信息

就是在打印的时候打印出是否是VIP的属性即可。

1.2.1.2

约定VIP用户可以在VIP窗口排队等候，或者在普通窗口队列中插队（即排到普通窗口队列最前面）。约定插队的VIP用户在普通窗口中按照到达顺序先后进行排队。约定VIP用户选择窗口的优先级为：空VIP窗口>空普通窗口>插队到普通窗口队列。

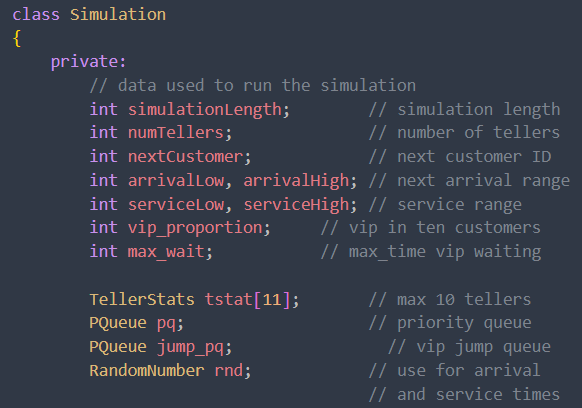
这一小问中，要求VIP如果没有排在VIP窗口，且普通窗口的队伍有人再排队时，需要直接插队，而这与第二问的要求有重复之处，只需要对于相同的测试用例，将VIP的最大等待时间改成0即可。

1.2.2　针对问题2

1. 新增变量或成员

需要插队的VIP队列：jump\_pq ：定义在Simulation类中

VIP的最长等待时间：max\_wait



1. 当有用户到来时，判断是否是VIP且需要插队



如果需要插队的话，则暂时不进去排队的队列中去，同样也不必生成离开事件，等待插队的时机。

1. 插队时机

该段代码位于判断当前取出的事件是离开事件时，代表此时有柜台服务完成了一个顾客，那么就可以让VIP插队，但是插队的前提是此队列中不存在其他VIP客户，否则可能导致后续的VIP本来不需要插队但是却需要插队了。



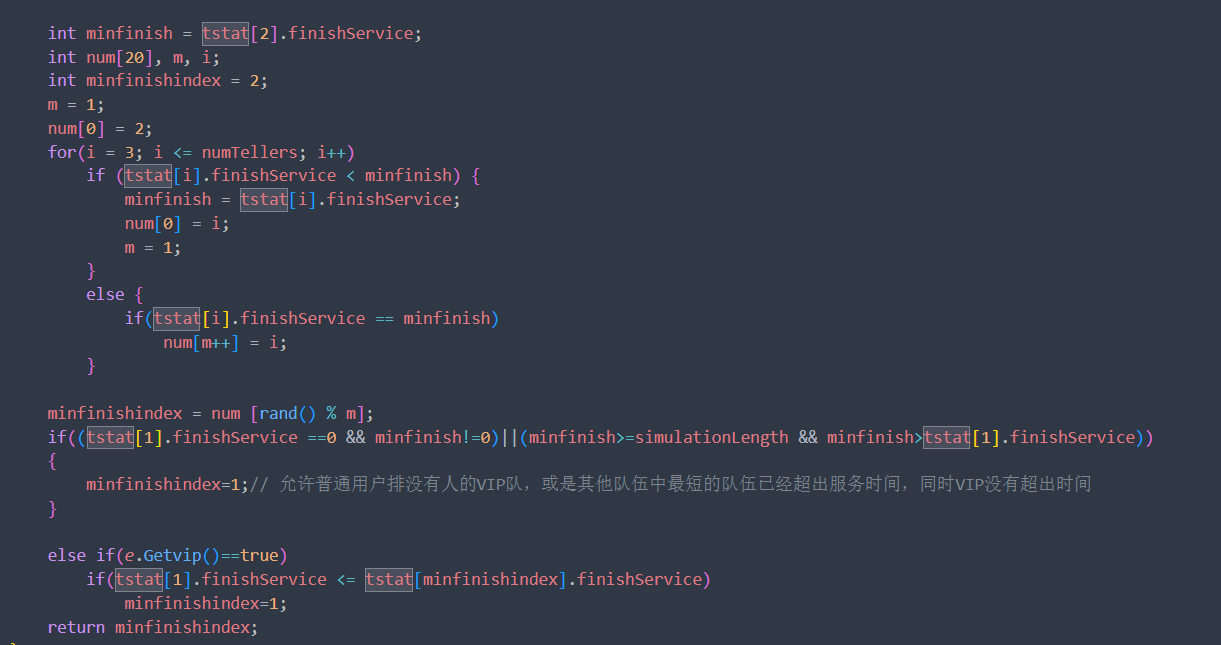


插队之后，将生成对应的插队VIP的等待时间，创建离开事件加入到队列中，同时，将该队列后续的所有用户的事件的属性进行更改。

1.2.3　针对问题3

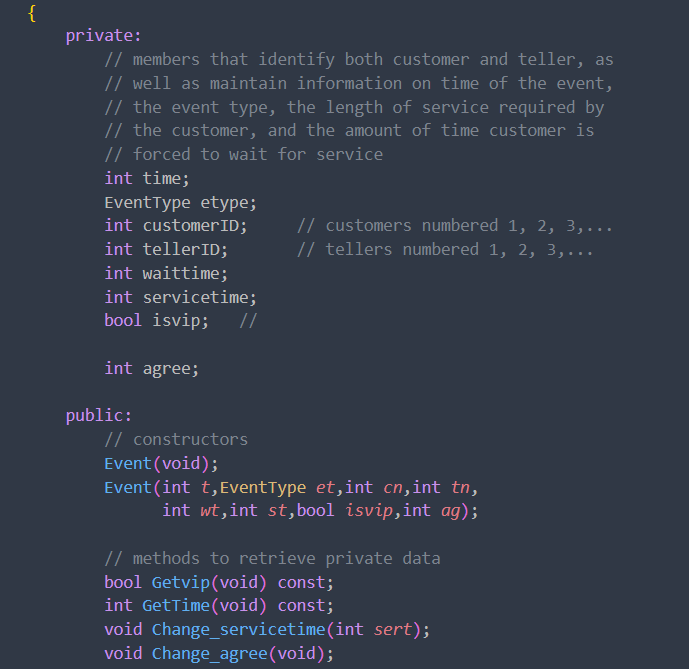
1. 窗口选择的改进

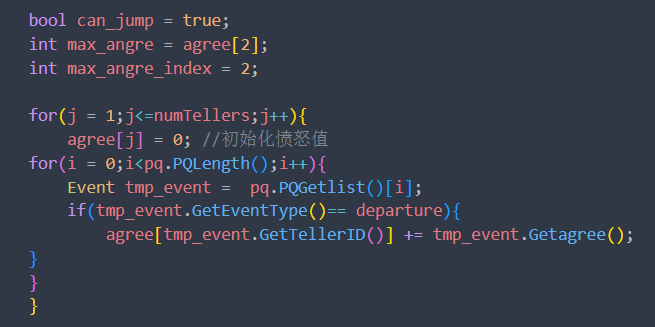
相比于问题二中的窗口选择，新增了一个判断条件，如果满足两个要求之一，那么就将普通用户的选择改为VIP窗口。



（2） 新增用户的愤怒程度

每次被插队一个都会增加愤怒程度，同时在每次的模拟中，为所有柜台增加了一个用户总愤怒程度，如果有用户被插队，那么该队伍中所有的用户的愤怒程度都会加一，如果有被插队的用户离开了队伍，那么也会从该柜台的总愤怒程度中减去离开的用户的愤怒程度。





（3） 插队的VIP需要考虑更多条件

除了该队列中是否有VIP，还有该队列的用户的愤怒值是不是最高的，如果是；则不能插该组用户的队，这是为了保证VIP不会连续地插同一个柜台的队伍。



第2章　实验结果

2.1　实验数据及实验设计

问题一，输入测试用例，只要能正确地计算出VIP用户与普通用户区别的离开时间。因此选择一定的VIP比例，使得有用户能排到VIP队伍中，同时，即使VIP的队比较短，普通用户也不能排VIP的队。

测试用例： 10 2 2 2 8 8 3

第二问：VIP用户直接插队的测试：

在问题二的代码中，直接将VIP最大等待时间改成0即可。

问题二： 要能够体现VIP插队的效果，故测试用例应该控制好固定的时间间隔以及服务时间，服务时间要够长以及VIP的等待时间必须足够短，因此就会出现插队现象，再通过插队的逻辑去计算打印结果是否正确。

测试用例：480 4 5 5 24 24 4 10

尽量测试时间够长使得VIP插队能够发生且发生的有一定数量

10min服务时间，2个柜台， 2min到来一个顾客，每个顾客服务10min ， vip顾客的比例是2/5 ， VIP最大等待时间为5min

问题三： 寻找一个最优的策略

问题分析：平均服务时间与平均到达时间是两个比较关键的参数，在窗口数固定的情况下决定了平均等待时间的期望值。例：一个窗口的情况下，2min来一个顾客，4min服务完一个顾客，那么平均等待时间，第一个顾客0min，第二个2min，第三个4min…以此类推…如果增加到两个窗口，那么平均等待时间应该是0min。

因此，在所有柜台都尽量不空闲的状态下，可以得出的粗浅的规律是：

服务时间/柜台数 = 平均服务时间。

平均服务时间 - 平均到来时间 = 新来的一个顾客需要等的时间（设为T0）

而这个新顾客需要等待的时间是累加的！！！！（来的顾客越多这个等待就越恐怖，当然这个数量也会被最长等待时间限制），但是理论上这个还是应该越小越好，这样才不至于太长的总等待时间，也可以让银行服务更多的顾客，当然这是在所有柜台的工作率100% 的情况下估计的，所以实际情况会更糟。但是我们仍然可以用该方法来估计一个大概的等待时间。

设顾客总数是N，新顾客等待时间是T0。I为顾客ID。

所以理想状态下的平均等待时间应该如下：

T\_average = / N

所以只要柜台的工作率足够，按道理来说这个等待时间就不会太低，所以更关键的还是用户的体验以及VIP用户的体验，VIP则在于VIP的最长等待时间，用户的则在于用户被插队的次数，即愤怒值。都是越低越好。

测试用例：480

4 5 5 24 24 5 10

与第二问测试集相同

2.2　实验结果

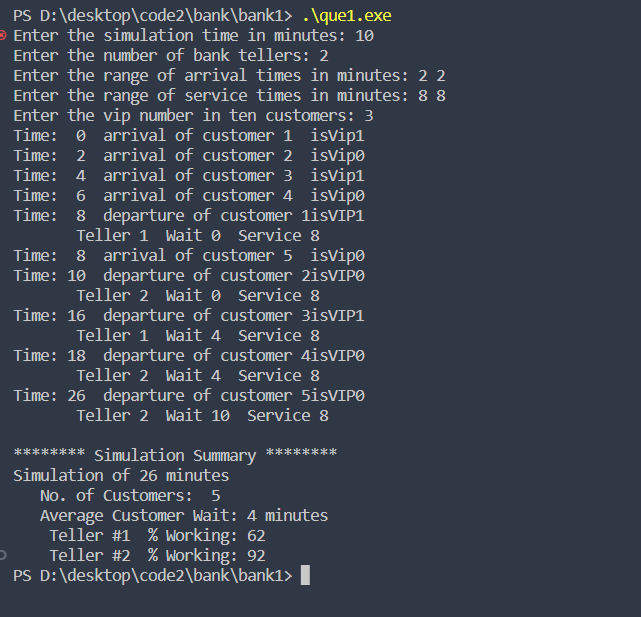
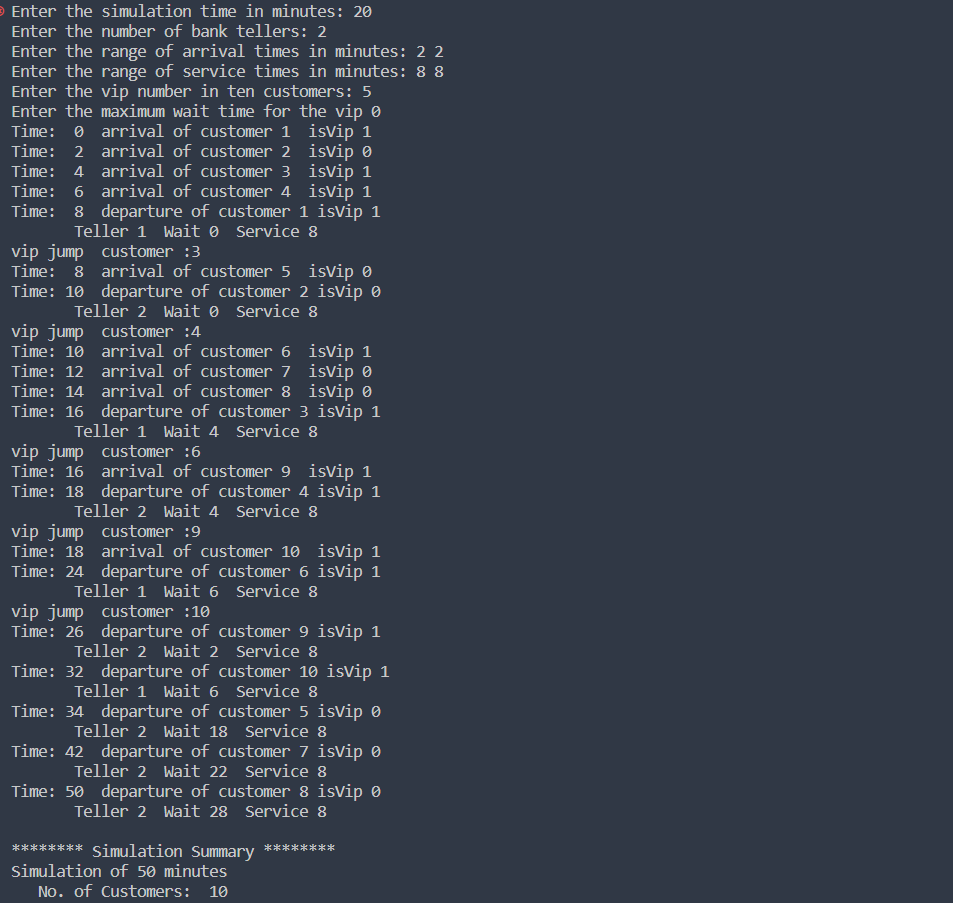
2.2.1.1　实验结果展示

|  |  |  |  |
| --- | --- | --- | --- |
| 到达队列情况 | 总用时 | 1号繁忙程度 | 2号繁忙程度 |
| PPV | 18 | 100 | 50 |
| PPV | 18 | 50 | 100 |

虽然看似繁忙程度没有变化，但是，实际用户总等待时间变大了。

2.2.1.2

将最大等待时间设为0，直接让VIP插队即可



设置两个窗口，到来时间固定2，服务时间固定8，

顾客1是VIP。0时刻到来，排在1号口，8时刻离开，没有等待

顾客2是普通用户，2时刻到来，排在2号，10时刻离开，没有等待

顾客3是VIP用户，4时刻到来，随机排在1号口，此时有1在服务期间，产生等待，等待时间为4.

顾客4是普通用户，6时刻到来，排在普通窗口2处，此时前面有2号顾客在处理，等待也是4

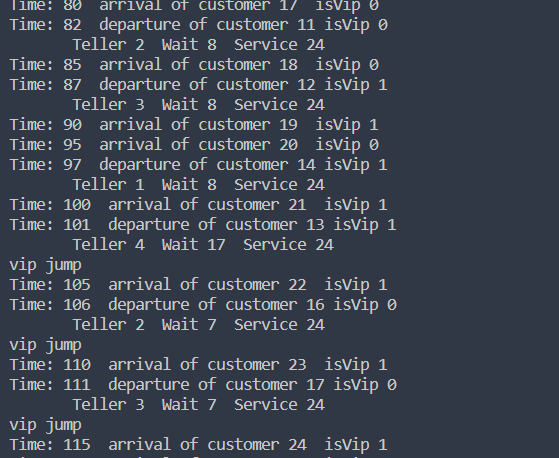
顾客5是普通用户，8时刻到来，此时1正好离开，但是只能排在前面有两个顾客的2号口，等待时间为4的服务时间和2剩余的服务时间，一共是10

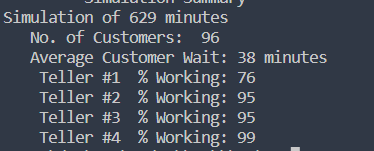
工作效率 = 服务时间/总时间

Work1 = 16/26 = 0.62

Work2 = 24/26 = 0.92

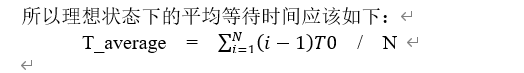
2.2.2　实验结果分析





因为结果太长，所以在打印时将插队信息打印出来。

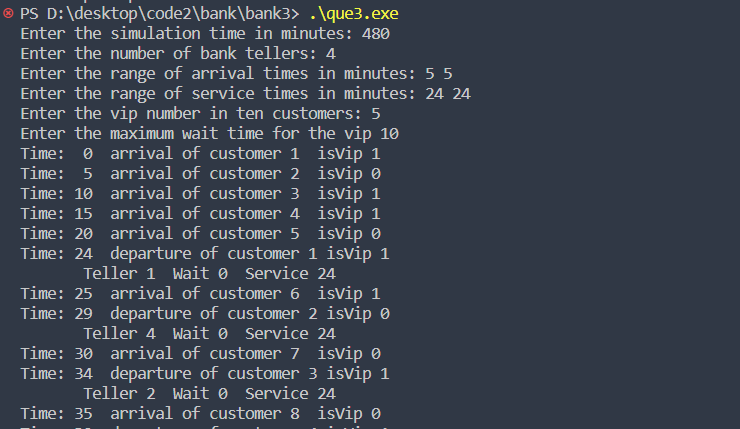
最后的总结果基本符合，其他窗口都是几乎一直在工作的原因就是有VIP一直插队，很难空闲，而VIP窗口普通用户无法排队。

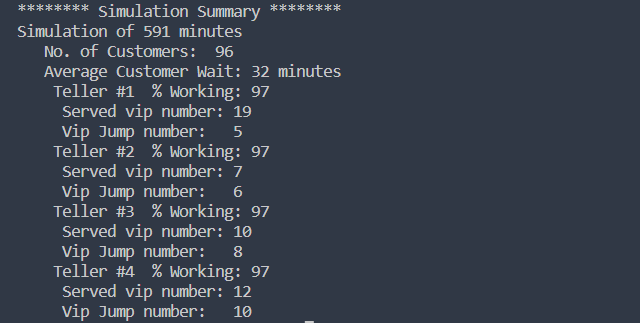


代入数据计算，粗略估计的最大平均等待时间为48，而实际等待时间为38

说明该方法是有优化作用的。

2.2.3　实验结果分析





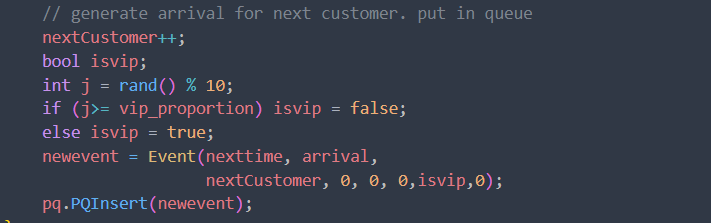
同样的测试集，平均等待时间，较第二问更少了，说明引入排队策略和愤怒值是由效果的。

第3章　源代码解析

作为提交的代码的说明，简述每一段核心算法代码是如何通过C++语言实现的

* 1. VIP生成

按比例生成VIP，随机生成



* 1. 为用户选择窗口

首先在第二个窗口，往后依次选择，选择一个最短结束时间的窗口，然后比较是否是VIP，如果不是VIP，而且若此时VIP窗口是空的，或是普通窗口满了但是VIP窗口还没排满，此时普通用户都可以排到VIP。 然后如果是VIP用户，且VIP窗口的长度较短，那么就排到VIP窗口。

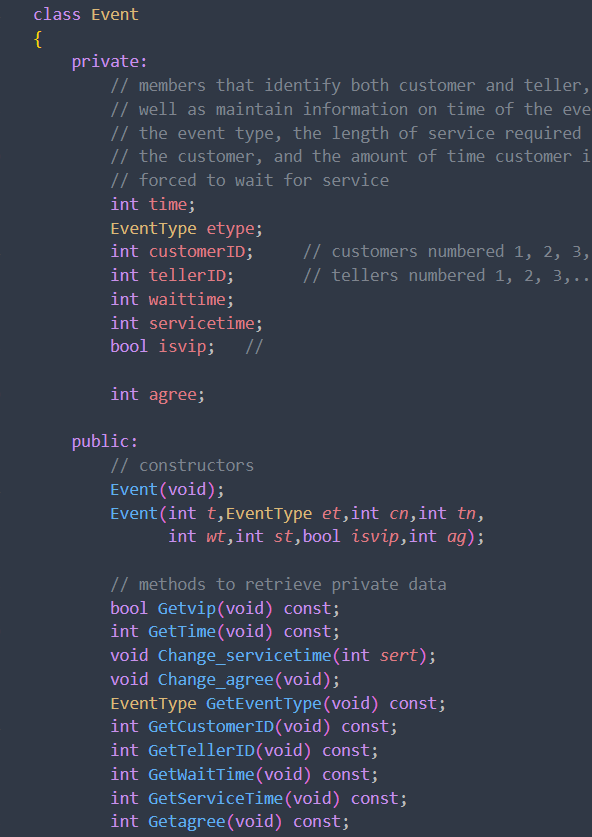


* 1. 给需要插队的VIP 加入待插队队列

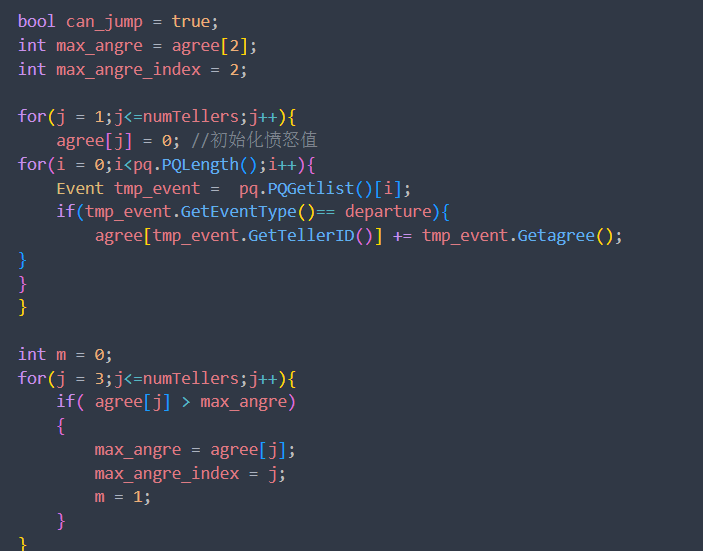
待插队的VIP加入插队队列的时候，只需要判断是否满足等待时间大于最长等待时间即可。需要注意的是要计算一下插队VIP的服务时间，因为最初生成的VIP到来事件是不包含服务时间的。



* 1. 给普通用户增加愤怒值（被插队次数）



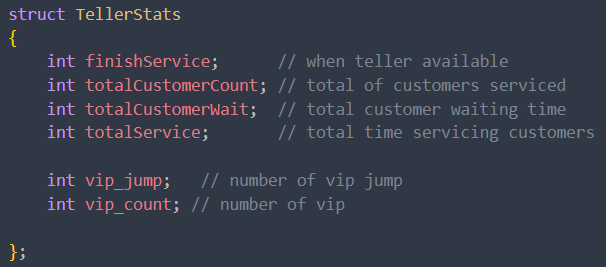
对于每次离开事件，对所有柜台的愤怒值重新计算，并且VIP需要插队时，寻找一个最合适的，愤怒值最少的柜台。







* 1. 为柜台增加属性，增加VIP个数和插队VIP个数，便于统计观察效果3



中间每次插队都会增加相应柜台的数字

