

BBM203 - ASSIGNMENT 2 - REPORT

Yusuf Emre Genç

21803663

1 Problem Definition

In this assignment, I developed an automation system of passenger ticket sales. The system divides passenger into 3 categories such as business, economy and standard. In business and economy classes, there are passengers have priority than others. The program handles this issue when selling tickets. Also the system supplies the maintenance of queues by transferring passenger to other queues.

2 Methods and Solution

I used stack data structure to hold data of seat that is not sold yet. Whenever program needs a seat whose class is already specified, the seat popped from stack. I also used queue data structure to be able to handle which passenger comes first. I implement priority queue structure based on linked list. So the program easily gets the next passenger and reorganize the queue. Meanwhile, by using priority queue, when a passenger come to queue and he/she has priority, the program inserts the new coming passenger to correct place. It also supports a dynamic structure so that if you have an infinite plane, you can use this program.

3 Functions

3.1 addseat

This function adds seat to specified flight's class which are given as parameters. If the flight does not exist, then the program automatically creates the flight and adds the seats to corresponding class. It also checks errors in the given command and it returns a meaningful integer value that indicates error type. The function prints the class and seats of that class data. In the implementation of this function, I used stack structure to store seats.

3.2 enqueue

This function handles the queue issue. Whenever a passenger comes to queue it checks the priority, class, name etc. and stores that passenger's data. The function checks errors in the given command and returns an integer value based on error of type or success. The function prints information about passenger and queue. In the implementation of this function, I used the priority queue structure. I used one priorities queue for each class's queue.

3.3 sell

This function takes passengers from queues and sells tickets. After running out tickets in a queue, it sends passengers to other queues in some conditions. The passengers who have priority lost their priority in other queues. After selling operation is done, the passengers go on waiting in queue, because new seats can be added again. This function handles all these issues and prints the information about remaining tickets. In the implementation part of this function, I used normal queue structure as well priority queue and stack structures.

3.4 close

This function tells the program that the flight is closed and there will not be any seat addition and ticket sales. After this function called, the flight turns into read-only mode. It also prints the passengers who cannot buy ticket.

3.5 report

This function gives an overview of what happened in given flight. This data includes which passenger in which class and how many tickets sold in classes.

3.6 info

This function gives data about passenger. This data includes flight code, the class passenger wished and the class passenger bought. The function also gives about passengers who can not buy ticket.

3.7 priorityQueueEnqueue

This function handles enqueue operation in a priority queue. It inserts the passenger to queue with respect to his/her priority.

3.8 Other Functions

There are also plenty of function which helps program to maintain and check errors.