Nicholas La Sala

CJ Mahoney

DSA Project Writeup

To complete the project we created to classes, the Plane class and the Runway class, without these classes the program would be complicated though it may still be completable, but it would be much more complex. The Plane class stores all the fields that define what a plane is as defined by the problem, and allows for this data to be retrieved as needed. The only field that can be changed while the program is the runway field, and that can only happen during the fifth method. The Runway class is a subclass that extends the queue ADT that was made in a prior lab, and adds a name field that can be retrieved to the existing functionality. This field is cannot be changed for an instance of the class during the runtime of the program.

To accomplish the required functionality for this program three resizable array based lists are created when the program is started, that represent the airport, the plane-lot, and finally a list of every flight number that is in the system be it on a runway, or in the plane lot. The airport list is of the Runway type, and the reason why the airport is implemented using a list of is so direct index access is possible so that planes can be enqueued to any runway without needing to remove runways from the airport. The planes list of the Plane type, and a list is used so that the user can have direct access to any plane in the lot during the third menu option, which is not possible with a queue, or stack. The reason why a list was created to store the flight numbers is because checking flight numbers against one ADT is much quicker than checking against every ADT. We could not find a reason why a Queue, stack, or double-ended queue would provide any advantage over the list ADT for any of these collections, whilst the disadvantages were apparent.

The menu was made using a switch inside of a while loop, and each case calls a function from the driver class, except for the eighth case which is just implemented within the switch itself. In each of these methods whenever an ADT needs to be traversed, a for loop is used to check the members of the collection, and while loop is used to ensure that valid inputs are being entered. The first option allows the user to enter a new plane into the system, by requesting them enter the flight number, destination, and runway, and then the constructor is called, and the plane is enqueued to the proper runway. Within the user enters their desired flight number, and runway the program checks that the flight number is unique, and that the runway is open by check against the number list and the airport list respectively.

The second option allows for the user to either allow the plane to either take-off or get moved to the plane lot. The program decides which plane’s turn it is by using a counter to track which runway will dequeue a plane from the front of the runway, and after the user enter valid choice the counter moves to the next runway at the airport. Regardless of the user’s choice a plane is being dequeue from the runway when this option is selected.

The third option allows the user to pick a plane by flight number from the plane lot, checks that the plane is in the lot, and then enqueues that plane into the proper runway, and removes it from the plane lot. This is accomplished using the direct access that the list ADT provides, which not only allows the program the check each planes flight number, but also allows the plane to be removed with ease.

The fourth option allows the user to open a new runway, a new instance of the Runway class is made, and the only restriction is that the user is not allowed to enter a name that another runway already has. When the program starts, and the user decides how many open runways the airport will start with this method is called.

The fifth option allows the user to close a runway by removing the it from the airport list and calling the set runway method for all the plane that were assigned to the runway both within the airport, and plane lot. To accomplish this both the airport, and the plane lot need to be traversed and checked so that every plane that is affected is properly reassigned.

The sixth and seventh options are accomplished in a similar way by traversing the appropriate collection and displaying the members of the collection for the user to see. The sixth option traverses the airport list and displays which planes are awaiting take-off, and the seventh option traverse the plane lot and displays which planes are awaiting clearance.

Rather than writing a method for eighth menu option, the value of the take-off counter is displayed within the switch itself. When the user is finished using the program they can enter the number 9, they will take from the menu loop, and the program will end.