Airport-program by Dilum De Silva.

I have implemented three classes for this program that will be used to simulate passengers queuing at a boarding gate and boarding an airplane. The three classes and their important fields and methods have described below. Include additional fields and methods as required to complete the solution.

1. **Passenger.** This class will contain the properties of a passenger relevant to our simulation.
   - **Fields** - firstName:String, surname:String, secondsInQueue:int
   - **Methods** - getters and setters for each field.

2. **PassengerQueue.** This class will represent the queue at the boarding gate.
   - **Fields** queueArray: Array of Passenger, first:int, last:int, maxSize:int
   - **Methods**: add(), remove(), display(), getMaxSize(), isEmpty(), isFull().

3. **Airport:** This is the main class that will drive the program.
   - **Fields:** passengers: Array of Passenger, queue1: PassengerQueue
   - **Methods:** main() and the other methods you implement from the menu.

I have created a menu system in the main() method of my Airport class which allows the user to choose which operation they want the program to do. Each operation has grouped in a separate procedure and the menu should allow for following operations:

- 'A' to add a passenger to the passengerQueue.
- 'V' to view the passengerQueue.
- 'D': Delete passenger from passengerQueue.
- 'S': Store passengerArray data into a plain text file.
- 'L': Load passenger data back from the file into the passengerArray.
- 'R' : Run the simulation and produce report (see details below).

Queue Details

Created my own queue object in my program and wrote my own methods within the object to add to queue and take from queue. Every time a passenger is added to a passengerQueue it will prompt for passenger details, then use the queue objects method to add the passenger to the queue. When the user selects to view, the details of all passengers currently in the queue will be displayed. The queue will be based on an array and hold 20 Passengers. When the queue items reach the end of the array they should be added to the start of the array (circular queue). If the queue becomes full then an error message should be displayed. Note: my solution is a console application (not windows).

For further I have created two queues and have each passenger join the shortest queue. Compare the different summary statistics with the single queue implementation.