Lab D

4. Implement the ADT queue by using a circular linked chain, as shown in Figure 24-12. Recall that this chain has only an external reference to its last node.

   - Recall that with a circular linked chain, you only have one reference to `lastNode`. To retrieve the first node, you can reference lastNode.getNextNode().
   - For full credit, this should be your only instance data variable.
   - Your class should implement `QueueInterface`, which I have provided.
   - Think about the special cases (namely empty lists and singleton list) when implementing methods.
   - I also provided a driver program to test your code.

   Refer to CircularLinkedQueue.java

5. Consider a new kind of queue that allows only a single copy of an object in the queue. If an object is added to the queue, but it is already there, leave the queue unchanged. This queue has another operation `moveToBack` that takes an object in the queue and moves it to the back. If an object is not in the queue, the operation adds it at the back of the queue.

   Create an interface `NoDuplicatesQueueInterface` that extends `QueueInterface`. Then write an array-based implementation of `NoDuplicatesQueueInterface`. Finally, write a program that adequately demonstrates your new class.

   - Use an array-based implementation.
   - For this ADT, if an object is added that is already in the queue, the queue will remain unchanged.
   - This ADT also has a method `moveToBack` that takes an object in the queue and moves it to the back of the queue. If the object is not already in the queue, it adds it to the back of the queue.
   - Your class should implement `NoDuplicatesQueueInterface`, which I have provided.
   - I also provide the driver program to test your implementation.
   - You might also find it helpful to review some of the files provided in previous assignments.

   Refer to NoDuplicatesArrayQueue.java