Cameron Moberg
CS260-02
September 9th, 2016

CS 260: Queue implemented by a LinkedList

The assignment was to create a Queue using a provided a LinkedList class. Since a queue can very easily be made using a LinkedList the first thing I did was think about how the methods relate to each other, for example, insertAtEnd is basically the same thing as adding to the queue. However some things had to be changed, for example for DeQueue, it is like deleting the first position in the list, however there isn't the right code in the LinkedList class, and since it would be too specific to change the LinkedList I decided to add some logic to the Queue class that would output that you can't dequeue an empty queue. Although I tried not to, I did have to add 2 getter methods to the CustomLinkedList class, "getStart" and "getEnd" since they were private variables, the only way I could get them so I could display in the CustomQueue class was by adding those methods, since it would be bad practice to change the private variables to public. I also added a getSize() method to Queue rather than just keeping the 5 methods (peek, enqueue, dequeue, isFull, isEmpty). The isFull method is practically useless because a LinkedList can virtually never be full, so I've made it so it always returns false.

Outputs: Read from left to right.

```
/usr/lib/jvm/java-8-jdk/bin/java ...
Custom Queue

Queue Options

1. EnQueue
2. DeQueue
3. Check Empty
4. Display
5. Peek
6. Get Size
7. Exit
1
Enter integer element to insert
10
Queue => 10
Front Pointer => 10
Rear Pointer  => 10
```

```
Queue Options

1. EnQueue
2. DeQueue
3. Check Empty
4. Display
5. Peek
6. Get Size
7. Exit

1
Enter integer element to insert
20
Queue => 10,20
Front Pointer => 10
Rear Pointer  => 20
```

```
Queue Options

1. EnQueue
2. DeQueue
3. Check Empty
4. Display
5. Peek
6. Get Size
7. Exit
1
Enter integer element to insert
30
Queue => 10,20,30
Front Pointer => 10
Rear Pointer  => 30
```

```
Queue Options

1. EnQueue
2. DeQueue
3. Check Empty
4. Display
5. Peek
6. Get Size
7. Exit
2
Queue => 20,30
Front Pointer => 20
Rear Pointer  => 30
```

```
Queue Options

1. EnQueue
2. DeQueue
3. Check Empty
4. Display
5. Peek
6. Get Size
7. Exit
2
Queue => 30
Front Pointer => 30
Rear Pointer  => 30
```

```
Queue Options

1. EnQueue
2. DeQueue
3. Check Empty
4. Display
5. Peek
6. Get Size
7. Exit
2
Queue is empty.
```

```
Queue Options

1. EnQueue
2. DeQueue
3. Check Empty
4. Display
5. Peek
6. Get Size
7. Exit
2

Cannot DeQueue an empty Queue.
Queue is empty.
```