

Credit
3110
Assignment
E4QueueList

How has your program changed from planning to coding to now? Please explain?

```
13 public class QueueList
14 {
15     private Node front;
16     private Node rear;
17     private int size;
18
19     public QueueList()
20     {
21         front = null;
22         rear = null;
23         size = 0;
24     }
25
26     // adds item to rear of queue
27     public void enqueue(String item)
28     {
29         Node newNode = new Node(item);
30
31         if (isEmpty())
32         {
33             front = newNode;
34             rear = newNode;
35         }
36         else
37         {
38             rear.next = newNode;
39             rear = newNode;
40         }
41
42         size++;
43     }
44
45     // adds item to front of queue
46     public void enqueueFront(String item)
47     {
48         Node newNode = new Node(item);
49
50         if (isEmpty())
51         {
52             front = newNode;
53             rear = newNode;
54         }
55         else
56         {
57             newNode.next = front;
58             front = newNode;
59         }
60
61         size++;
62     }
63 }
```

```
49
50    if (isEmpty())
51    {
52        front = newNode;
53        rear = newNode;
54    }
55    else
56    {
57        newNode.next = front;
58        front = newNode;
59    }
60    size++;
61 }
62
63 // removes item from front
64 public String dequeue()
65 {
66     String item = null;
67
68     if (!isEmpty())
69     {
70         item = front.data;
71         front = front.next;
72         size--;
73
74         if (front == null)
75         {
76             rear = null;
77         }
78     }
79
80     return item;
81 }
82
83 // checks if queue empty
84 public boolean isEmpty()
85 {
86     return size == 0;
87 }
88
89 // returns queue size
90 public int size()
91 {
92     return size;
93 }
94
95 // node class
96 private class Node
97 {
98
99     // returns queue size
100    public int size()
101    {
102        return size;
103    }
104
105    // node class
106    private class Node
107    {
108        private String data;
109        private Node next;
110
111        public Node(String d)
112        {
113            data = d;
114            next = null;
115        }
116    }
117 }
118 }
```

```
5 public class QueueListTest
6 {
7     public static void main(String[] args)
8     {
9         Scanner input = new Scanner(System.in);
0         QueueList queue = new QueueList();
1
2         int choice = 0;
3
4         while (choice != 4)
5         {
6             System.out.println("\nQUEUE MENU");
7             System.out.println("1. Add item");
8             System.out.println("2. Remove item (front)");
9             System.out.println("3. Display size");
0             System.out.println("4. Exit");
1             System.out.print("Enter choice: ");
2
3             choice = input.nextInt();
4             input.nextLine(); // clear buffer
5
6             if (choice == 1)
7             {
8                 System.out.println("Add to:");
9                 System.out.println("1. Front");
0                 System.out.println("2. Rear");
1                 System.out.print("Enter choice: ");
2
3                 int addChoice = input.nextInt();
4                 input.nextLine();
5
6                 System.out.print("Enter string: ");
7                 String item = input.nextLine();
8
9                 if (addChoice == 1)
0                 {
1                     queue.enqueueFront(item);
2                 }
3                 else if (addChoice == 2)
4                 {
5                     queue.enqueue(item);
6                 }
7             }
8             else if (choice == 2)
9             {
0                 if (!queue.isEmpty())
1                 {
2                     System.out.println("Removed: " + queue.dequeue());
3
4                 }
5                 else if (choice == 2)
6                 {
7                     if (!queue.isEmpty())
8                     {
9                         System.out.println("Removed: " + queue.dequeue());
10                    }
11                    else
12                    {
13                        System.out.println("Queue is empty.");
14                    }
15                }
16            }
17            else if (choice == 3)
18            {
19                System.out.println("Queue size: " + queue.size());
20            }
21        }
22        input.close();
23    }
24}
```