

Credit Name:CSE3110  
Assignment Name:QueueList

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

Step 1: Define the QueueList Class

- Declare the QueueList class, which represents a queue implemented using a linked list.
- Initialize the head node as null in the constructor.

Step 2: Implement the addatfront Method

- Create a new Node object with the provided string and set its next node to the current head.
- Update the head to point to the new node, effectively adding the item at the front of the queue.

Step 3: Implement the dequeue Method

- Update the head to point to the next node, effectively removing the front item from the queue.

Step 4: Implement the front Method

- Return the data stored in the head node, representing the front item of the queue.

Step 5: Implement the Top Method

- Return the data stored in the head node, representing the top item of the queue.

Step 6: Implement the toString Method

- Traverse the linked list starting from the head and build a string representation of the queue.
- If the head is not null, initialize listString with the first item's data.
- Iterate through the remaining items, appending their data to listString.
- Return the built string representation of the queue.

Step 7: Implement the Size Method

- Traverse the linked list starting from the head and count the number of items in the queue.
- Return a string indicating the count of items in the queue.

Step 8: Implement the enqueue Method

- Create a new Node object with the provided string.
- If the head is null, set the new node as the head.
- Otherwise, iterate through the linked list until the last node is reached and append the new node to the end.

Step 9: Implement the makeEmpty Method

- Set the head to null, effectively emptying the queue.

Step 10: Define the QueueListTester Class

- Declare the QueueListTester class, which contains the main method.

Step 11: Create a QueueList Object and Perform Operations

- Create a QueueList object named ql.
- Add items ("red", "blue", "yellow", "green") to the queue using ql.enqueue("...").
- Print the element at the beginning of the queue using System.out.println("Beginning of queue: " + ql.Top()).

- Print the number of items in the queue using `System.out.println("Items in queue: " + ql.Size())`.
- Remove the front item from the queue using `ql.dequeue()`.
- Print the new front item of the queue using `System.out.println("Front of queue: " + ql.Top())`.
- Print the updated number of items in the queue using `System.out.println("Items in queue: " + ql.Size())`.