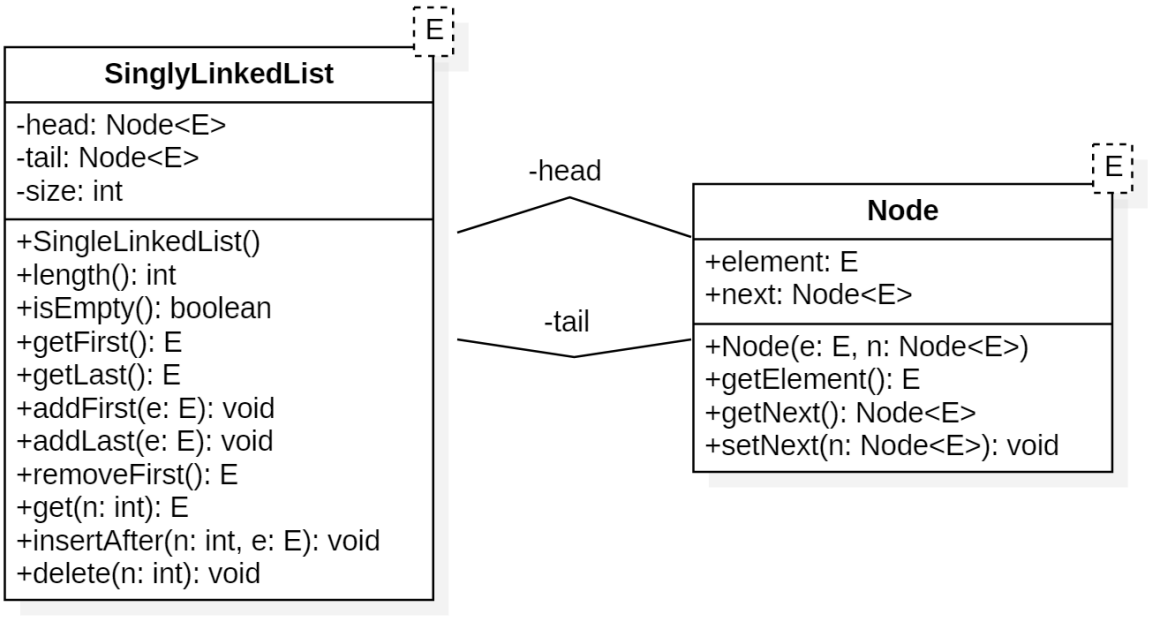
Linked List Implementation

1. Overview

The implementation of the Generic `SinglyLinkedList` that was covered in class is missing some important operations. Complete the implementation by proving the code for the methods described below. Test your code to make sure that your List is working properly.

The complete design of the class is shown below:



- `E removeLast()`
  Remove the last element from the List and return it. If the List is empty, return null.
- `E get(int n)`
  Return the n^{th} element of the List. The first element is number 0. If n ≥ length(), return null.
- `void insertAfter(int n, E e)`
  Insert the element e after the n^{th} element of the List. If n ≥ length(), do not insert the element. No error is returned or exception raised.
- `void delete(int n)`
  Remove the n^{th} element of the List (including n=0). If n ≥ length(), do not delete an element. No error is returned or exception raised.

When your implementation is complete, run it with the test driver EnhancedListTester.java (provided). Review the results to make sure that your modified class is working correctly. Your output must include your name.

2. Notes

- Start with the partially implemented code found with this assignment: SinglyLinkedList.java. The class Node is defined as a private inner class of `SinglyLinkedList`.
- Turn in only your modified source files: SinglyLinkedList.java and EnhancedListTester.java.
- Make sure your class is not in a package (that is, it is in the *default* package).
- You will not need to modify EnhancedListTester.java, except to add your name.

3. Required Main Class

EnhancedListTester, provided

4. Required Input

Not applicable

5. Required Output

No example is given. Review your results based on your understanding of List operations to ensure that your modified class is working correctly.