C:\Users\Rich\Documents\NetBeansProjects\Lab06\src\LinkedStackBad.java

```
<empty>
1
2 /**
                                                                                                                SinglyLinkedList()
                                                                                                                addFirst(E e)
3 * Using the adapter pattern to implement a stack as

  addLast(E e)

4 * a linked list. Each new element(last element added) is added as the last node and the
                                                                                                               equals(Object o) : boolean ↑ Object
                                                                                                               first() : E
5 * last node element is removed.(LIFO)
                                                                                                               isEmpty(): boolean
6 * @author Rich
                                                                                                               | last() : E
7 * @version 02/24/2017
                                                                                                               o removeFirst() : E
                                                                                                               o removeLast() : E
8 */
                                                                                                               size(): int
9 public class LinkedStackBad<E> implements Stack<E> {
                                                                                                                oString(): String ↑ Object
                                                                                                               head : Node <E>
      private SinglyLinkedList<E> list = new SinglyLinkedList<>();
                                                                                                               - 🧓 size : int
      public LinkedStackBad(){}
                                              //empty list
                                                                                                             ⊟ Node<E>
12
      public int size()
                                                                                                                 · ♦ Node(E e, Node<E> n)
13
                                                                                                                 getElement() : E
14
         return list.size();
                                                                                                                 - @ getNext(): Node<E>
                                                                                                                  setNext(Node<E> newNext)
15
                                                                                                                 · element : E
16
      public boolean isEmpty()
                                                                                                                mext:Node<E>
17
18
         return list.isEmpty();
19
20
      public void push(E element)
21
22
         list.addLast(element);
     public E top(){return list.first();} // home have been lipt last(), public E pop(){return list.removeLast();} O(N)
23
24
25
26 }
27
28
```