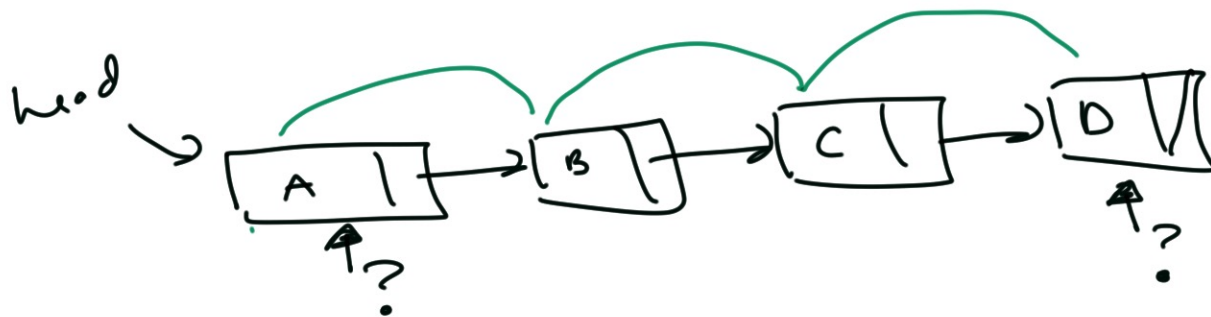


Stack Linked Chain - Based Implementation

- push, pop, & peek all access the top of the stack



- what should represent the top?

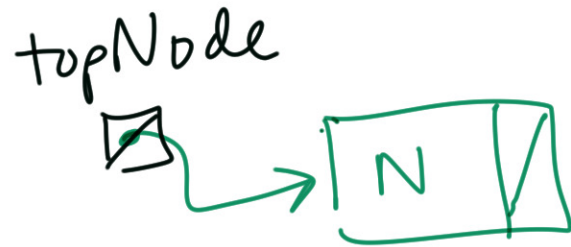
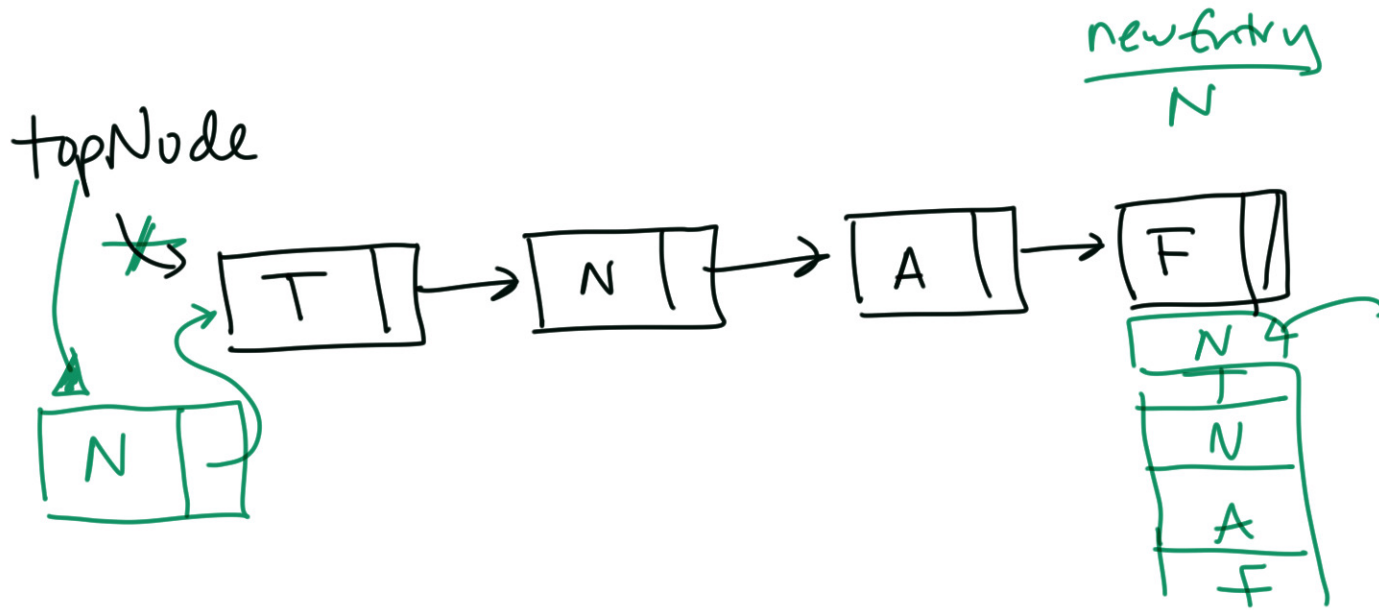


- design choices

```

10 public final class LinkedStack<T> implements StackInterface<T>
11 {
12     private Node topNode; // References the first node in the chain
13
14     public LinkedStack()
15     {
16         topNode = null;
17     } // end default constructor
18
19     /**@inheritDoc***/
20     @Override
21     public void push(T newEntry)
22     {
23         topNode = new Node(newEntry, topNode);
24
25     } // end push
26

```



↺

topNode



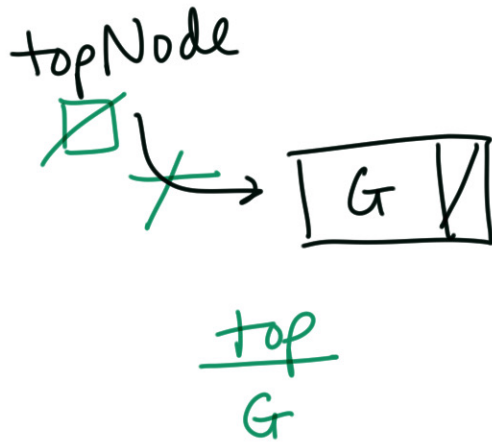
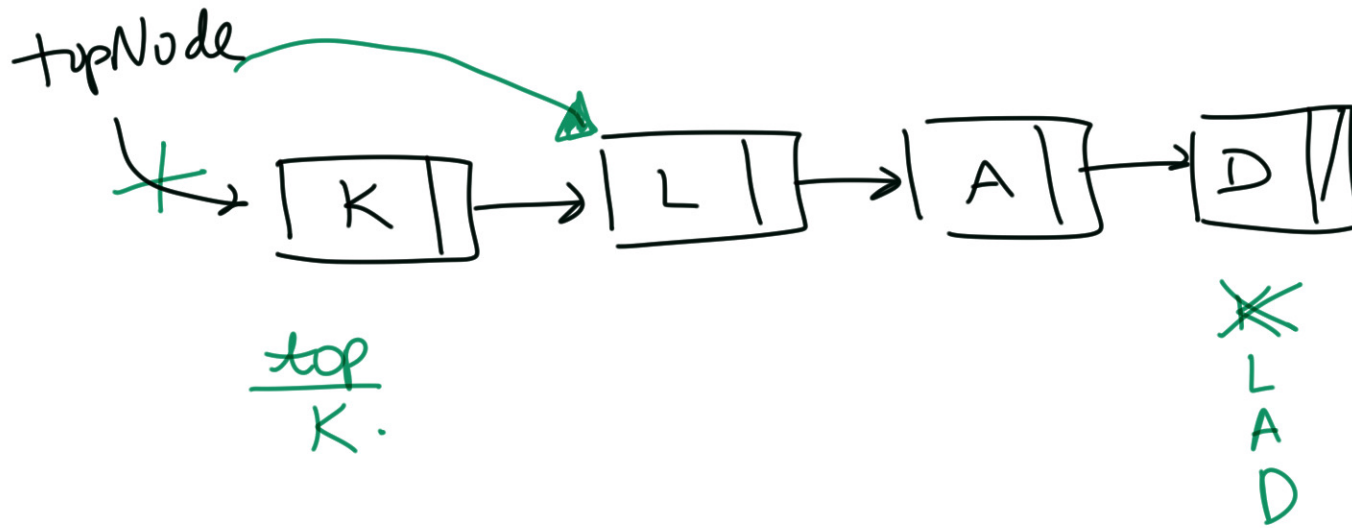
topNode



```

27     /**@inheritDoc***/
28     @Override
29     public T peek()
30     {
31         if (isEmpty())
32             throw new EmptyStackException();
33         else
34             return topNode.getData();
35     } // end peek
36
37     /**@inheritDoc***/
38     @Override
39     public T pop()
40     {
41         T top = peek(); // Might throw EmptyStackException
42         topNode = topNode.getNext();
43         return top;
44     } // end pop
45
46     /**@inheritDoc***/
47     @Override
48     public boolean isEmpty()
49     {
50         return topNode == null;
51     } // end isEmpty
52
53     /**@inheritDoc***/
54     @Override
55     public void clear()
56     {
57         topNode = null; // Causes deallocation of nodes in the chain
58     } // end clear
59
60

```



~~*~~