cohort44E / basic_programming / lesson_34 / code / demo20240529_GenericLinkedList / src / MyLinkedList.java

Andy179176 20240529 461fabc · 9 hours ago

110 lines (93 loc) · 2.3 KB

Code Blame Raw

```java
public class MyLinkedList<E> implements MyList<E> {

    private Node<E> head = null;
    private Node<E> tail = null;
    private int size = 0;



    @Override
    public boolean add(E element) {
        Node<E> node = new Node<>(tail,null,element);
        size++;
        if(tail!=null){
            tail.setNext(node);
        }
        if(head==null){
            head = node;
        }
        tail = node;
        return true;
    }

    @Override
    public boolean add(int index, E element) {
        return false;
    }

    @Override
    public E get(int index) {
        Node<E> node = getNode(index);
        return (node!=null)?node.getValue():null;
    }

    private Node<E> getNode(int index){
        if(index>=size || index<0 || head==null){
            return null;
```

```
37              }
38              int counter = 0;
39              Node<E> aktiveNode=head;
40              while (aktiveNode!=null && counter< index){
41                  aktiveNode=aktiveNode.getNext();
42                  counter++;
43              }
44              return aktiveNode;
45          }
46
47          @Override
48          public int size() {
49              return size;
50          }
51
52          private E remove(Node<E> node){
53              if(node==null){
54                  return null;
55              }
56              Node<E> prev = node.getPrev();
57              Node<E> next = node.getNext();
58
59              if(prev!=null){
60                  prev.setNext(next);
61              } else {
62                  head = next;
63              }
64              if(next!=null){
65                  next.setPrev(prev);
66              } else {
67                  tail = prev;
68              }
69              size--;
70              node.setPrev(null);
71              node.setNext(null);
72              E removedValue = node.getValue();
73              node = null;
74              return removedValue;
75          }
76
77
78          @Override
79          public E remove(int index) {
80              Node<E> node = getNode(index);
81              return remove(node);
82          }
83
84
85          @Override
86          public E remove() {
87              return remove(tail);
88          }
89
90          @Override
```

```
91          public E set(int index, E element) {
92              return null;
93          }
94
95          @Override
96          public String toString() {
97
98              if(head==null){
99                  return "[]";
100             };
101             StringBuilder sb = new StringBuilder();
102             Node<E> currentNode=head;
103             while (currentNode!=null){
104                 sb.append(currentNode.getValue()).append(";");
105                 currentNode=currentNode.getNext();
106             }
107             sb.setLength(sb.length()-1);
108             return "[" + sb.toString() + ']';
109         }
110     }
```