

```
1 package StackAndQueue;
2
3 class ImplementQueueWithLinkedList<E> {
4
5     private class Node<E>{
6         E val;
7         Node<E> next;
8         Node(E val) {
9             this.val = val;
10            this.next = null;
11        }
12    }
13
14    private Node<E> headNode;
15    private Node<E> tailNode;
16
17    public ImplementQueueWithLinkedList() {
18
19        headNode = null;
20        tailNode = null;
21    }
22
23
24    public void push(E val) {
25
26        if ( headNode == null ) {
27            headNode = new Node<E>(val);
28            tailNode = headNode;
29        } else {
30            tailNode.next = new Node<E>(val);
31            tailNode = tailNode.next;
32        }
33    }
34
35    public E pop() {
36
37        if ( headNode == null ) {
38            return null;
39        } else {
40
41            E val = headNode.val;
42            headNode = headNode.next;
43            if ( headNode == null ) {
44                tailNode = null;
45            }
46
47            return val;
48        }
49    }
50
51    public E peek() {
52        if ( headNode == null ) {
53            return null;
54        } else {
55            return headNode.val;
56        }
57    }
58
59
60    public static void main(String[] args) {
61
62        ImplementQueueWithLinkedList<Integer> queue
63            = new ImplementQueueWithLinkedList<>();
64
65        for (int i = 0; i < 10; i++) {
66            queue.push(i);
67        }
68
69        System.out.println(queue.peek());
70
71        for (int i = 0; i < 15; i++) {
72            System.out.println(queue.pop());
73        }
74    }
75 }
```

74 }

75 }