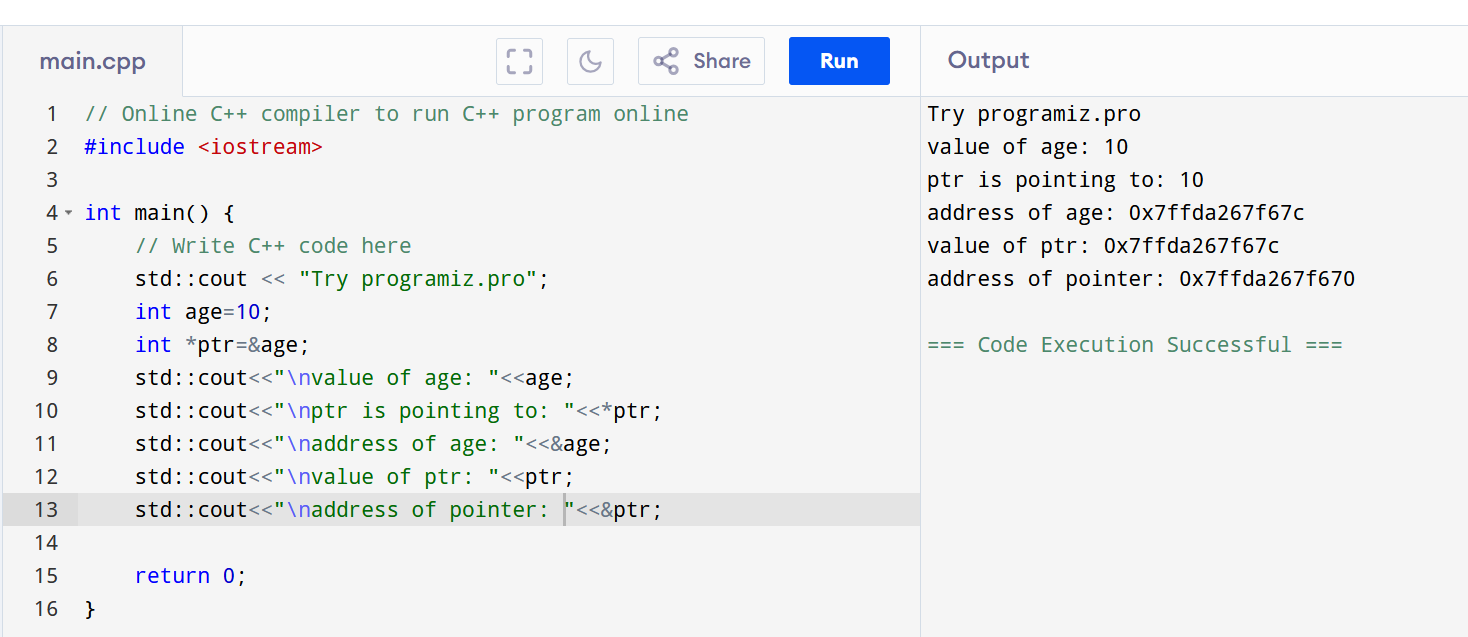
Day 13 - 104608492 - Shirisha Perapagu

Pointers & LinkedList

Task 01

Pointers in C++



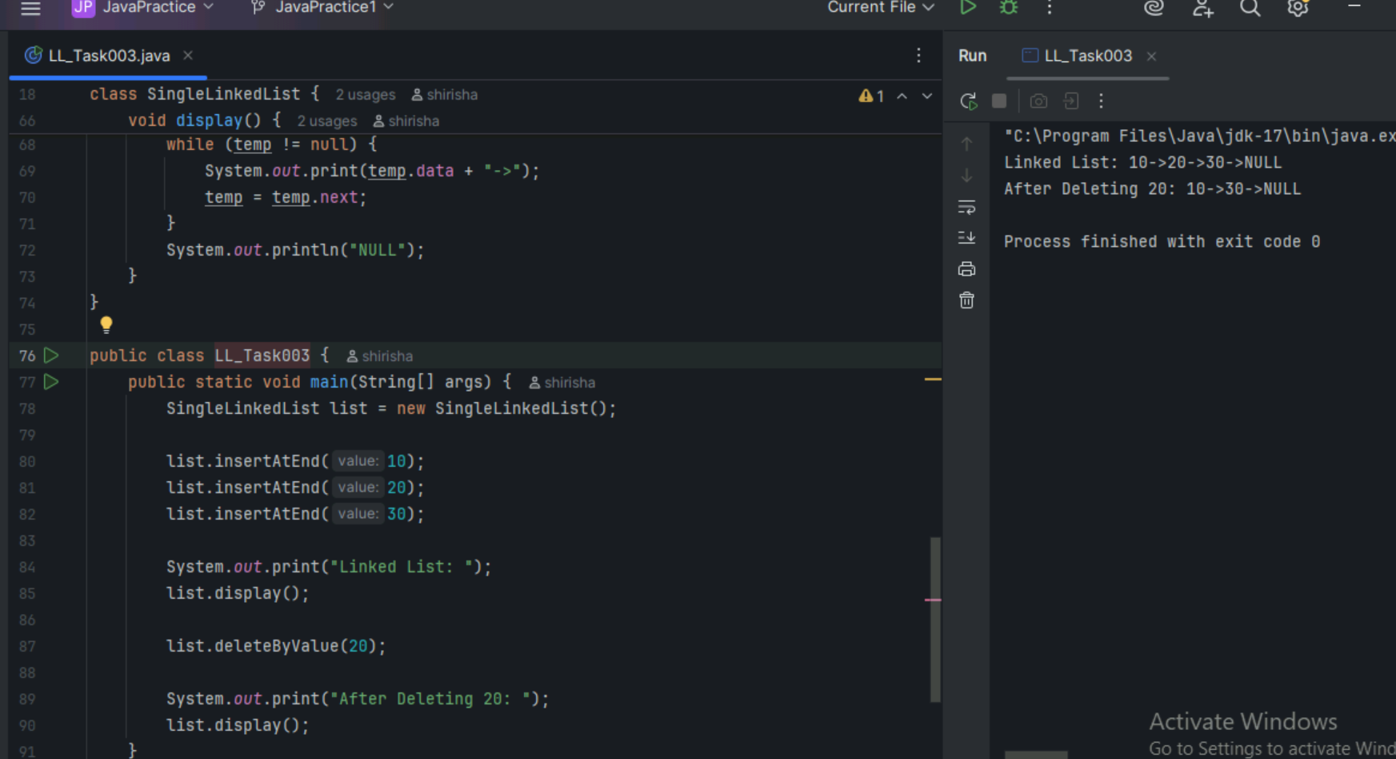
Task 02

Linked List example in C++



Task 03

Use the above code to create a Java code which creates a linked list.

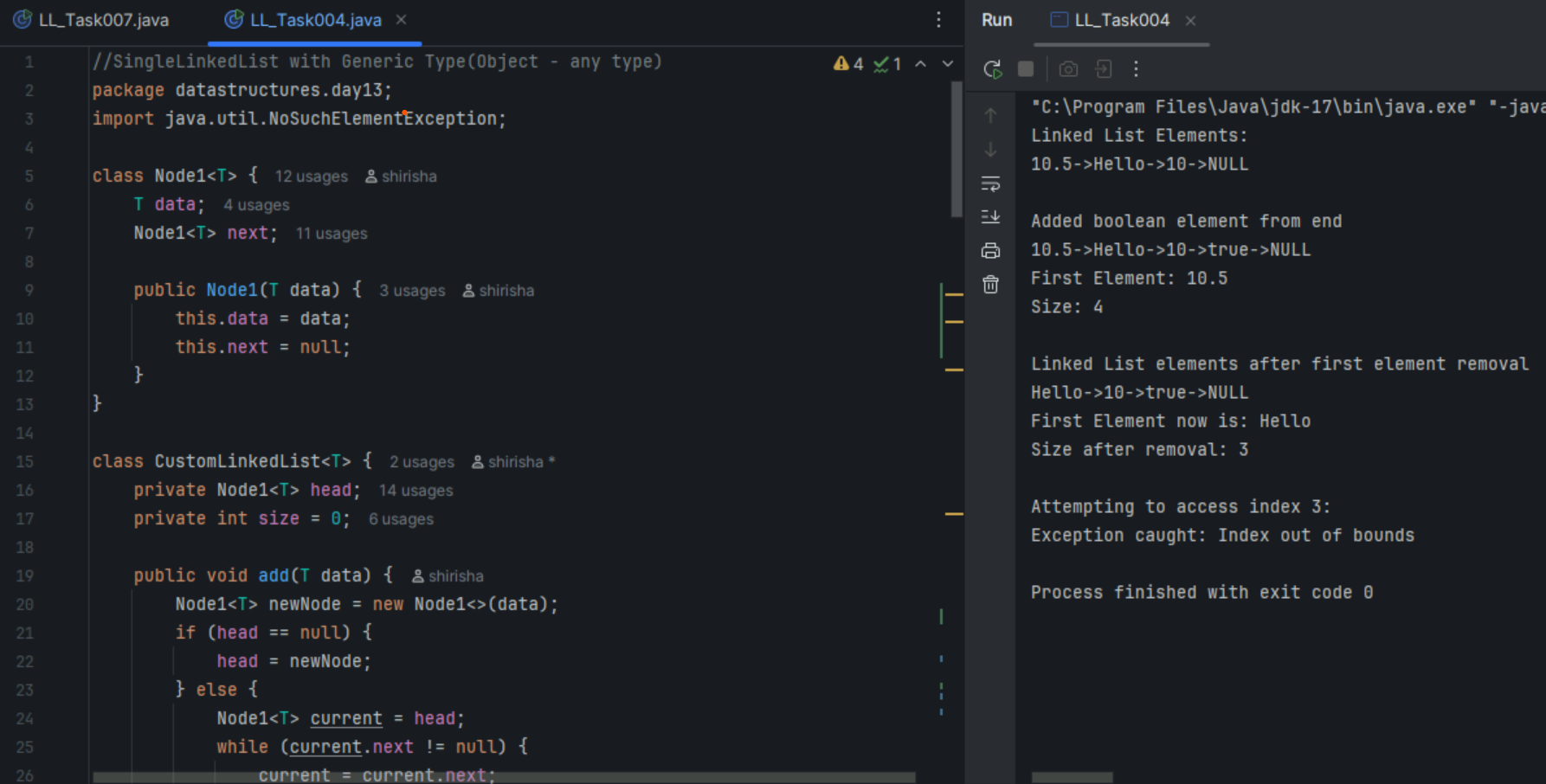


Task 04

Try to create a node and add a value to it.. Which can take any kind of data in the Node..

And perform operations like adding element from end, remove element, display elements,

Size of linked list and try to catch the index out of bounds exception.



Task 05

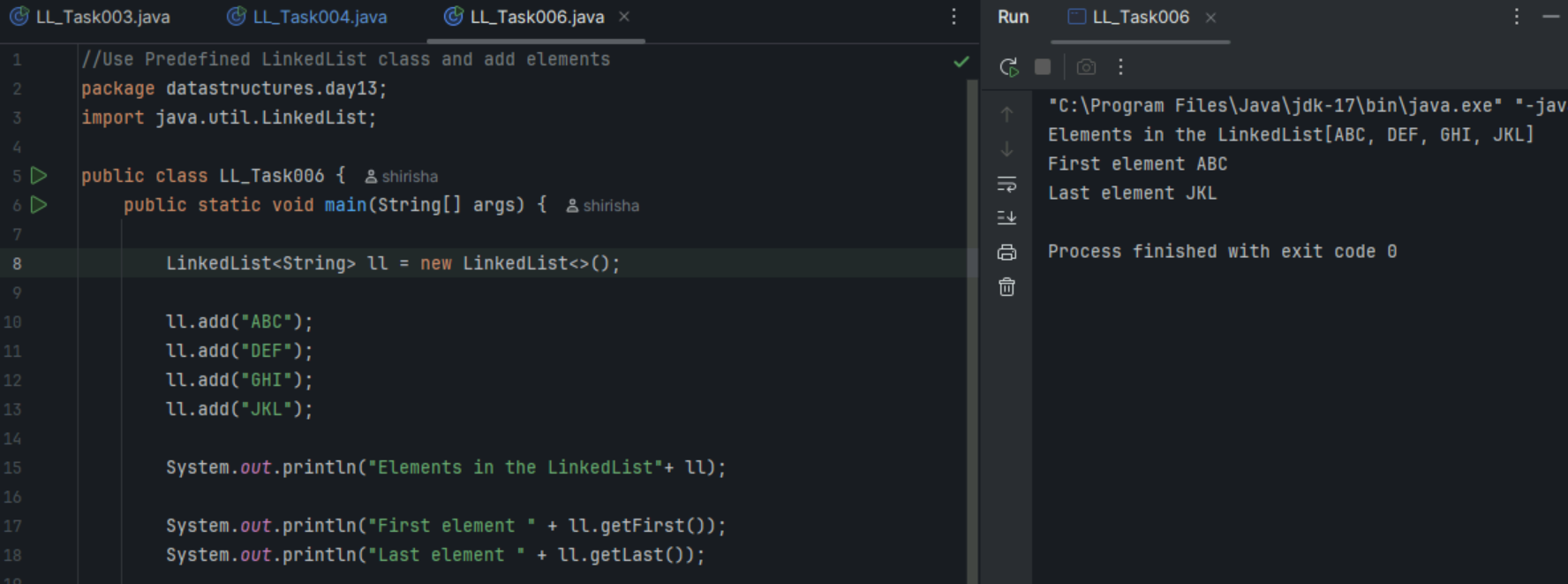
List down all the methods of Linked list

Method Description

1. add() adds an item to the list
2. addAll() adds a collection of items to the list
3. addFirst() adds an item in the beginning of the list
4. addLast() adds an item to the end of the list
5. clear() remove all items from the list
6. clone() create a copy of the LinkedList
7. contains() checks whether an item exists in the list
8. forEach() perform an action on every item in the list
9. get() return the item at a specific position in the list
10. getFirst() returns the first item in the list
11. getLast() returns the last item in the list
12. element() gets the first item in the list same ad getFirst()
13. indexOf() returns the position of the first occurrence of an item in the list
14. isEmpty() checks whether the list is empty
15. iterator() return an Iterator object for the LinkedList
16. lastIndexOf() return the position of the last occurrence of an item in the list
17. listIterator() return a ListIterator object for the LinkedList
18. offer() adds an item at the end of the list
19. offerFirst() adds an item at the beginning of the list
20. offerLast() adds an item at the end of the list
21. peek() retrieves the first item in the list same as getFirst()
22. peekFirst() retrieves the first item in the list same as peek()
23. peekLast() retrieves the last item in the list
24. poll() retrieves and removes the first item in the list.
25. pollFirst() retrieves and removes the first item in the list same as poll()
26. pollLast() retrieves and removes the last item in the list.
27. pop() returns the first element in the list same as removeFirst()
28. push() adds an item to the beginning of the list same as addFirst()
29. remove() remove an item from the list
30. removeAll() remove a collection of items from the list
31. removeFirst() removes the first item in the list
32. removeFirstOccurrence() removes the first occurrence of a specified item in the list
33. removeIf() remove all items from the list which meet a specified condition
34. removeLast() removes the last item in the list
35. removeLastOccurrence() removes the last occurrence of a specified item in the list
36. replaceAll() replace each item in the list with the result of an operation on that item
37. retainAll() remove all elements from the list which do not belong to a specified collection
38. set() replace an item at a specified position in the list
39. size() return the number of items in the list
40. sort() Sort the list
41. spliterator() return a Spliterator object for the LinkedList
42. toArray() return an array containing the list's items Object[]

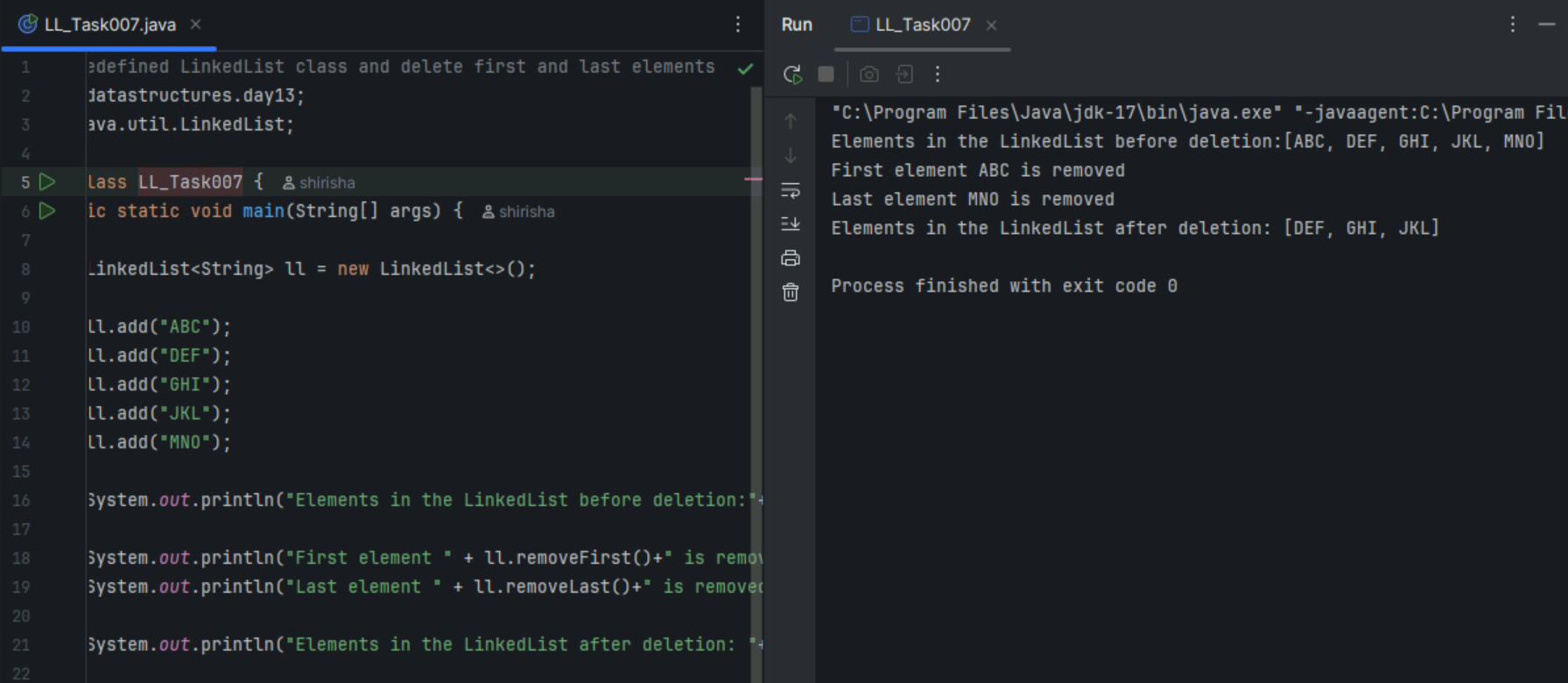
Task 06

Create linked list using Pre defined class and add elements to it.



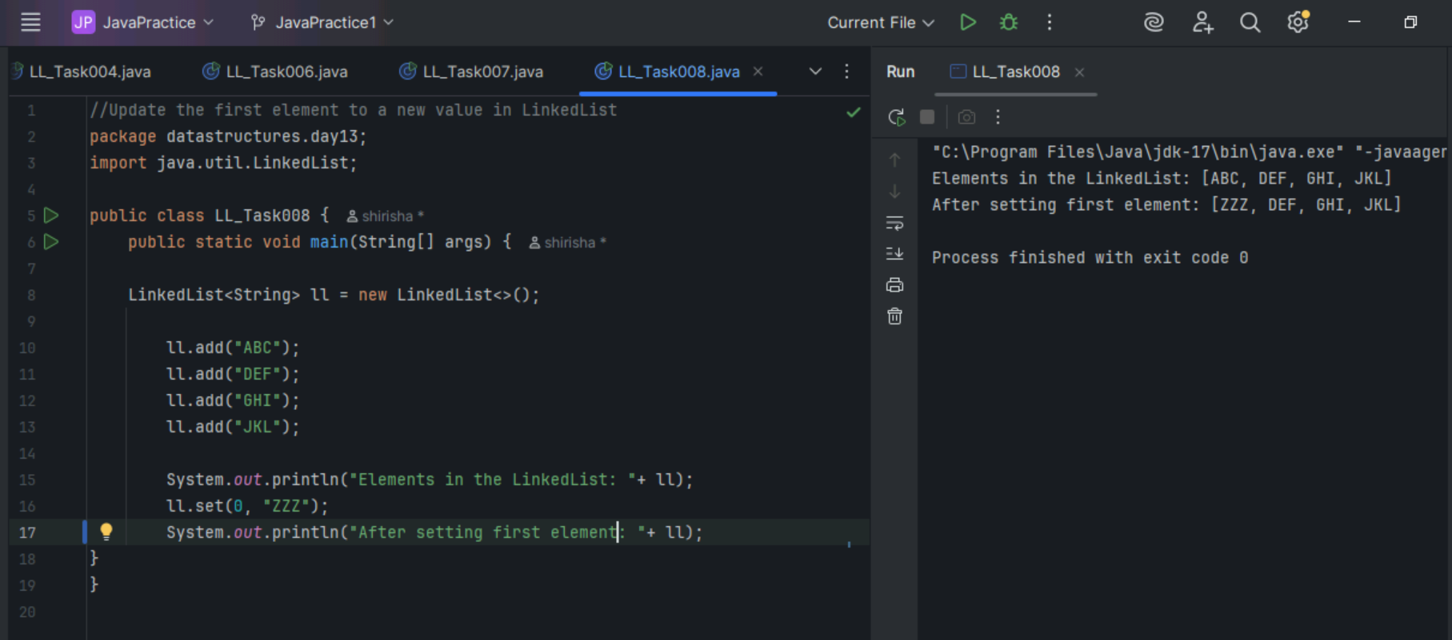
Task 07

Remove first and remove last element and display all elements in the linked list



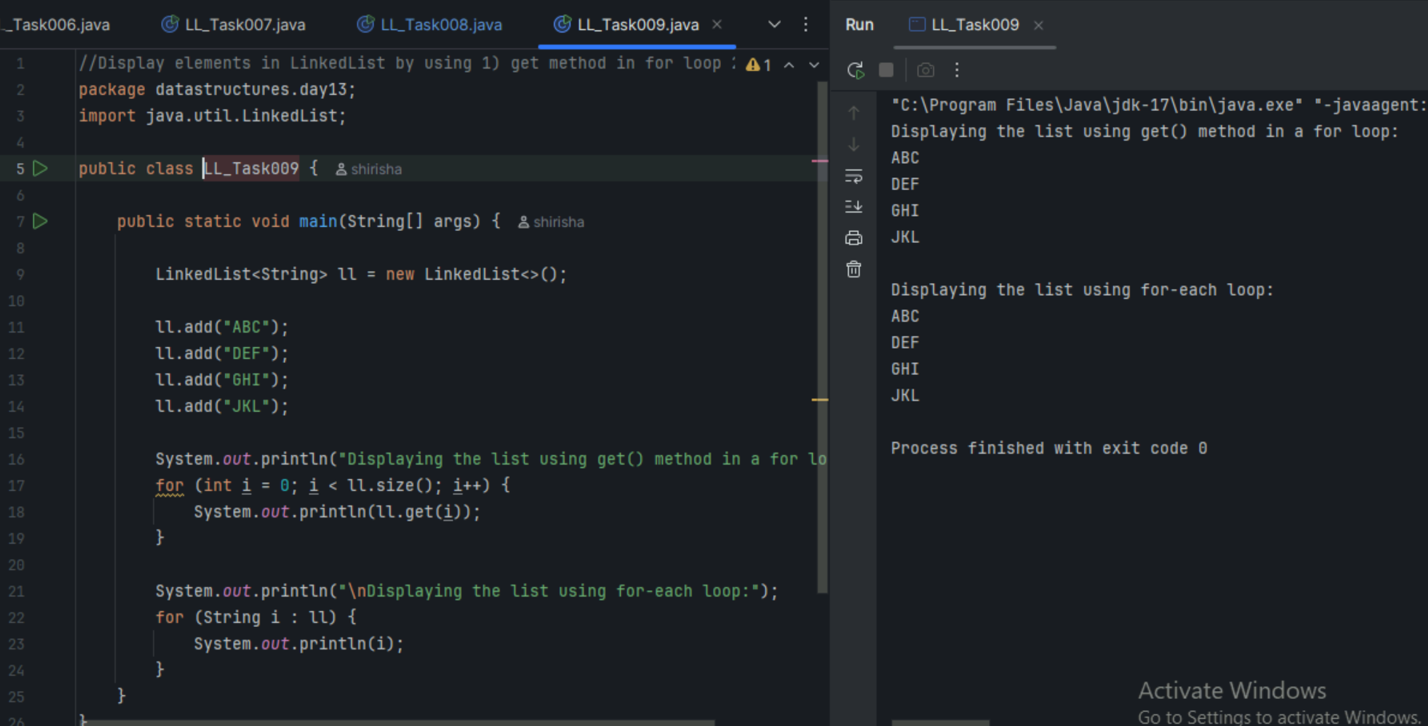
Task 08

In the list update the 1st element to a new value.



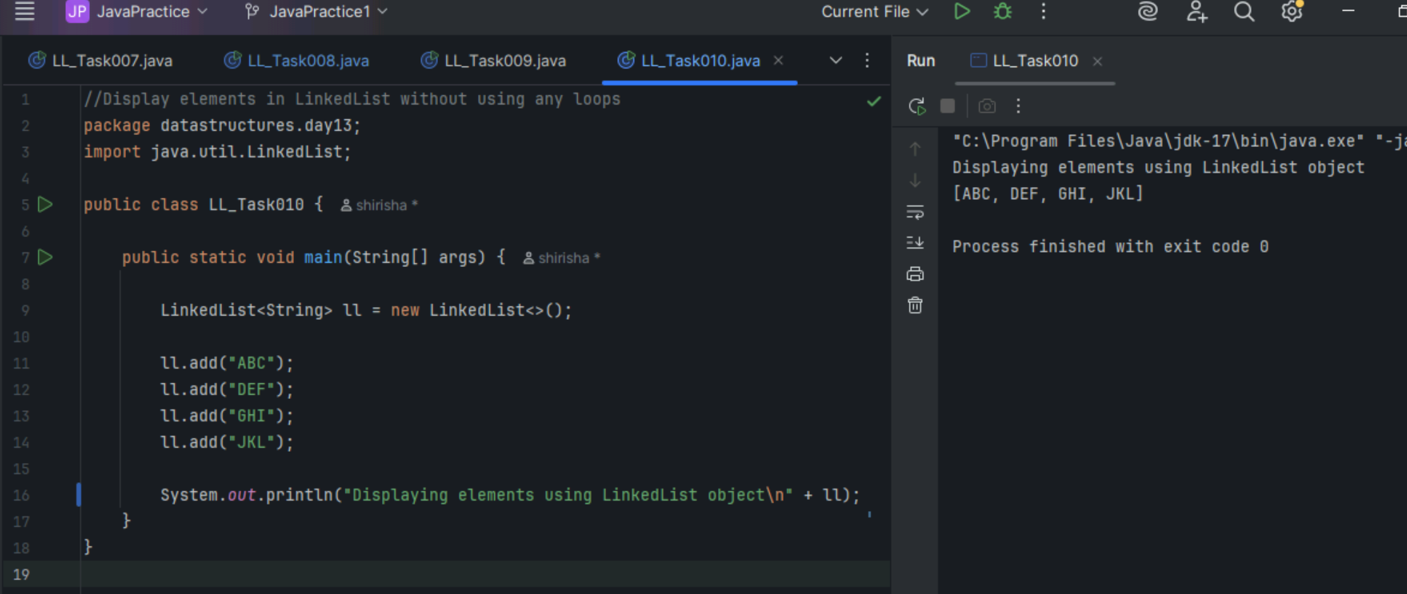
Task 09

Display the list twice 1..... with get method in for loop and 2 ... for each loop



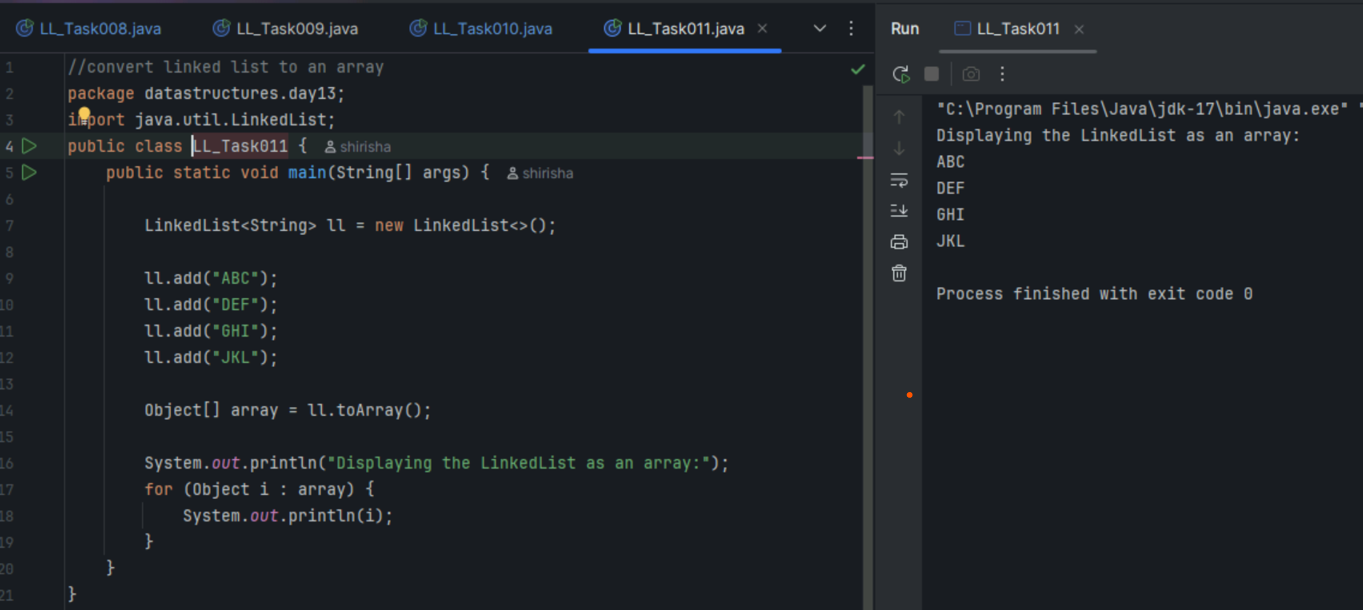
Task 10

Display the elements of the linked list without loops



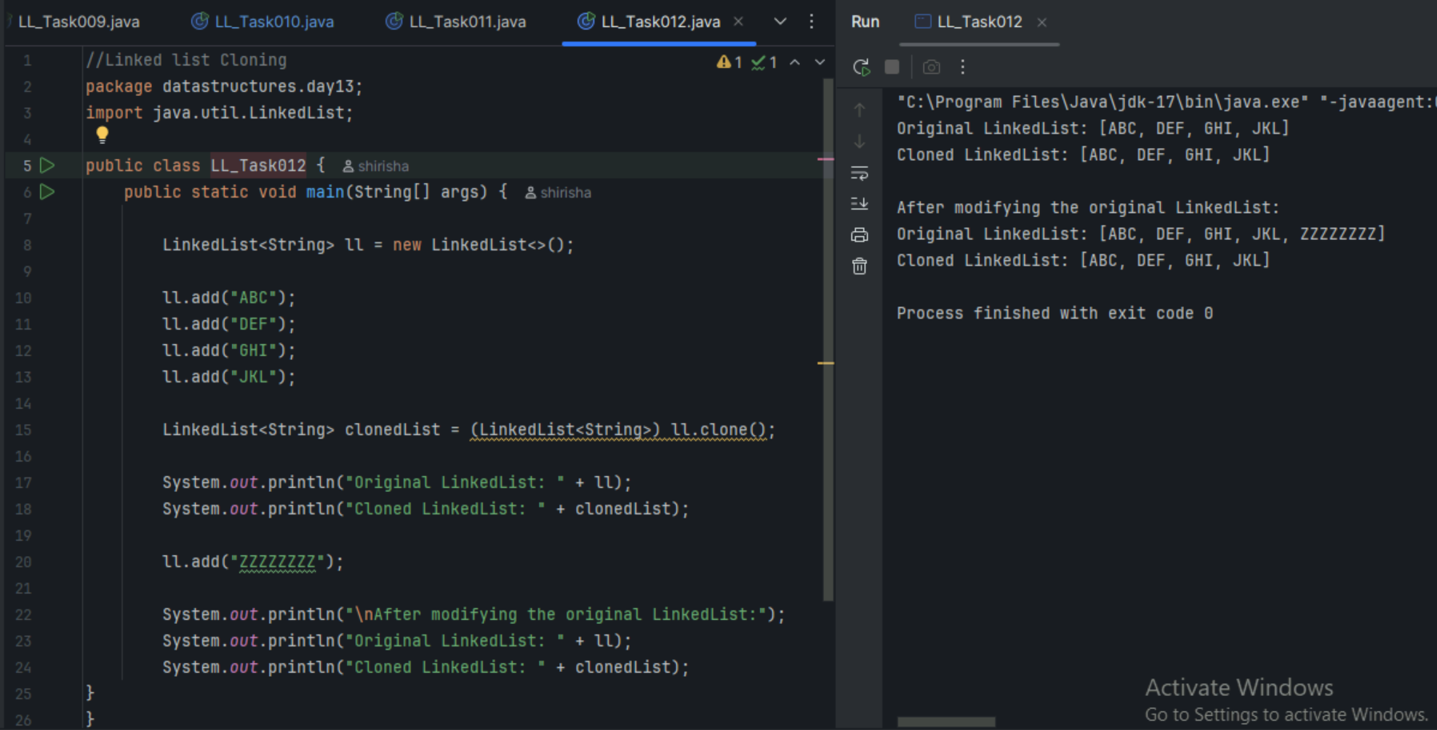
Task 11

Convert the linked list to an array and display.



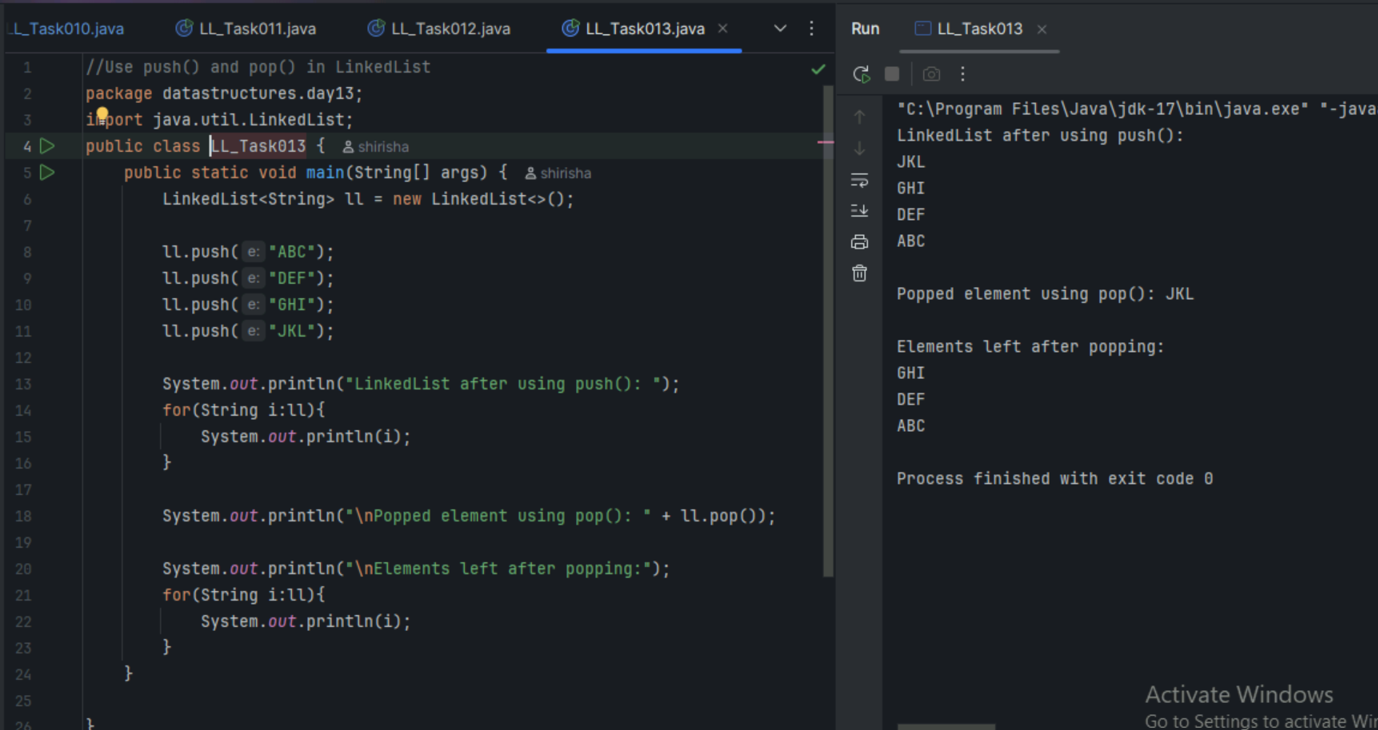
Task 12

Clone the linked list to check if its getting cloned.



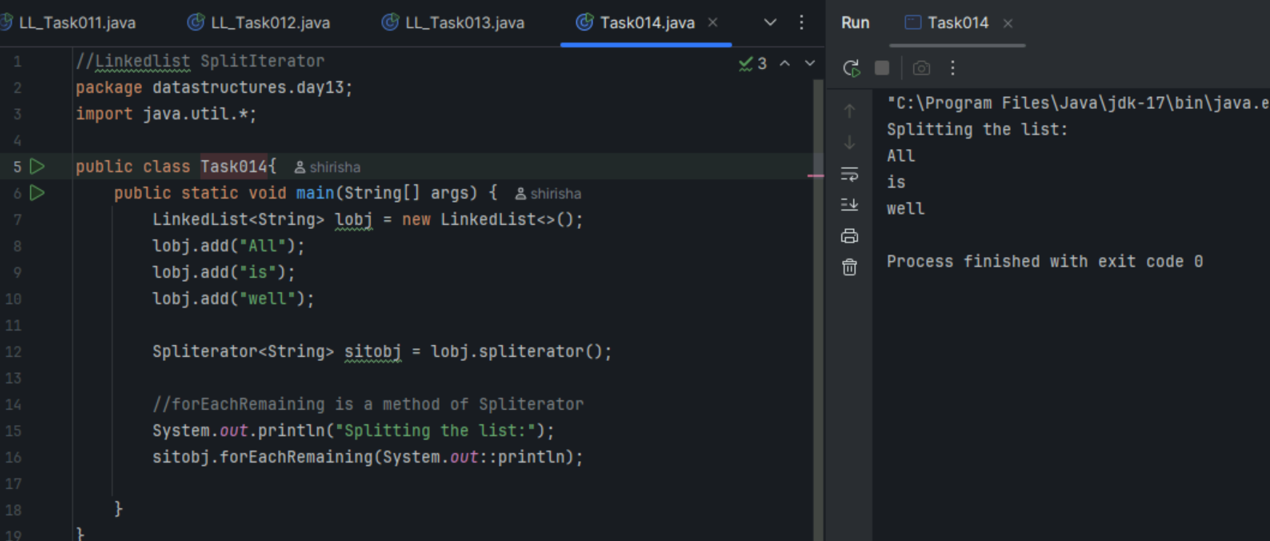
Task 13

Use pop and push methods on linked list..



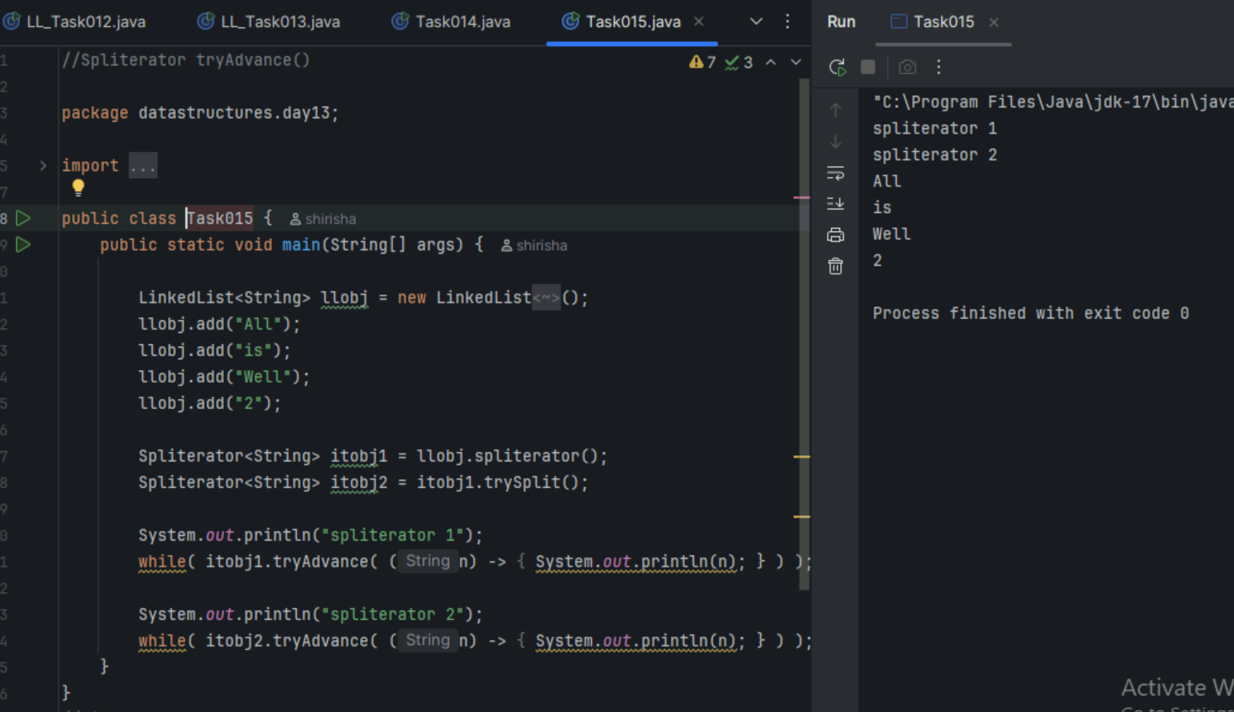
Task 14

Splititerator



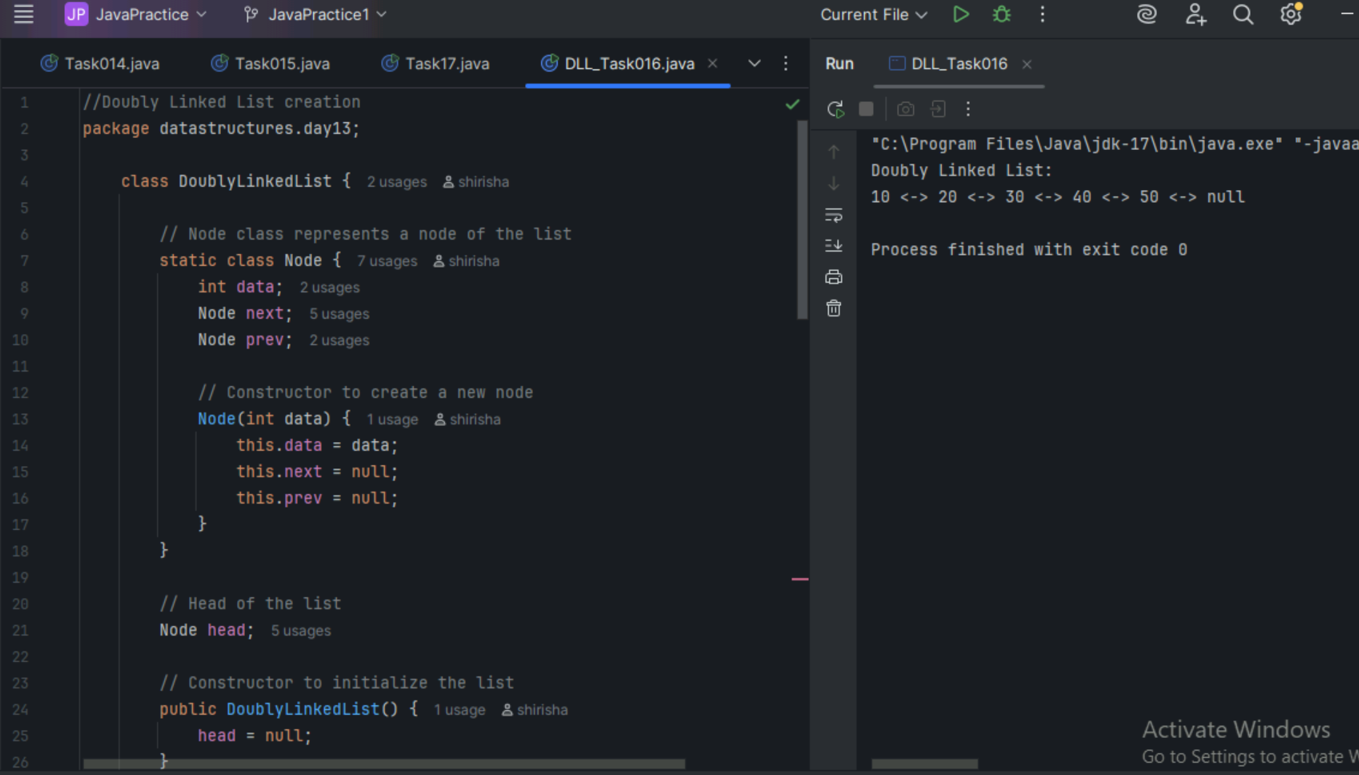
Task 15

Splititerator tryAdvance()



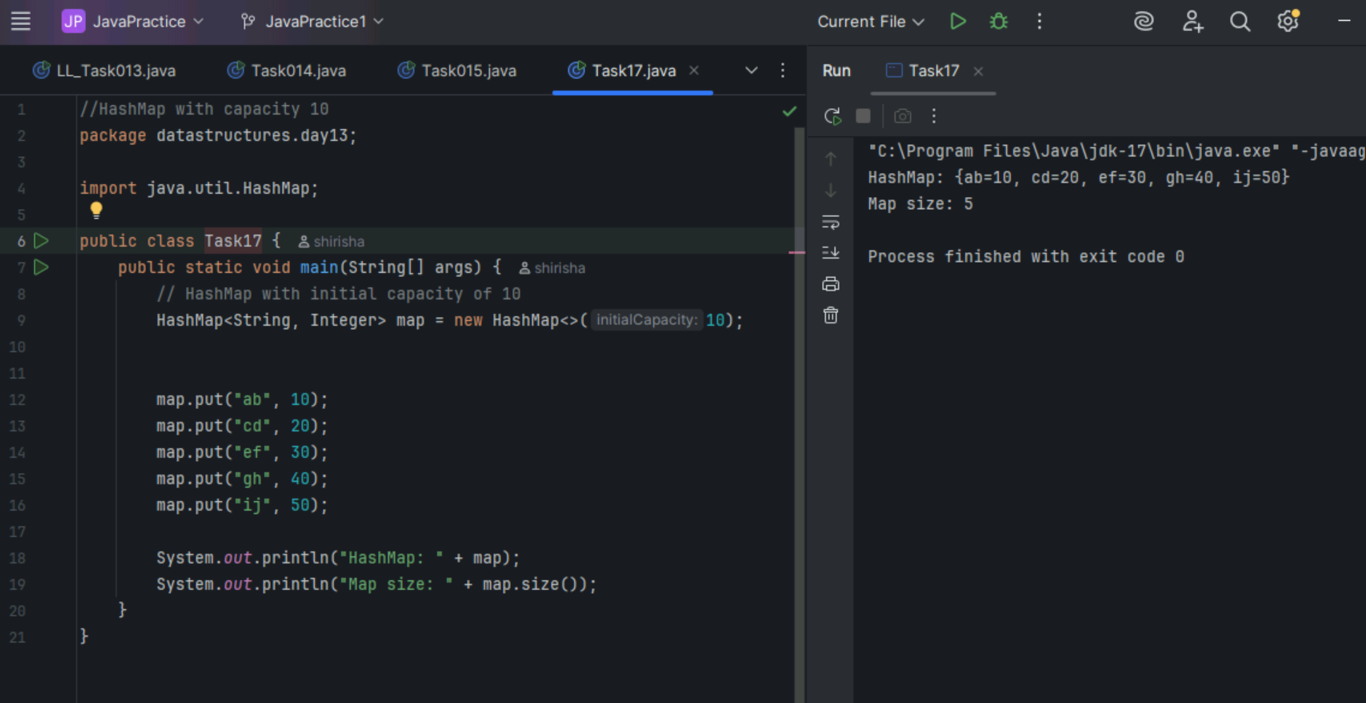
Task 16

Create a doubly linked list..



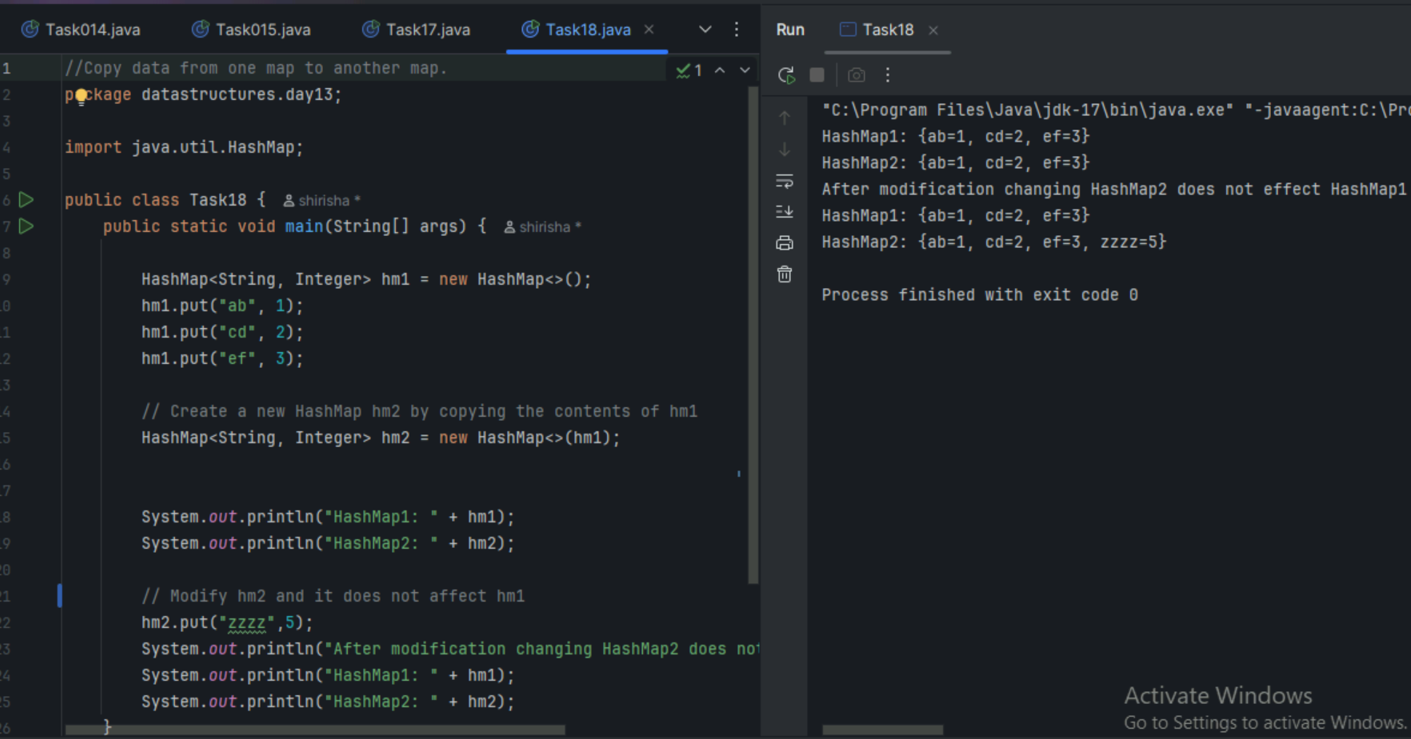
Task 17

Create a Hash MAp of capacity 10.



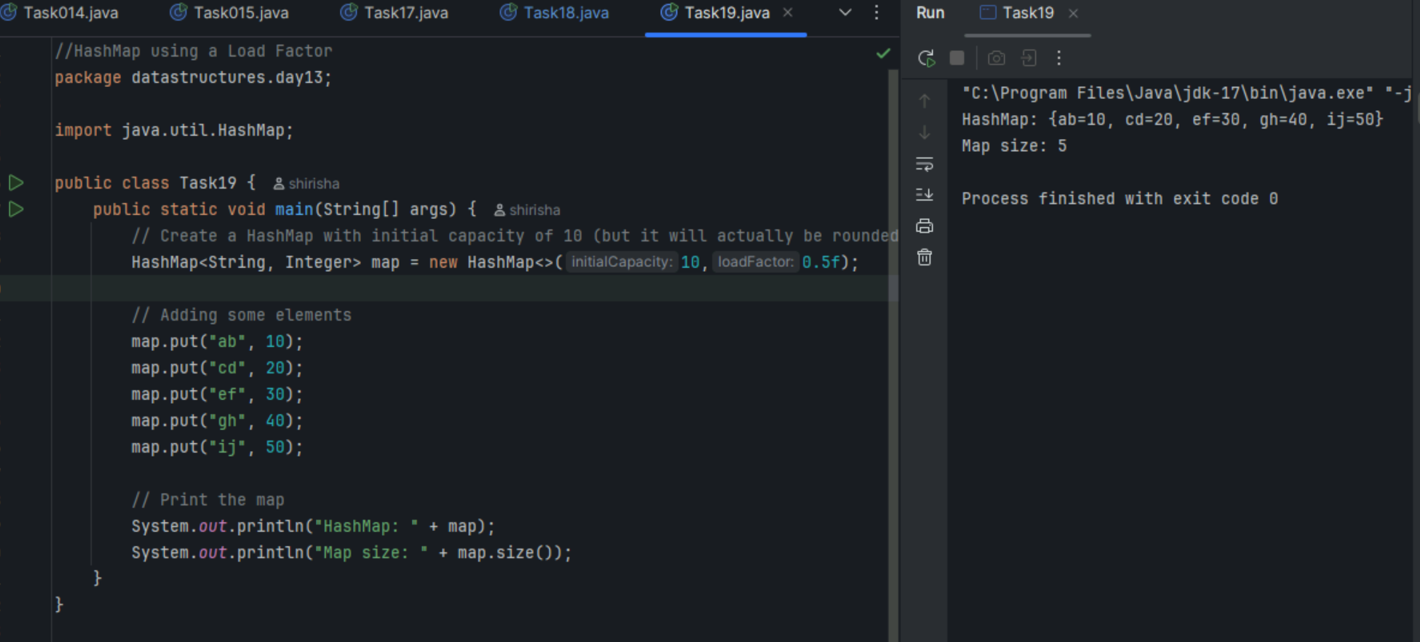
Task 18

Copy data from one map to another map.



Task 19

Create a HashMap using a load factor



Task 20

Use custom method of Creating  a circular linked list and traverse the elements (display)

