GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 3 Report

Melihcan Çilek

1801042092

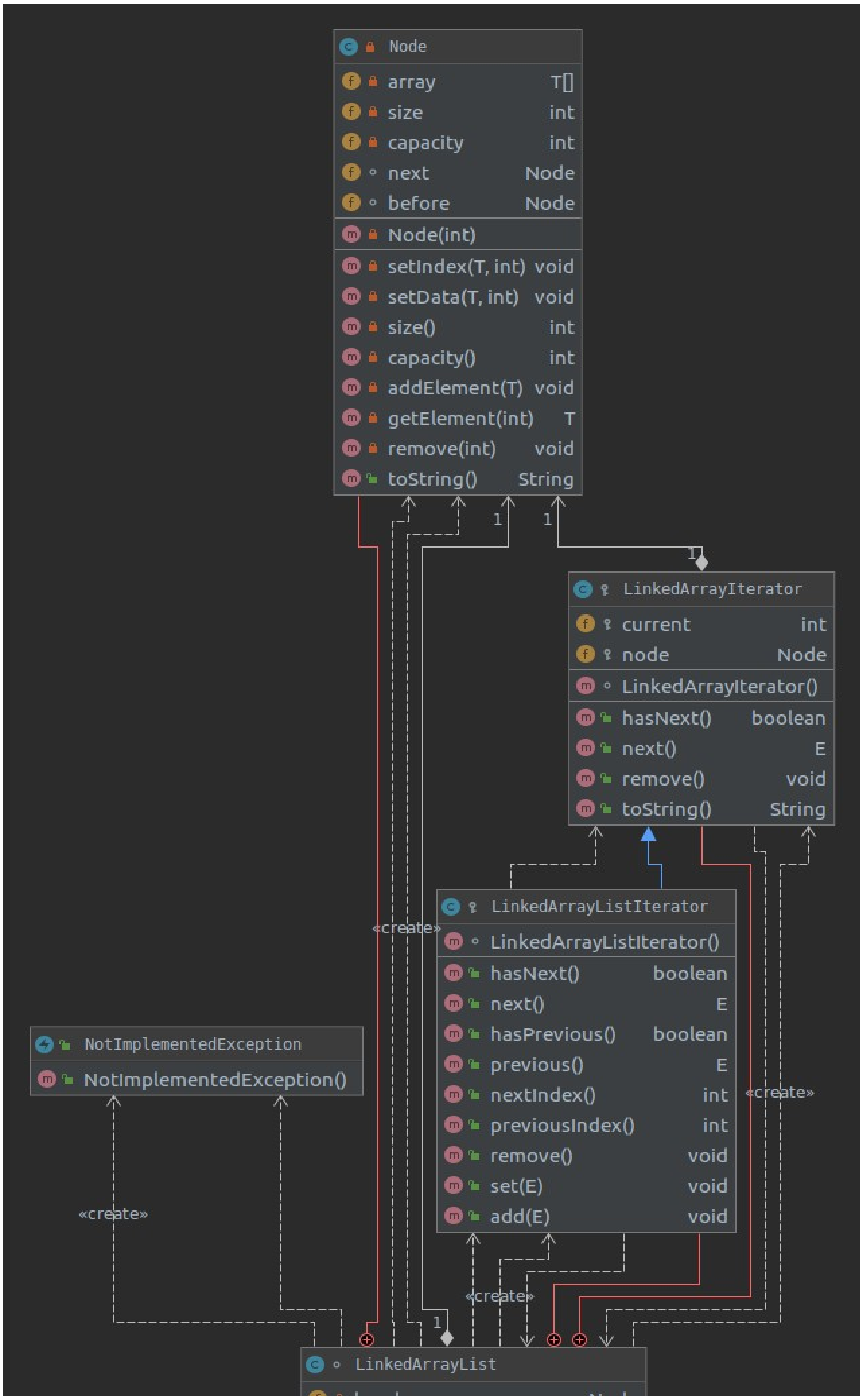
QUESTION 1

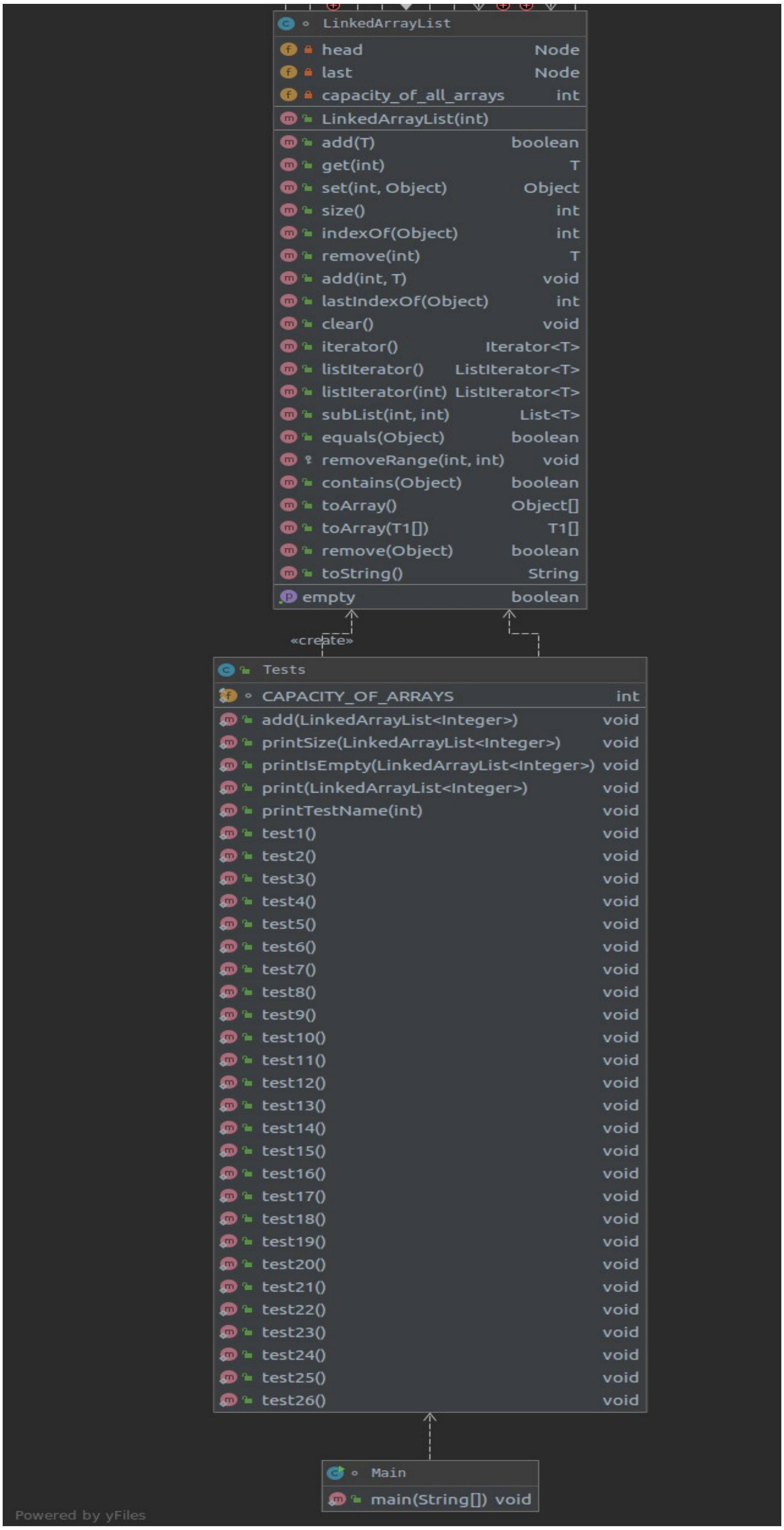
Problem solution approach:

In this problem, what wanted from us is that implementing linked list with nodes are constant elemented array. First, we have to create custom created linked list and private node class that holds array which is constant size and pre-dedicated from at the beginning of the Main method. Second, we have to implement methods that are said at pdf which are add and remove main methods that have to be tested. For implementing these methods, I tried to use arraylist and linkedlist from java.util Collection hierarchy. After that, I try to make implementation according to that classes.

Class Diagrams:

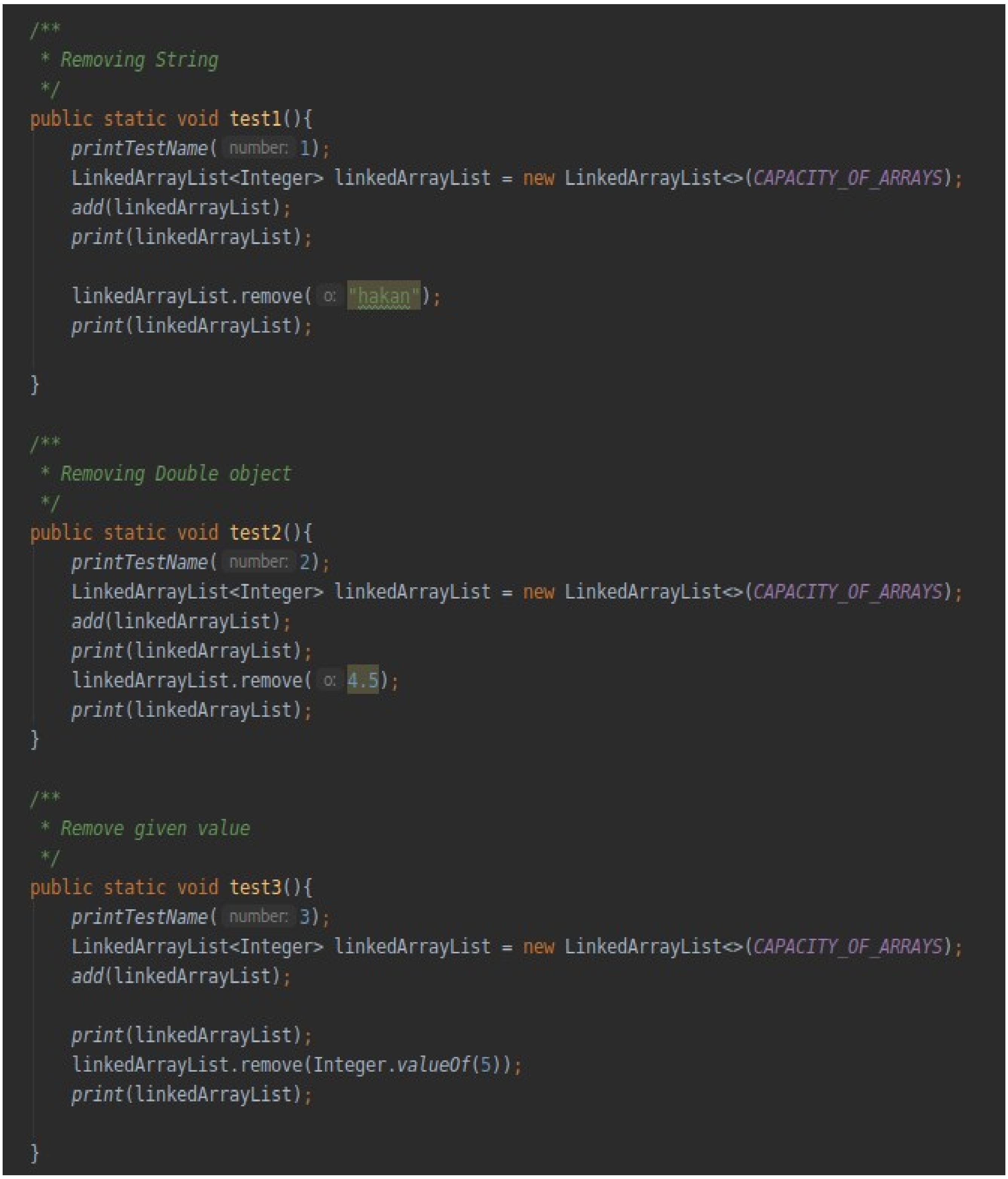
In this problem, we need to implement add and remove methods in our own custom style. Because of that our LinkedArrayList will extend AbstractList abstract class and implement List interface with generic type. Inside abstract AbstractList abstract class, we have to implement ListIterator and Iterator interfaces as well. Some of the methods don’t have to be implemented so we have to throw an exception that throw NotImplemented exception. At next page, class diagrams are added.


Node
array T[]
size int
capacity int
next Node
before Node
Node(int)
setIndex(T, int) void
setData(T, int) void
size() int
capacity() int
addElement(T) void
getElement(int) T
remove(int) void
toString() String
1
1
1
LinkedArrayIterator
current int
node Node
LinkedArrayIterator()
hasNext() boolean
next() E
remove() void
toString() String
LinkedArrayListIterator
LinkedArrayListIterator()
hasNext() boolean
next() E
hasPrevious() boolean
previous() E
nextIndex() int
previousIndex() int
remove() void
set(E) void
add(E) void
«create»
«create»
NotImplementedException
NotImplementedException()
«create»
«create»
1
LinkedArrayList

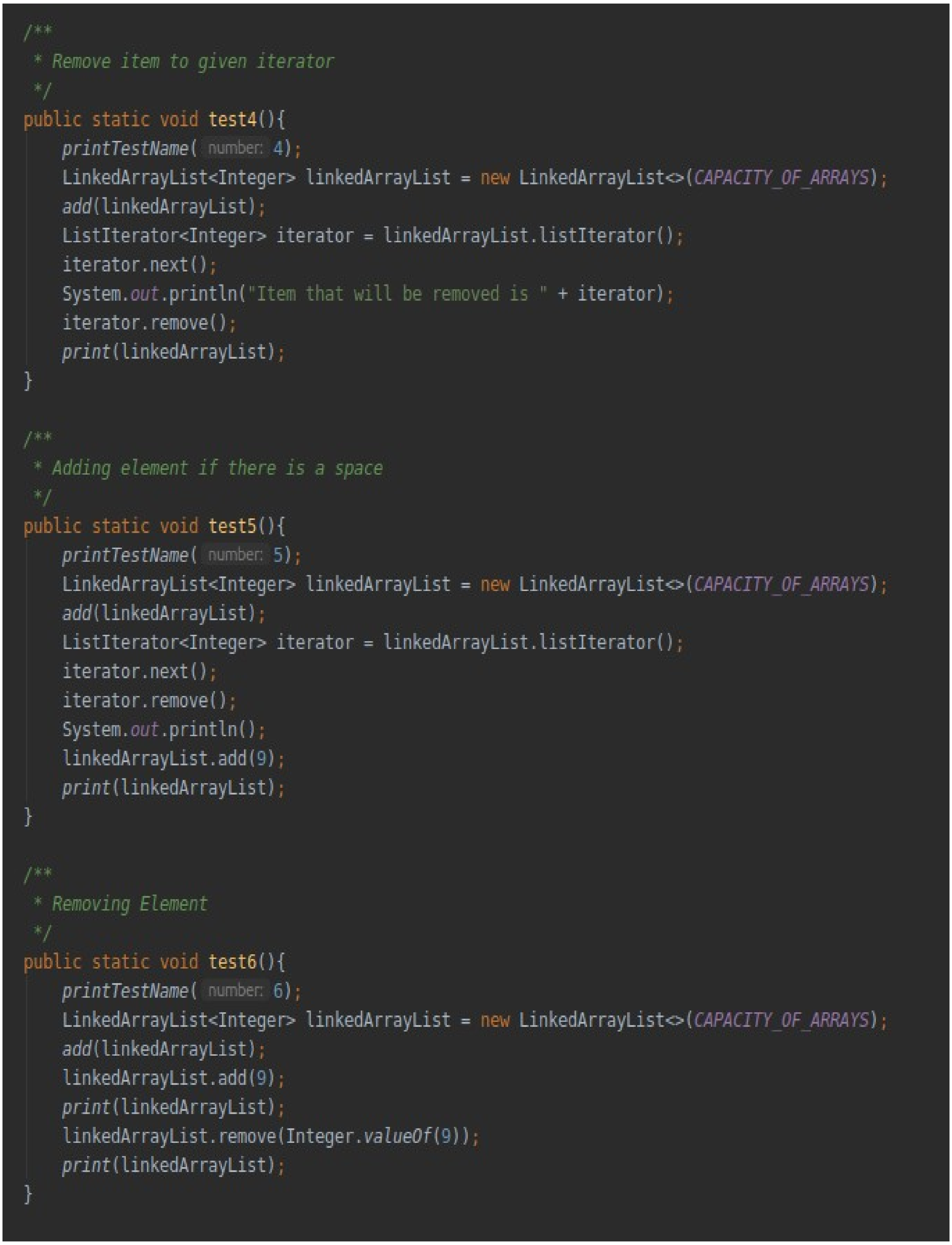

LinkedArrayList
head Node
last Node
capacity_of_all_arrays int
LinkedArrayList(int)
add(T) boolean
get(int) T
set(int, Object) Object
size() int
indexOf(Object) int
remove(int) T
add(int, T) void
lastIndexOf(Object) int
clear() void
iterator() Iterator<T>
listIterator() ListIterator<T>
listIterator(int) ListIterator<T>
subList(int, int) List<T>
equals(Object) boolean
removeRange(int, int) void
contains(Object) boolean
toArray() Object[]
toArray(T1[]) T1[]
remove(Object) boolean
toString() String
empty boolean
«create»
Tests
CAPACITY_OF_ARRAYS int
add(LinkedArrayList<Integer>) void
printSize(LinkedArrayList<Integer>) void
printIsEmpty(LinkedArrayList<Integer>) void
print(LinkedArrayList<Integer>) void
printTestName(int) void
test1() void
test2() void
test3() void
test4() void
test5() void
test6() void
test7() void
test8() void
test9() void
test10() void
test11() void
test12() void
test13() void
test14() void
test15() void
test16() void
test17() void
test18() void
test19() void
test20() void
test21() void
test22() void
test23() void
test24() void
test25() void
test26() void
Main
main(String[]) void
Powered by yFiles

Test Cases:

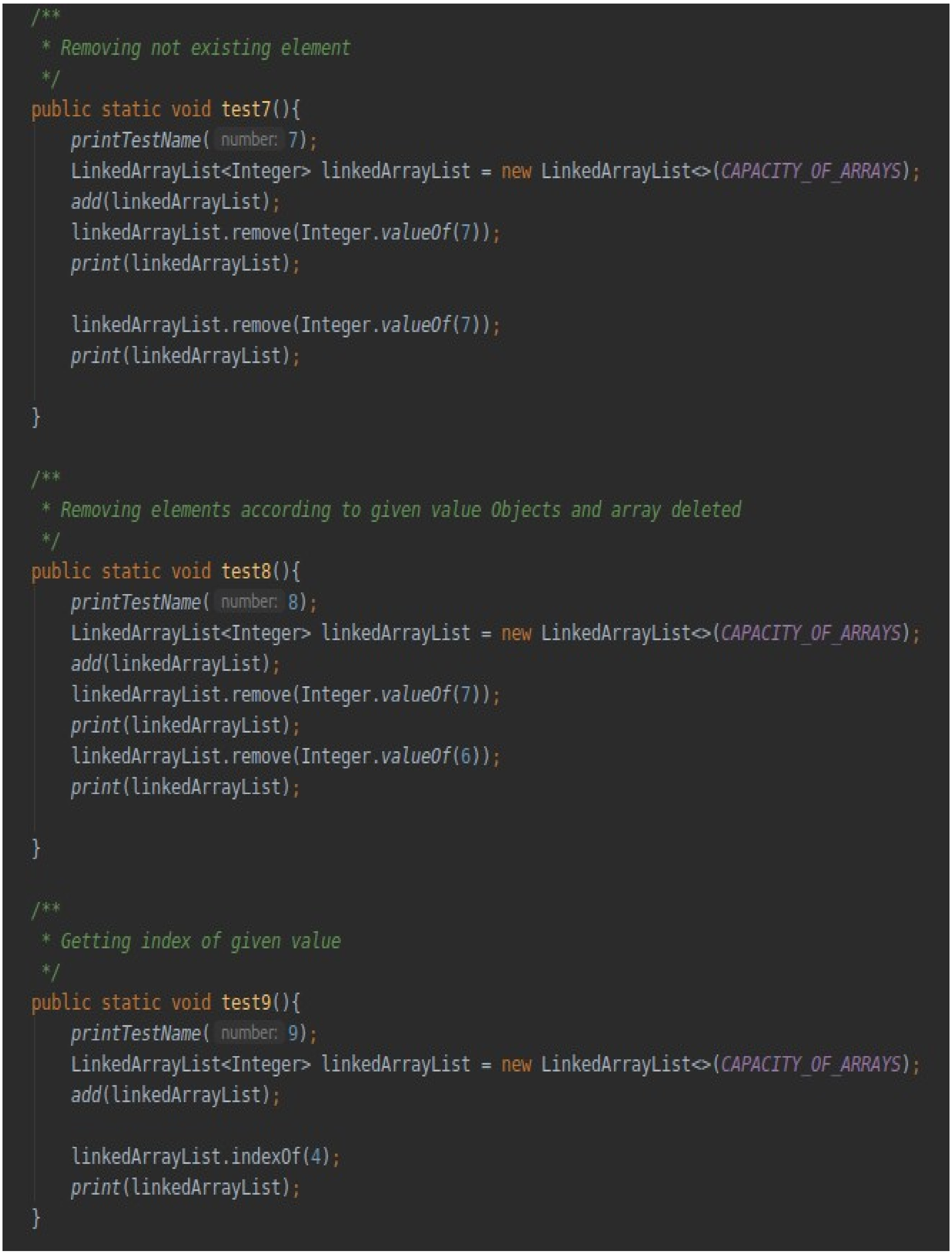
Tests:

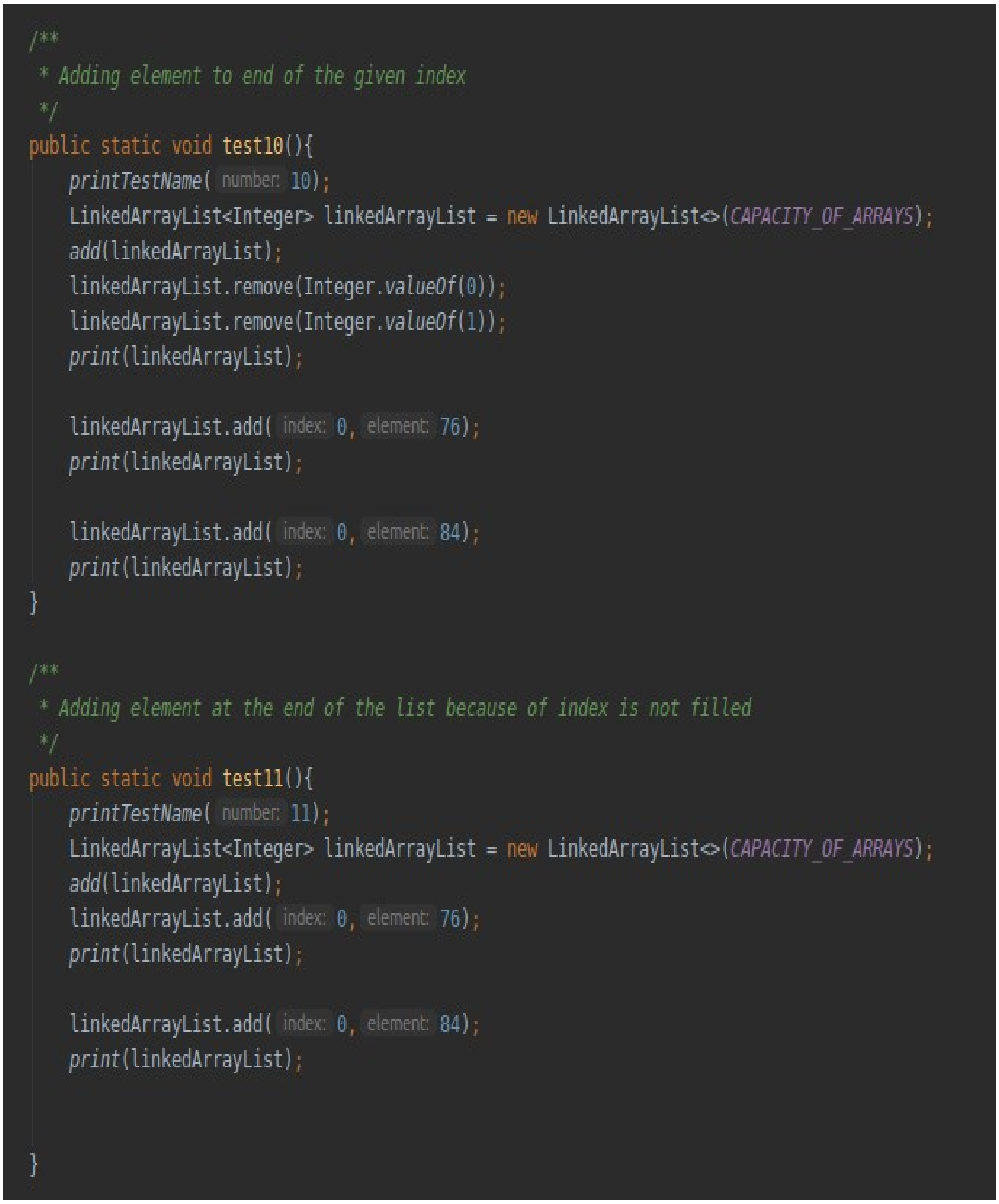


```java
/**
 * Removing String
 */
public static void test1(){
    printTestName( number: 1);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    print(linkedArrayList);

    linkedArrayList.remove( o: "hakan");
    print(linkedArrayList);

}

/**
 * Removing Double object
 */
public static void test2(){
    printTestName( number: 2);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    print(linkedArrayList);
    linkedArrayList.remove( o: 4.5);
    print(linkedArrayList);
}

/**
 * Remove given value
 */
public static void test3(){
    printTestName( number: 3);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);

    print(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(5));
    print(linkedArrayList);

}
```

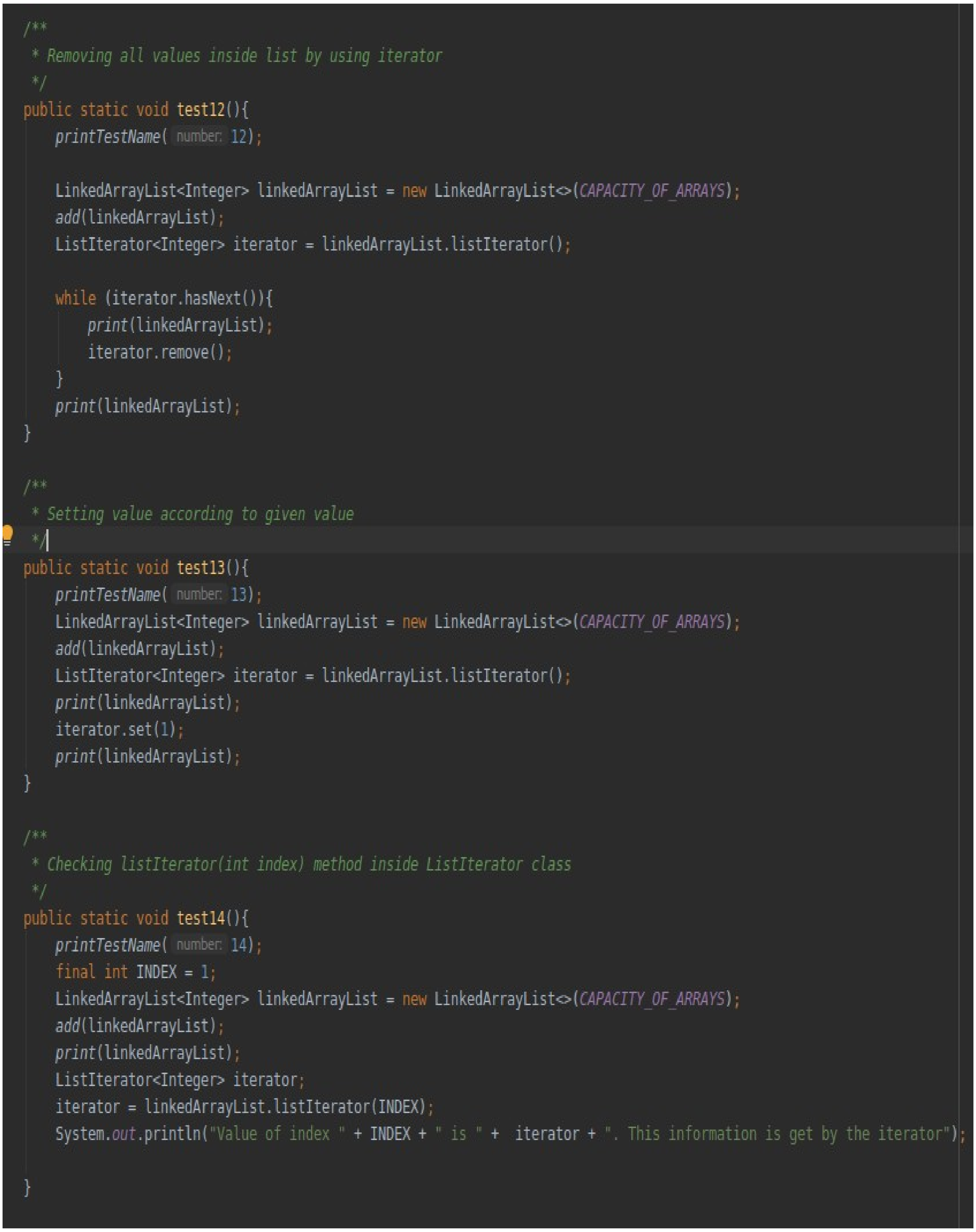


```java
/**
 * Remove item to given iterator
 */
public static void test4(){
    printTestName( number: 4);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    ListIterator<Integer> iterator = linkedArrayList.listIterator();
    iterator.next();
    System.out.println("Item that will be removed is " + iterator);
    iterator.remove();
    print(linkedArrayList);
}

/**
 * Adding element if there is a space
 */
public static void test5(){
    printTestName( number: 5);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    ListIterator<Integer> iterator = linkedArrayList.listIterator();
    iterator.next();
    iterator.remove();
    System.out.println();
    linkedArrayList.add(9);
    print(linkedArrayList);
}

/**
 * Removing Element
 */
public static void test6(){
    printTestName( number: 6);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    linkedArrayList.add(9);
    print(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(9));
    print(linkedArrayList);
}
```

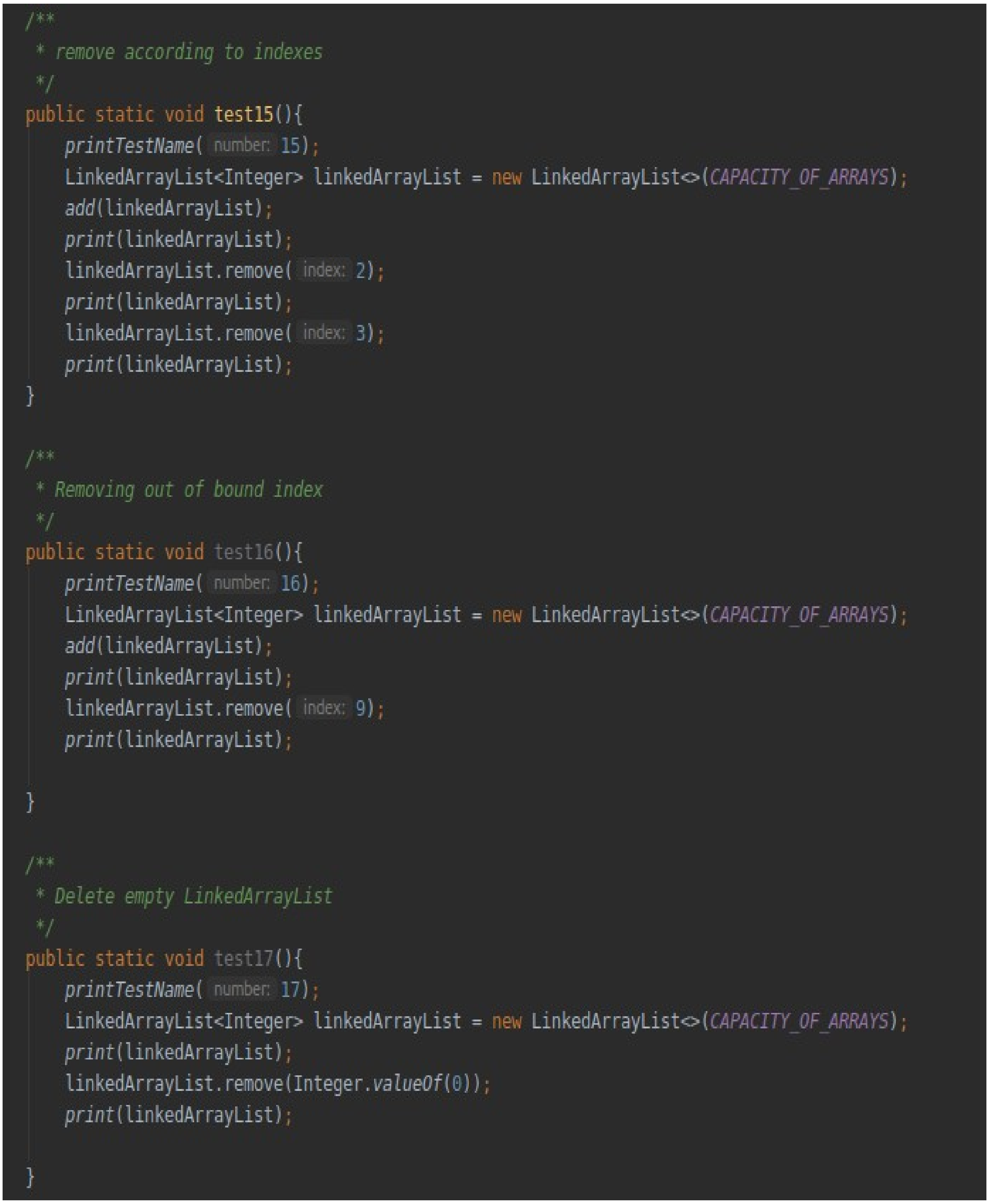

```java
/**
 * Removing not existing element
 */
public static void test7(){
    printTestName( number: 7);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(7));
    print(linkedArrayList);

    linkedArrayList.remove(Integer.valueOf(7));
    print(linkedArrayList);

}

/**
 * Removing elements according to given value Objects and array deleted
 */
public static void test8(){
    printTestName( number: 8);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(7));
    print(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(6));
    print(linkedArrayList);

}

/**
 * Getting index of given value
 */
public static void test9(){
    printTestName( number: 9);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);

    linkedArrayList.indexOf(4);
    print(linkedArrayList);
}
```



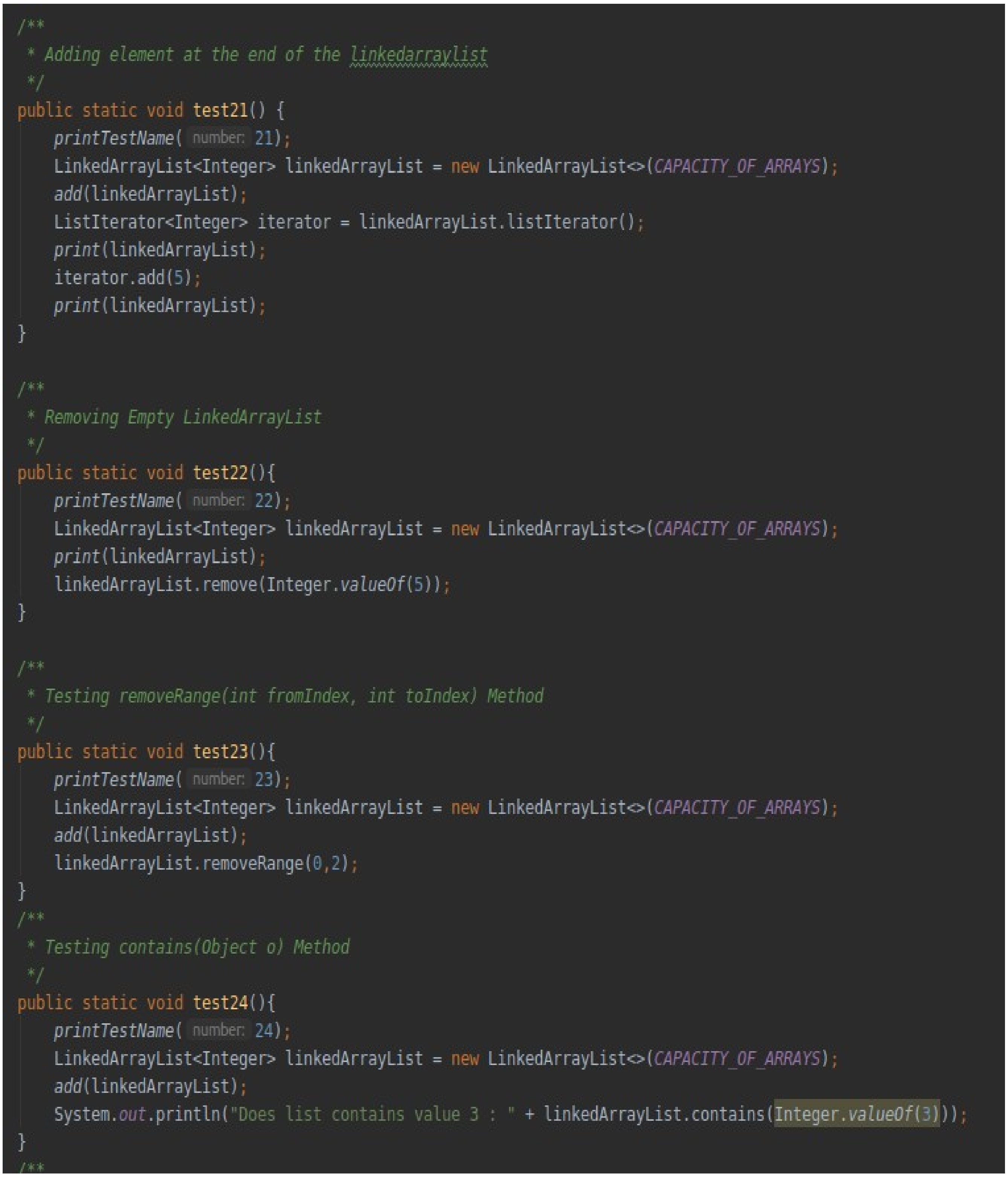
```java
/**
 * Adding element to end of the given index
 */
public static void test10(){
    printTestName( number: 10);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(0));
    linkedArrayList.remove(Integer.valueOf(1));
    print(linkedArrayList);

    linkedArrayList.add( index: 0, element: 76);
    print(linkedArrayList);

    linkedArrayList.add( index: 0, element: 84);
    print(linkedArrayList);
}

/**
 * Adding element at the end of the list because of index is not filled
 */
public static void test11(){
    printTestName( number: 11);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    linkedArrayList.add( index: 0, element: 76);
    print(linkedArrayList);

    linkedArrayList.add( index: 0, element: 84);
    print(linkedArrayList);


}
```



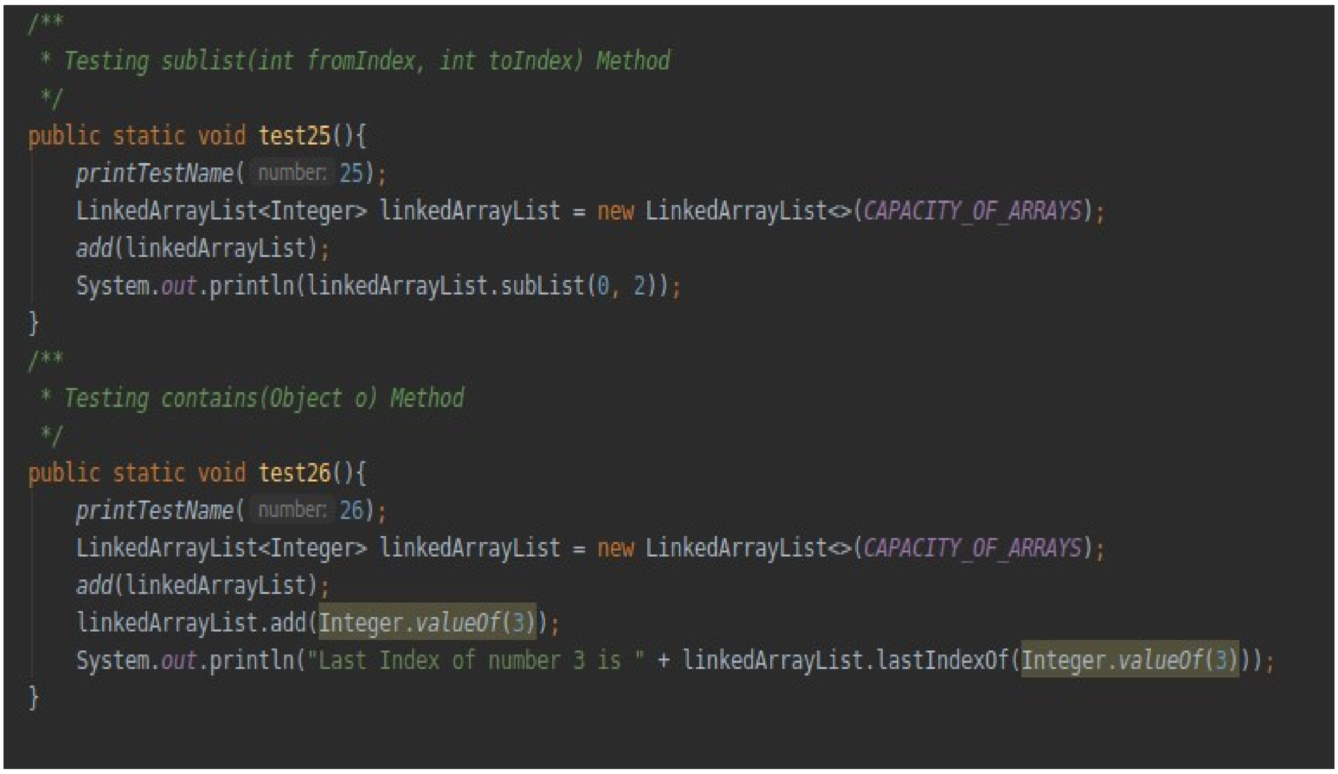
```java
/**
 * Removing all values inside list by using iterator
 */
public static void test12(){
    printTestName( number: 12);

    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    ListIterator<Integer> iterator = linkedArrayList.listIterator();

    while (iterator.hasNext()){
        print(linkedArrayList);
        iterator.remove();
    }
    print(linkedArrayList);
}

/**
 * Setting value according to given value
 */
public static void test13(){
    printTestName( number: 13);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    ListIterator<Integer> iterator = linkedArrayList.listIterator();
    print(linkedArrayList);
    iterator.set(1);
    print(linkedArrayList);
}

/**
 * Checking listIterator(int index) method inside ListIterator class
 */
public static void test14(){
    printTestName( number: 14);
    final int INDEX = 1;
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    print(linkedArrayList);
    ListIterator<Integer> iterator;
    iterator = linkedArrayList.listIterator(INDEX);
    System.out.println("Value of index " + INDEX + " is " +  iterator + ". This information is get by the iterator");

}
```



```java
/**
 * remove according to indexes
 */
public static void test15(){
    printTestName( number: 15);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    print(linkedArrayList);
    linkedArrayList.remove( index: 2);
    print(linkedArrayList);
    linkedArrayList.remove( index: 3);
    print(linkedArrayList);
}

/**
 * Removing out of bound index
 */
public static void test16(){
    printTestName( number: 16);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    print(linkedArrayList);
    linkedArrayList.remove( index: 9);
    print(linkedArrayList);

}

/**
 * Delete empty LinkedArrayList
 */
public static void test17(){
    printTestName( number: 17);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    print(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(0));
    print(linkedArrayList);

}
```



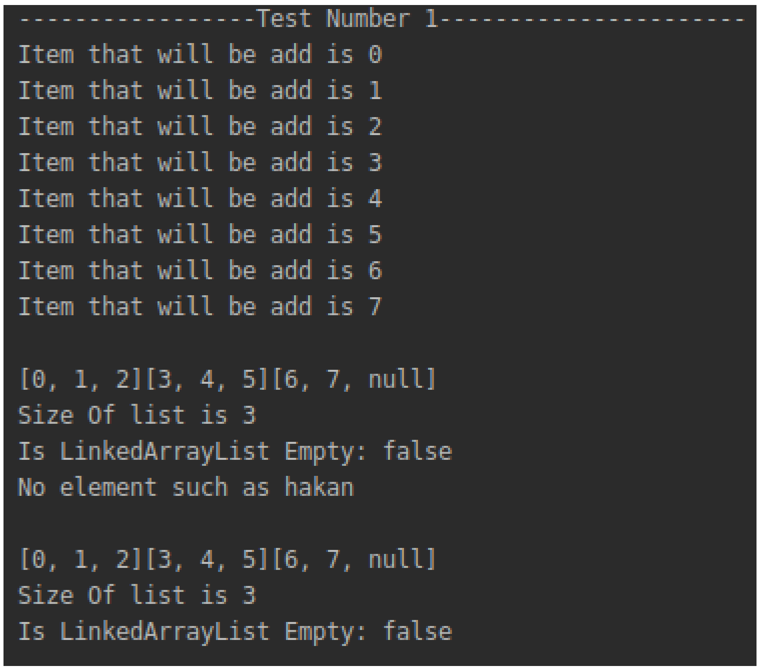
```
/**
 * Adding element at the end of the linkedarraylist
 */
public static void test21() {
    printTestName( number: 21);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    ListIterator<Integer> iterator = linkedArrayList.listIterator();
    print(linkedArrayList);
    iterator.add(5);
    print(linkedArrayList);
}

/**
 * Removing Empty LinkedArrayList
 */
public static void test22(){
    printTestName( number: 22);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    print(linkedArrayList);
    linkedArrayList.remove(Integer.valueOf(5));
}

/**
 * Testing removeRange(int fromIndex, int toIndex) Method
 */
public static void test23(){
    printTestName( number: 23);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    linkedArrayList.removeRange(0,2);
}
/**
 * Testing contains(Object o) Method
 */
public static void test24(){
    printTestName( number: 24);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    System.out.println("Does list contains value 3 : " + linkedArrayList.contains(Integer.valueOf(3)));
}
/**
```



```
/**
 * Testing sublist(int fromIndex, int toIndex) Method
 */
public static void test25(){
    printTestName( number: 25);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    System.out.println(linkedArrayList.subList(0, 2));
}
/**
 * Testing contains(Object o) Method
 */
public static void test26(){
    printTestName( number: 26);
    LinkedArrayList<Integer> linkedArrayList = new LinkedArrayList<>(CAPACITY_OF_ARRAYS);
    add(linkedArrayList);
    linkedArrayList.add(Integer.valueOf(3));
    System.out.println("Last Index of number 3 is " + linkedArrayList.lastIndexOf(Integer.valueOf(3)));
}
```

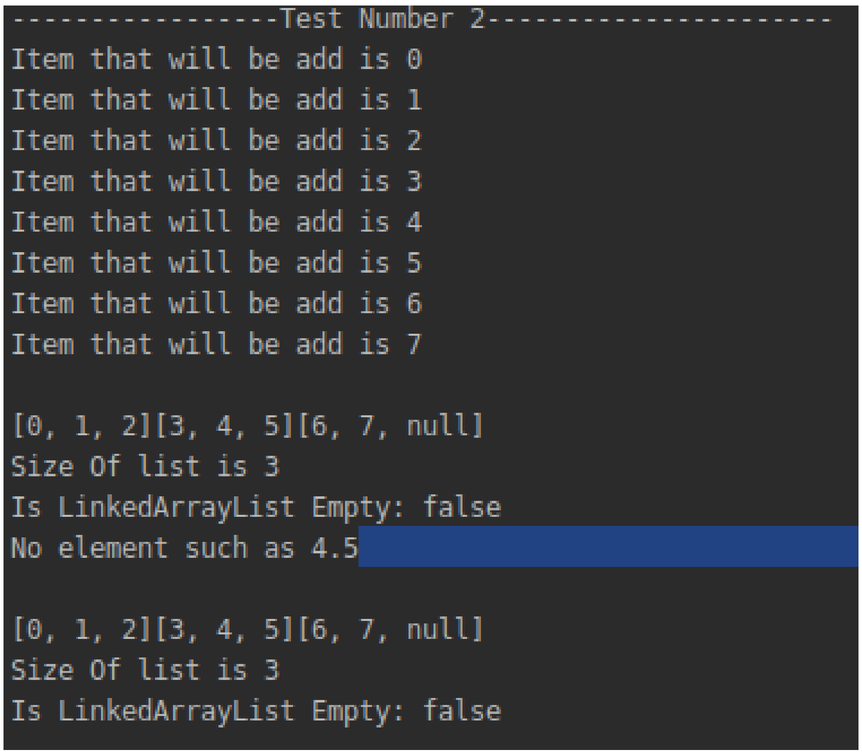
Running Commands And Results:

Test1:This test tests removing string from Integer object generic.



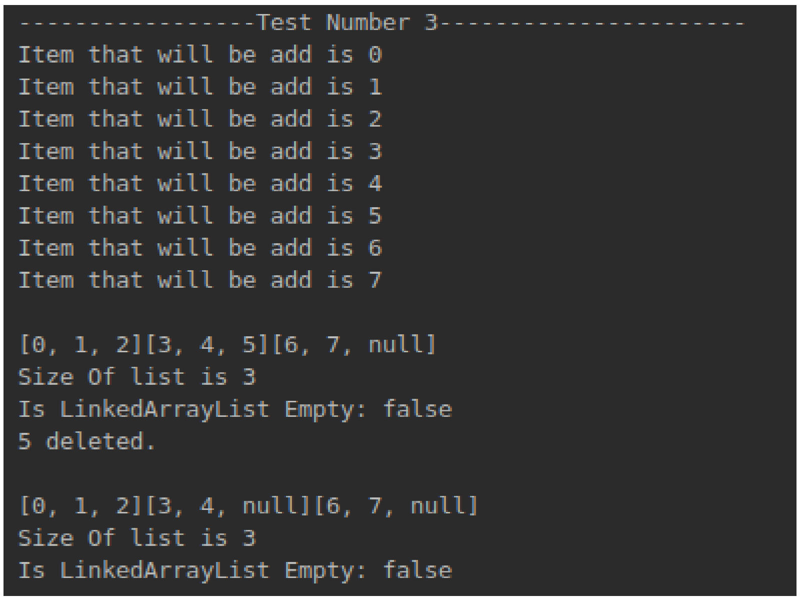
```
-----------------Test Number 1----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
No element such as hakan

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test2:This test tests removing double by generic.



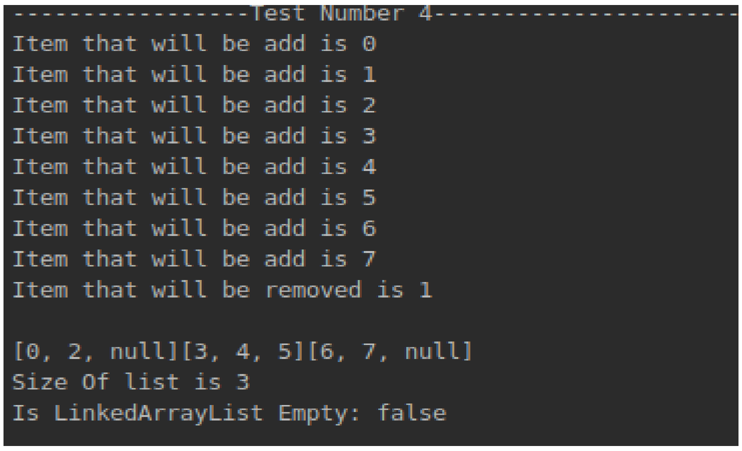
```
-----------------Test Number 2----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
No element such as 4.5

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test3: Remove given Integer object value



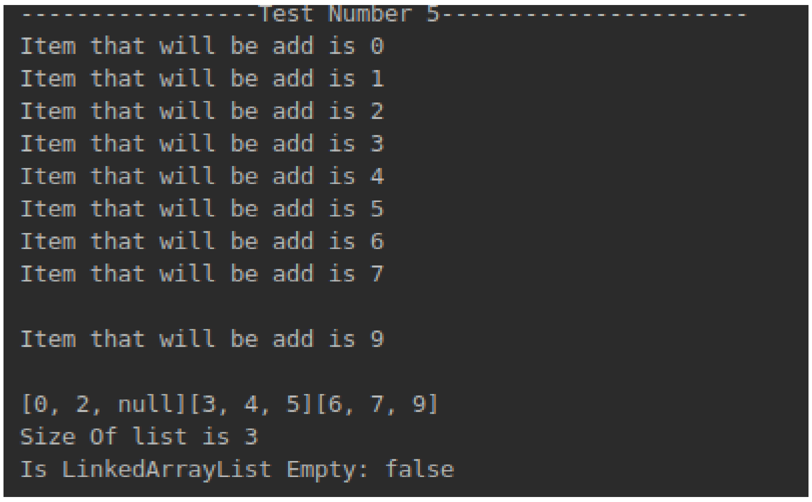
```
-----------------Test Number 3---------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
5 deleted.

[0, 1, 2][3, 4, null][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test4: Remove item to given iterator



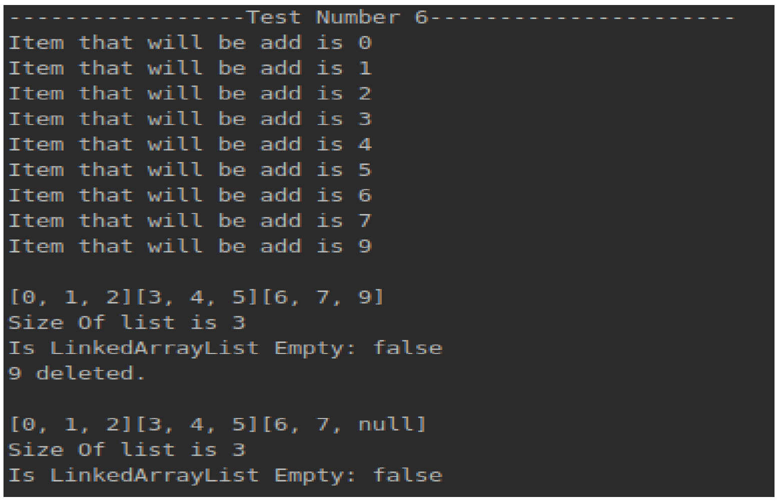
```
-----------------Test Number 4---------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
Item that will be removed is 1

[0, 2, null][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test5: Adding element if there is space by using iterator



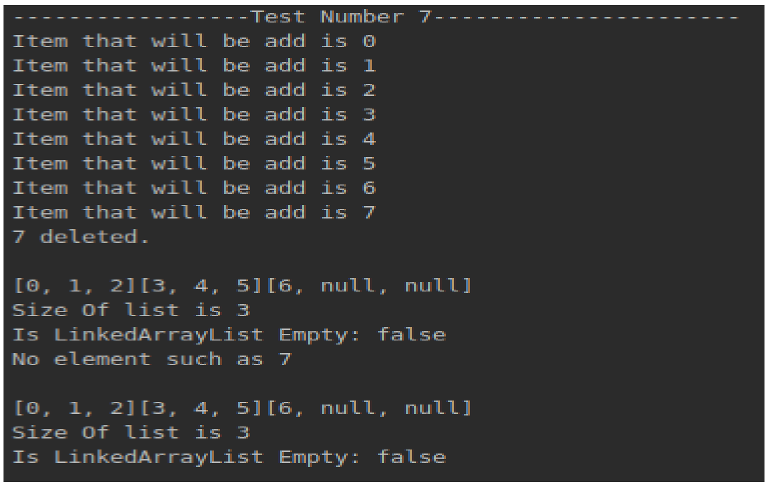
```
-----------------Test Number 5---------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

Item that will be add is 9

[0, 2, null][3, 4, 5][6, 7, 9]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test6: Remove element by using LinkedArrayList class method



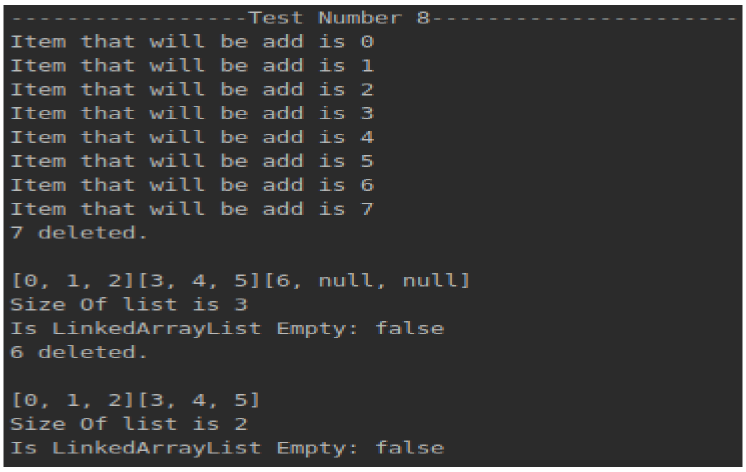
```
-----------------Test Number 6----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
Item that will be add is 9

[0, 1, 2][3, 4, 5][6, 7, 9]
Size Of list is 3
Is LinkedArrayList Empty: false
9 deleted.

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test7: Remove element which is not existed by using LinkedArrayList class method



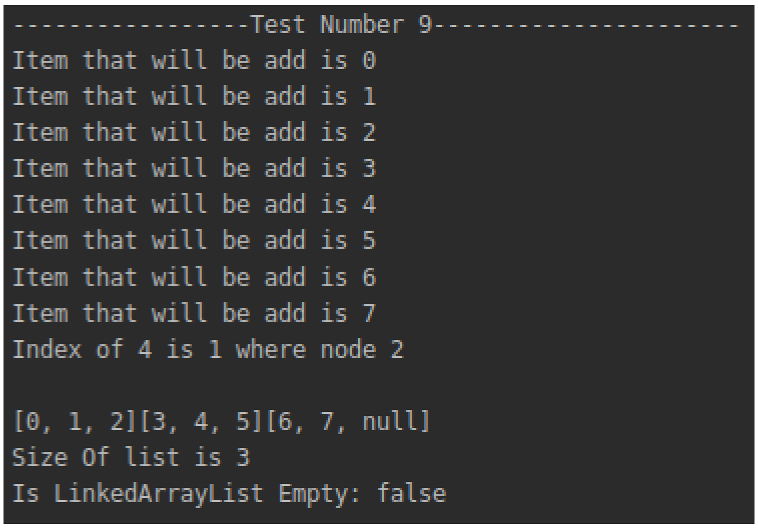
```
-----------------Test Number 7----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
7 deleted.

[0, 1, 2][3, 4, 5][6, null, null]
Size Of list is 3
Is LinkedArrayList Empty: false
No element such as 7

[0, 1, 2][3, 4, 5][6, null, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test8: Remove element and node deleted by using LinkedArrayList class method



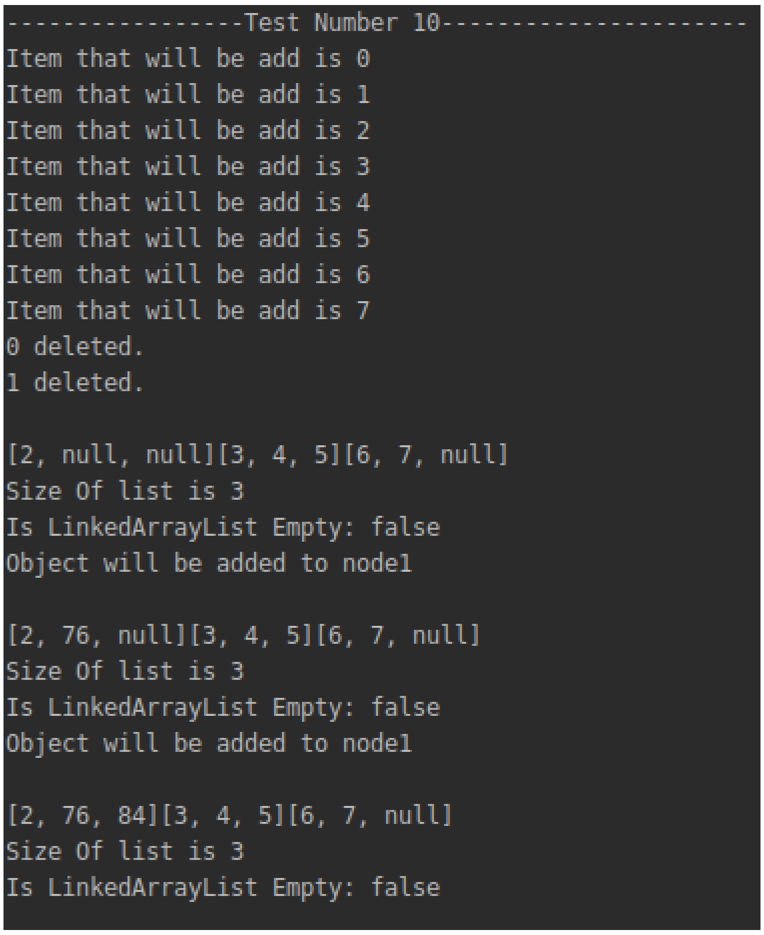
```
-----------------Test Number 8----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
7 deleted.

[0, 1, 2][3, 4, 5][6, null, null]
Size Of list is 3
Is LinkedArrayList Empty: false
6 deleted.

[0, 1, 2][3, 4, 5]
Size Of list is 2
Is LinkedArrayList Empty: false
```

Test9: Getting index of given value



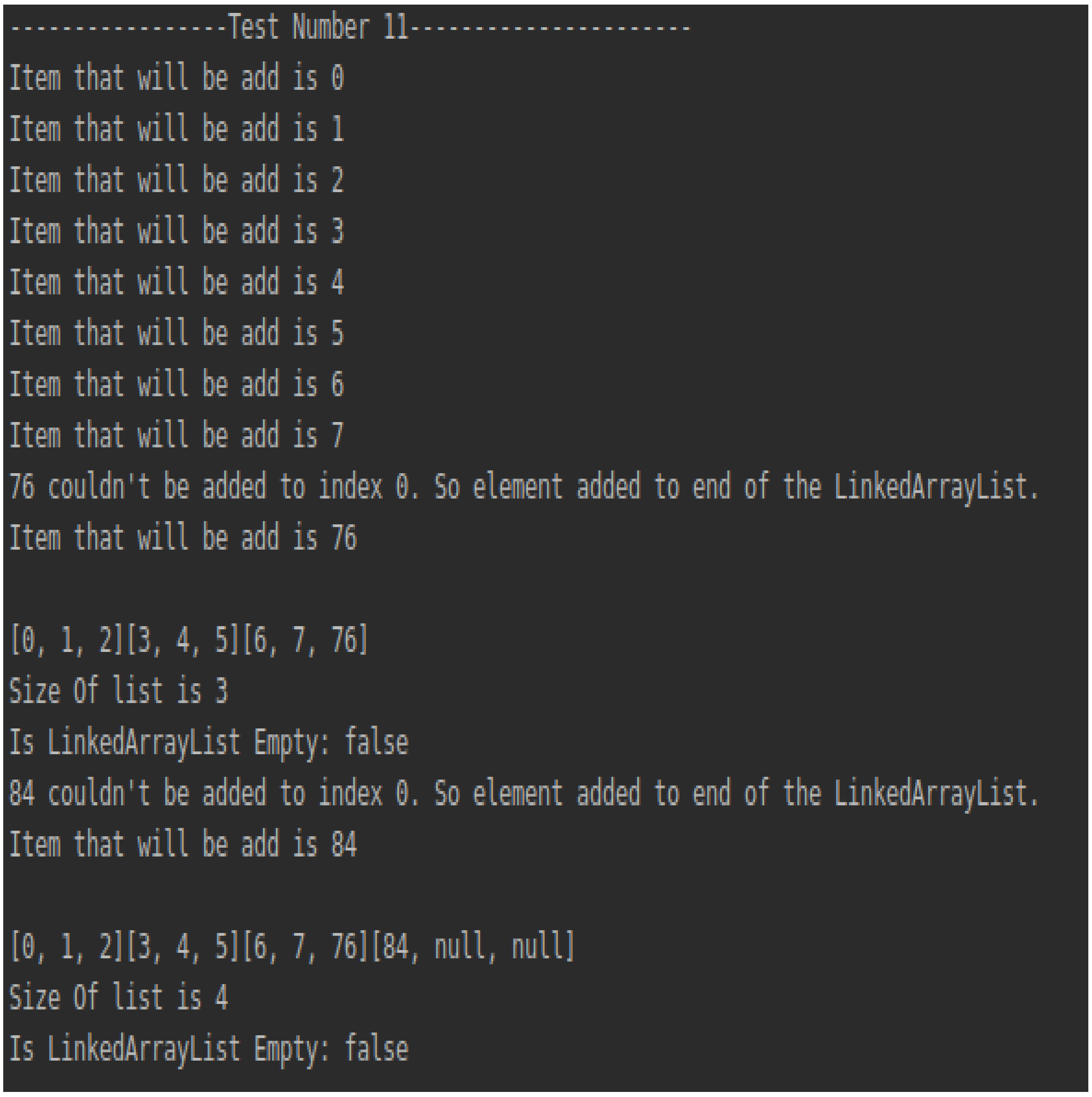
```
-----------------Test Number 9----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
Index of 4 is 1 where node 2

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test10:Adding eleemnt end of given the index



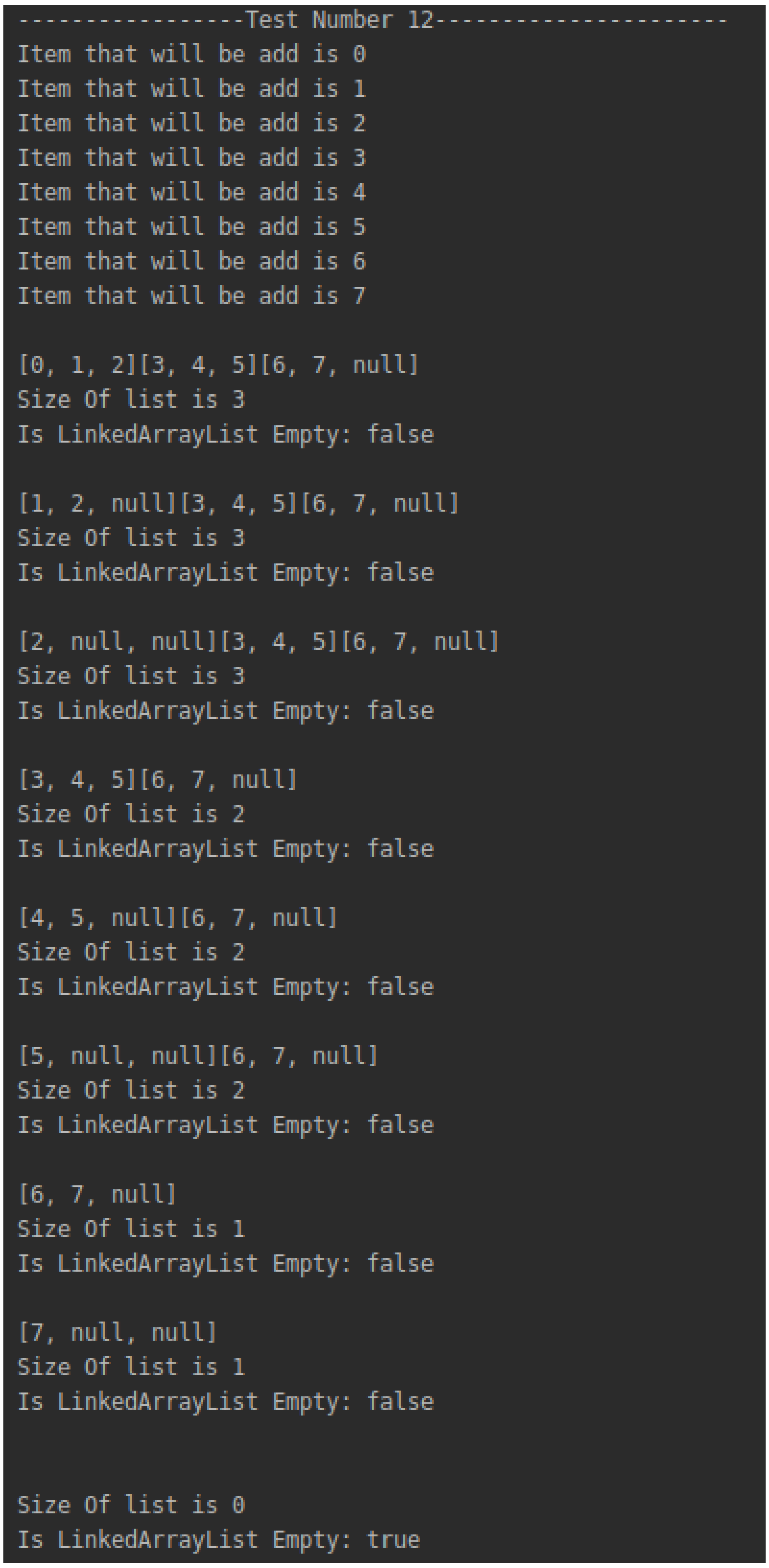
```
-----------------Test Number 10----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
0 deleted.
1 deleted.

[2, null, null][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Object will be added to node1

[2, 76, null][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Object will be added to node1

[2, 76, 84][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test11: Adding element at the end of the given list index and because of list is filled, value will be added at the end of the LinkedArrayList



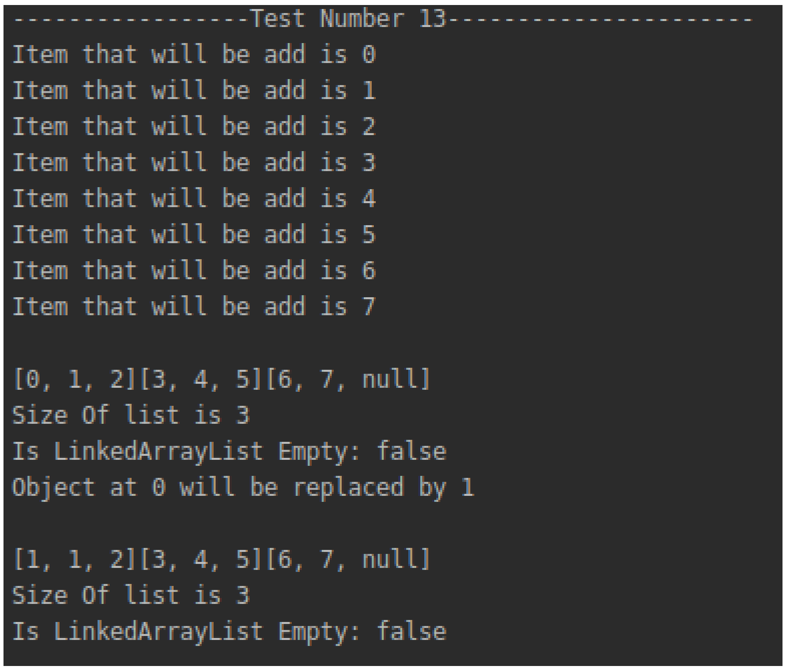
```
-----------------Test Number 11---------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
76 couldn't be added to index 0. So element added to end of the LinkedArrayList.
Item that will be add is 76

[0, 1, 2][3, 4, 5][6, 7, 76]
Size Of list is 3
Is LinkedArrayList Empty: false
84 couldn't be added to index 0. So element added to end of the LinkedArrayList.
Item that will be add is 84

[0, 1, 2][3, 4, 5][6, 7, 76][84, null, null]
Size Of list is 4
Is LinkedArrayList Empty: false
```

Test12: Removing all values inside list by using iterator



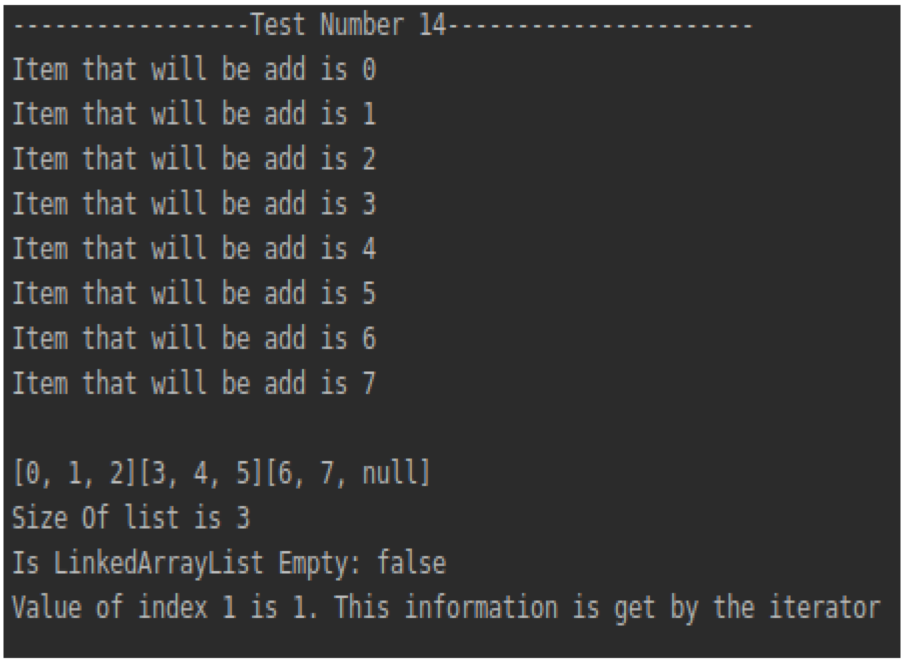
```
-----------------Test Number 12----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false

[1, 2, null][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false

[2, null, null][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false

[3, 4, 5][6, 7, null]
Size Of list is 2
Is LinkedArrayList Empty: false

[4, 5, null][6, 7, null]
Size Of list is 2
Is LinkedArrayList Empty: false

[5, null, null][6, 7, null]
Size Of list is 2
Is LinkedArrayList Empty: false

[6, 7, null]
Size Of list is 1
Is LinkedArrayList Empty: false

[7, null, null]
Size Of list is 1
Is LinkedArrayList Empty: false



Size Of list is 0
Is LinkedArrayList Empty: true
```

Test13: Setting value according to given value by iterator



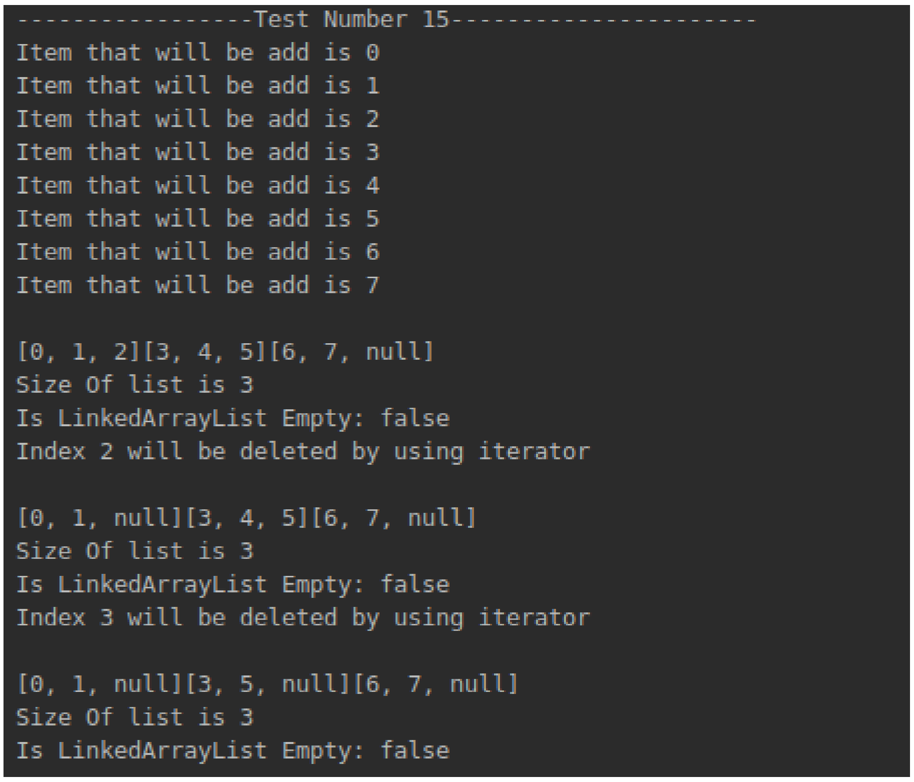
```
-----------------Test Number 13----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Object at 0 will be replaced by 1

[1, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test14: Checking listIterator(int index) method inside ListIterator class



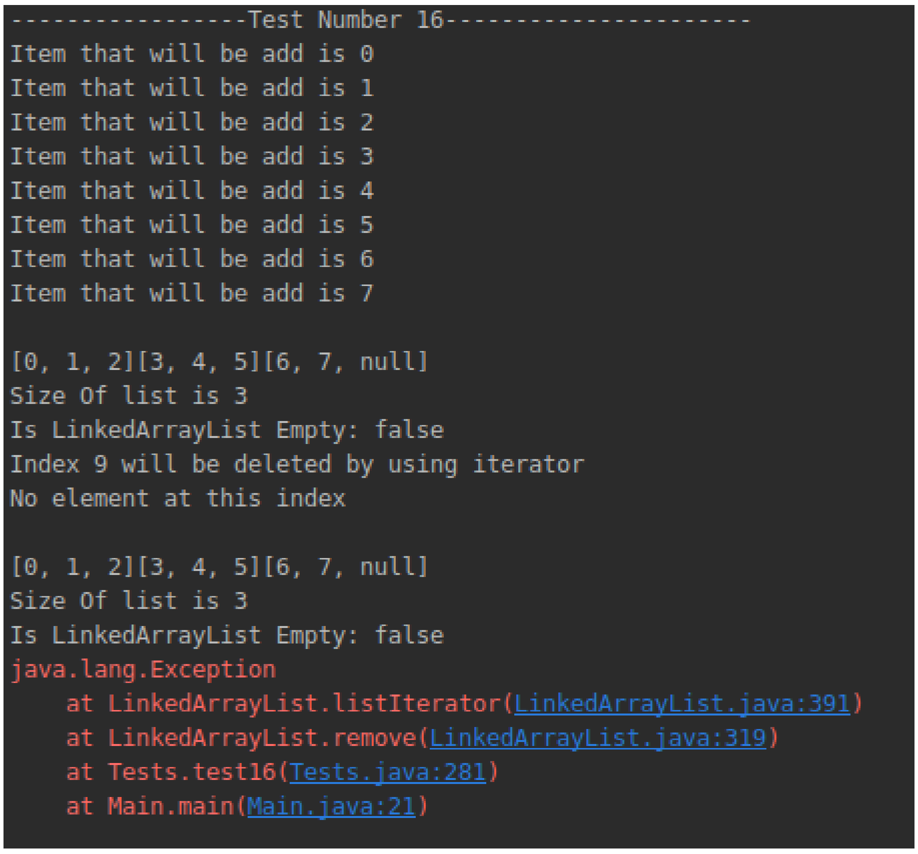
```
-----------------Test Number 14---------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Value of index 1 is 1. This information is get by the iterator
```

Test15:Remove according to indexes



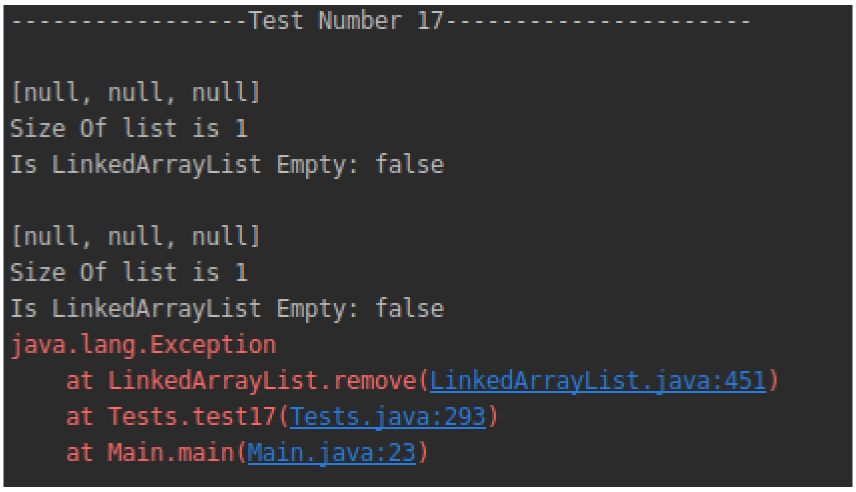
```
-----------------Test Number 15----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Index 2 will be deleted by using iterator

[0, 1, null][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Index 3 will be deleted by using iterator

[0, 1, null][3, 5, null][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test16:Remove out of bound index



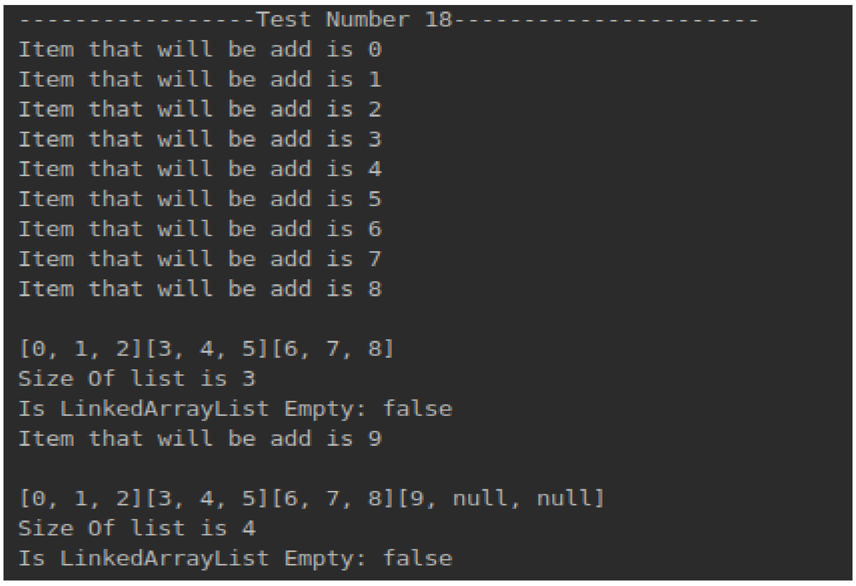
```
-----------------Test Number 16---------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Index 9 will be deleted by using iterator
No element at this index

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
java.lang.Exception
    at LinkedArrayList.listIterator(LinkedArrayList.java:391)
    at LinkedArrayList.remove(LinkedArrayList.java:319)
    at Tests.test16(Tests.java:281)
    at Main.main(Main.java:21)
```

Test17:Deleting empty LinkedArrayList



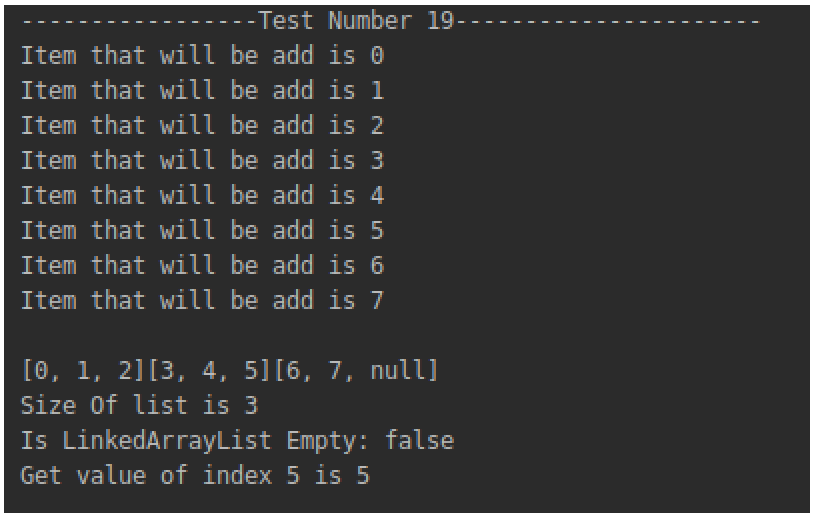
```
-----------------Test Number 17----------------------

[null, null, null]
Size Of list is 1
Is LinkedArrayList Empty: false

[null, null, null]
Size Of list is 1
Is LinkedArrayList Empty: false
java.lang.Exception
    at LinkedArrayList.remove(LinkedArrayList.java:451)
    at Tests.test17(Tests.java:293)
    at Main.main(Main.java:23)
```

Test18: Adding new element after that new array will be created



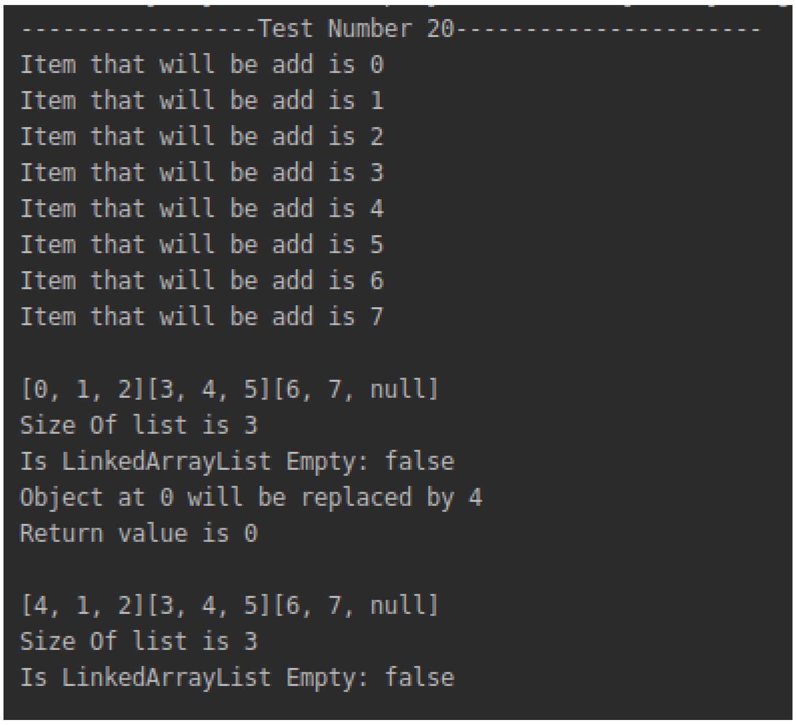
```
-----------------Test Number 18----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
Item that will be add is 8

[0, 1, 2][3, 4, 5][6, 7, 8]
Size Of list is 3
Is LinkedArrayList Empty: false
Item that will be add is 9

[0, 1, 2][3, 4, 5][6, 7, 8][9, null, null]
Size Of list is 4
Is LinkedArrayList Empty: false
```

Test19: Get method test



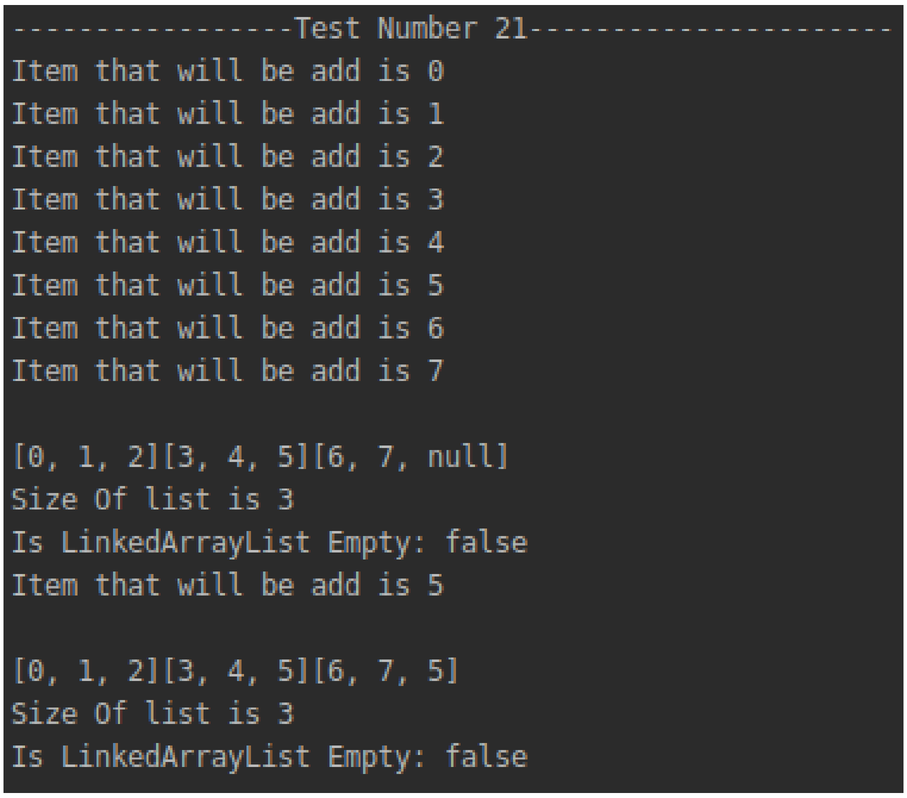
```
-----------------Test Number 19----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Get value of index 5 is 5
```

Test20: Set method test



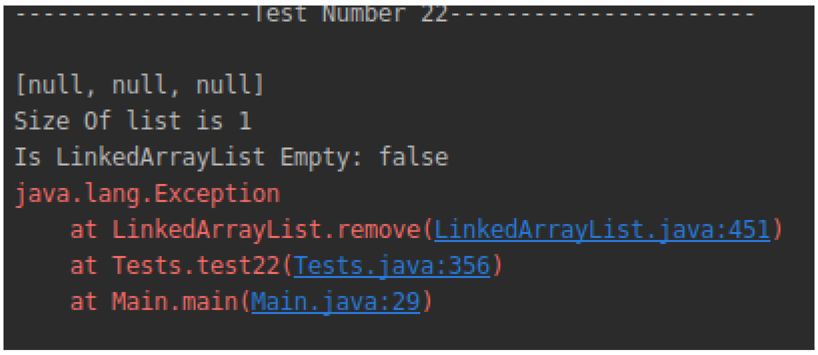
```
-----------------Test Number 20----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Object at 0 will be replaced by 4
Return value is 0

[4, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test21: Adding element at the end of the LinkedArrayList



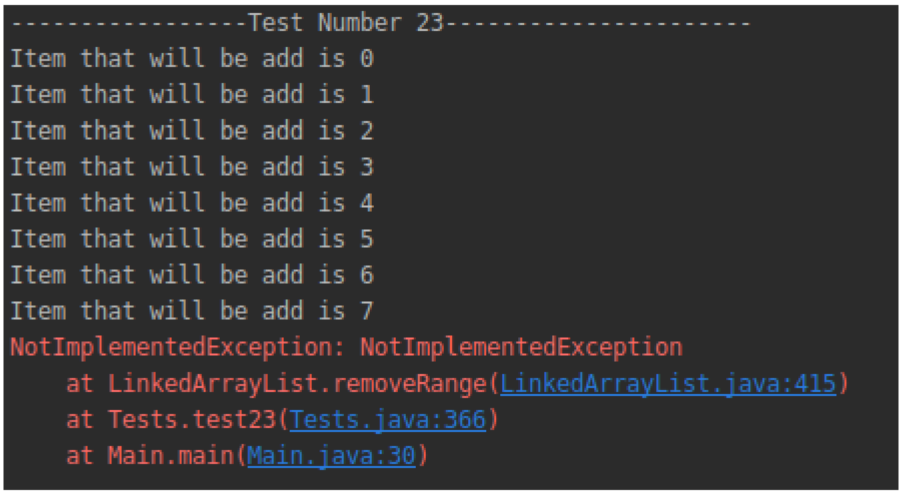
```
-----------------Test Number 21----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7

[0, 1, 2][3, 4, 5][6, 7, null]
Size Of list is 3
Is LinkedArrayList Empty: false
Item that will be add is 5

[0, 1, 2][3, 4, 5][6, 7, 5]
Size Of list is 3
Is LinkedArrayList Empty: false
```

Test22: Removing Empty LinkedArrayList



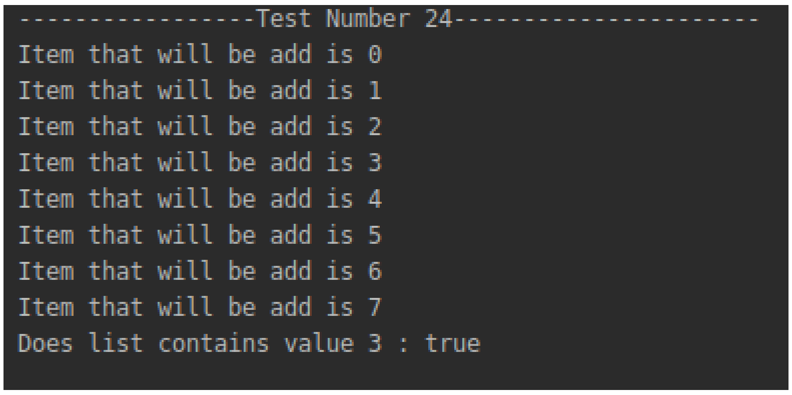
```
-----------------Test Number 22----------------------

[null, null, null]
Size Of list is 1
Is LinkedArrayList Empty: false
java.lang.Exception
    at LinkedArrayList.remove(LinkedArrayList.java:451)
    at Tests.test22(Tests.java:356)
    at Main.main(Main.java:29)
```

Test23: Testing removeRange(int fromIndex, int toIndex) method



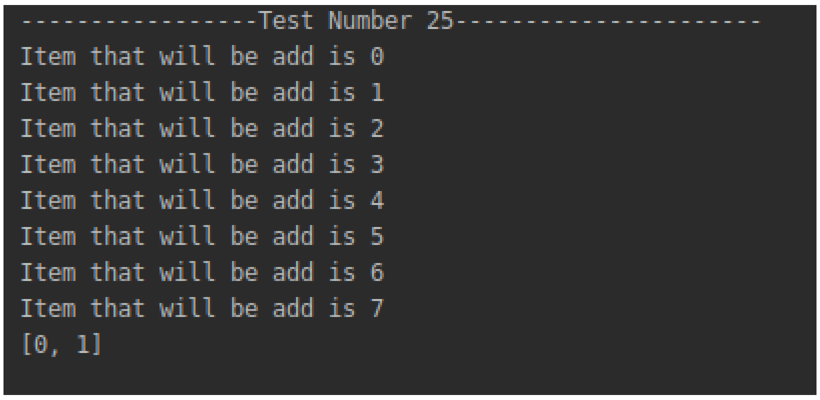
```
-----------------Test Number 23----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
NotImplementedException: NotImplementedException
    at LinkedArrayList.removeRange(LinkedArrayList.java:415)
    at Tests.test23(Tests.java:366)
    at Main.main(Main.java:30)
```

Test24: Testing contains(Object o) method



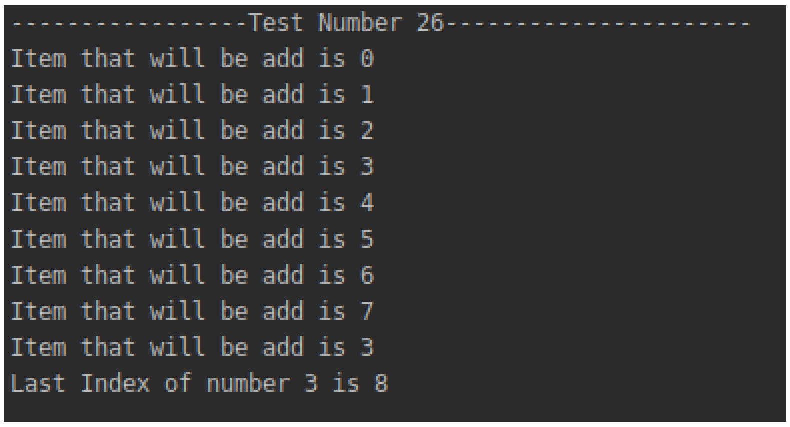
```
-----------------Test Number 24----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
Does list contains value 3 : true
```

Test25: Testing sublist(int fromIndex, int toIndex) method



```
-----------------Test Number 25----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
[0, 1]
```

Test26: Testing lastIndexOf(Object o) method



```
-----------------Test Number 26----------------------
Item that will be add is 0
Item that will be add is 1
Item that will be add is 2
Item that will be add is 3
Item that will be add is 4
Item that will be add is 5
Item that will be add is 6
Item that will be add is 7
Item that will be add is 3
Last Index of number 3 is 8
```