Student Name: **Natalia Reeck Zanini** Course: **CS 211**

Instructor: **Craig Niiyama**

QA Document for Assignment #9

Console Screenshot - Testing Methods:

```
Console
<terminated> Assignment9 [Java Application] /Library/Java/JavaVirtualMachines/jdk-14.0.1.jdk/Contents/Home/bin/java  (Jun 3, 2022,
Testing Methods:

Output for the lastIndexOf method:
-1

Output for the countDuplicates method:
7

First output for the hasTwoConsecutive method:
true

Second output for the hasTwoConsecutive method:
false

Outputs for the deleteBack method:

If the list is: front -> [1] -> [18] -> [17] -> [2] -> [7] -> [39] -> [18] -> [40] -> [8] /

The deleted value is: 8

If the list is empty:
Exception in thread "main" java.util.NoSuchElementException
        at LinkedIntList.deleteBack(LinkedIntList.java:324)
        at Assignment9.main(Assignment9.java:69)
```

Assignment9.java code:

CS-211 - Assignment #9/src/Assignment9.java - Eclipse IDE

LinkedIntList.java | *Assignment9.java

Assignment #9

```java
// Student: Natalia Reeck Zanini
// Course: CS 211
// Assignment #9
// June 2nd, 2022


// This file contains the code required to test the methods:
// lastIndexOf
// countDuplicates
// hasTwoConsecutive
// deleteBack

public class Assignment9{

    public static void main(String[] args) {

        System.out.println("Testing Methods:");
        System.out.println();

        int arr[] = {1, 18, 2, 7, 18, 39, 18, 40};

        int list2 = LinkedIntList.lastIndexOf(arr, 3);

        System.out.println("Output for the lastIndexOf method:");
        System.out.println(list2);
        System.out.println();


        int arr1[] = {1, 1, 1, 3, 3, 6, 9, 15, 15, 23, 23, 23, 40, 40};

        LinkedIntList list3 = new LinkedIntList(arr1);

        System.out.println("Output for the countDuplicates method:");
        System.out.println(list3.countDuplicates());
        System.out.println();

        int arr2[] = {1, 18, 2, 7, 8, 39, 18, 40};

        LinkedIntList list4 = new LinkedIntList(arr2);

        System.out.println("First output for the hasTwoConsecutive method:");
        System.out.println(list4.hasTwoConsecutive());
        System.out.println();

        int arr3[] = {1, 18, 17, 2, 7, 39, 18, 40, 8};

        LinkedIntList list5 = new LinkedIntList(arr3);

        System.out.println("Second output for the hasTwoConsecutive method:");
        System.out.println(list5.hasTwoConsecutive());
        System.out.println();


        int arr4[] = {1, 18, 17, 2, 7, 39, 18, 40, 8};
        LinkedIntList list6 = new LinkedIntList(arr4);

        System.out.println("Outputs for the deleteBack method:");

        System.out.println();
        System.out.println("If the list is: " + list6);
        System.out.println();
        System.out.println("The deleted value is: " + list6.deleteBack());
        System.out.println();

        int arr5[] = {};
        LinkedIntList list7 = new LinkedIntList(arr5);

        System.out.println("If the list is empty:");
        System.out.println(list7.deleteBack());

    }
}
```

Writable | Smart Insert | 70 : 9 : 1841

LinkedIntList.java - Methods code section:

CS-211 - Assignment #9/src/LinkedIntList.java - Eclipse IDE

LinkedIntList.java | *Assignment9.java

```java

    // YOUR CODE GOES HERE

    // #1) This method returns the index in the list of the last occurrence
    // of that value, or -1 if the value is not found in the list:

    public static int lastIndexOf(int[] arr, int num) {
        // Initiating index to -1:
        int index = -1;

        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == num) {
                index = i;
            }
        }
        return index;
    }

    // #2)This method returns the number of duplicates in a sorted list:

    public int countDuplicates() {

        //If the list is empty, returns zero:

        if (front == null) {
            return 0;
        }

        int count = 0;
        int curr = front.data;

        //Looping through remaining nodes:
        for (ListNode n = front.next; n != null; n = n.next) {
            if (n.data == curr) {
                count++;
            }
            else {
                curr = n.data;

            }
        }

        return count;

    }

    // #3) This method will return whether or not a list of integers has two
    // adjacent numbers that are consecutive integers:

    public boolean hasTwoConsecutive() {

        //Here if the list is empty/has only one element it will returning false:
        if (front == null || front.next == null) {
            return false;
        }

        ListNode n = front;
        //The loop will continue until n.next is null.
        while (n.next != null) {
            // Comparing n and n.next
            if (n.data == (n.next.data - 1)) {
                //If consecutive is found:
                return true;
            }

            //If not, it will advance to next node:
            n = n.next;
        }

        //In case it is not found:
        return false;

    }

```

Writable | Smart Insert | 42 : 5 : 1319

```java
    }

    // #4)This method deletes and returns the last value.
    // If the list is empty, then it throws NoSuchElementException error.

    public int deleteBack() {
        if(front == null)
            // If list is empty throws NoSuchElementException.
            throw new NoSuchElementException();

        int data;
        // If front is the last node it will remove the front.
        if(front.next == null) {
            data = front.data;
            front = null;
            // Will return the value.
            return data;
        }

        // The loop will continue until n.next is the last node.
        ListNode n = front;
        while(n.next.next != null)
            n = n.next;
        data = n.next.data;
        //removing n.next
        n.next = null;
        return data;
    }
}

```

Writable | Smart Insert | 316 : 6 : 9065