

Tyler Bartlett

5/6/18

CS260

## Lab 2 Write Up

Let me start by saying that I wrote this lab in Java. I wrote the majority of the DLList methods in C++ in Qtcreator but I ran into a compile error with QT. I am not familiar with Qtcreator (I mainly use Visual Studio) and could not figure out how to fix the error. The following screenshot is the bulk of the output for the error.

```
cl : Command line error D8021 : invalid numeric argument '/Wmissing-include-dirs'
jom: D:\Documents\source\QT\build-CS260_Lab2-Desktop_Qt_5.10.1_MSVC2017_64bit-Debug\Makefile.Debug [debug\main.obj] Error 2
jom: D:\Documents\source\QT\build-CS260_Lab2-Desktop_Qt_5.10.1_MSVC2017_64bit-Debug\Makefile [debug] Error 2
13:02:22: The process "C:\Qt\Tools\QtCreator\bin\jom.exe" exited with code 2.
Error while building/deploying project CS260_Lab2 (kit: Desktop Qt 5.10.1 MSVC2017 64bit)
The kit Desktop Qt 5.10.1 MSVC2017 64bit has configuration issues which might be the root cause for this problem.
When executing step "Make"
13:02:22: Elapsed time: 00:00.
```

I am not familiar with c++ anyways, so I decided to switch to java. Converting all the methods from c++ to java was fairly straight forward. I was running into some issues with my Node.java file and the DLList.java file so I scrapped the node file and made it an imbedded class in the DLList.java. I got all the methods written in java and started working on writing the testing class by hand converting it from c++ to java. I was not able to finish converting all the test cases but the ones I have got written, my DLList has passed so far. The tests I have completed up to this point have been: testing getSize, testing isEmpty, testing contains, testing add, and testing get (element). The following screenshots are the outputs of my code from those tests in the order listed.

```
Size must be 0 after constructor
Size is: 0
Size must be 0 after clear
Size is: 0
Size must be 1 after first add
Size is: 1
Size must be 2 after second add
Size is: 2
Size must be 3 after third add
Size is: 3
Size must be 4 after fourth add
Size is: 4
Size must be 5 after fifth add
Size is: 5
Size must be 4 after first remove
Size is: 4
Size must be 3 after second remove
Size is: 3
Size must be 0 after clear
Size is: 0
List is empty. nothing to print.
```

```
adding 10,20,30,40,15,25,35,45 to list
Contains returns false for 80
80 in list? - false
Contains returns false for 5
5 in list? - false
Contains returns true for 30
30 in list? - true
Contains returns true for 45
45 in list? - true
Contains returns true for 10
10 in list? - true
clearing list
Contains returns false for 45
45 in list? - false
```

```
isEmpty returns true after constructor
isEmpty: true
isEmpty returns true after clear
isEmpty: true
isEmpty returns false after adds (adding 2 things)
isEmpty: false
isEmpty returns false after adds (adding 3 more things)
isEmpty: false
isEmpty returns false after adds (adding 3 more things)
isEmpty: false
isEmpty returns true after clear
isEmpty: true
```

```
adding 10 to list
tail must be 10 after add
10
adding 20 to list
tail must be 20 after add
20
adding 30 to list
tail must be 30 after add
30
adding 40 to list
tail must be 40 after add
40
adding 15 to list
tail must be 15 after add
15
adding 25 to list
tail must be 25 after add
25
adding 35 to list
tail must be 35 after add
35
adding 45 to list
tail must be 45 after add
45
```

```
adding 10,20,30,40,15,25,35,45 to list
head of list must be 10 after the adds
value at head: 10
tail of list must be 45 after the adds
value at tail: 45
4th item of list must be 40 after the adds
value at head: 40
6th item of list must be 25 after the adds
value at head: 25
```

I have only just started working on testing the remove function of the DLList at 11pm 5/6/18 have gotten the first test for removing 99 complete but I am getting a null pointer exception after removing the 45 from my list.

```
adding 10,20,30,40,15,25,35,45 to list
remove returns false for 99
element not in list. Nothing to remove.
size must be 8 after failed remove. size: 8
remove returns true for 45:
Exception in thread "main" java.lang.NullPointerException
    at DLList.remove(DLList.java:200)
    at SystemTest.testRemove(SystemTest.java:222)
    at Driver.main(Driver.java:18)
PS E:\Source Code\Java\CS260_Lab2>
```

I was having a bit of trouble when I began my testing. The clear method was giving me a null pointer exception also but I have since fixed that. There was (and probably still is) a small issue with the removeAt(index) method for when the list is of size 0 or 1. I don't think I am removing everything from the list in that which I believe would lead to a memory leak? I was able to avoid the null pointer exception with an if statement but have not gotten around to thoughtfully looking at it and trying to fix it. The following is a screenshot of that.

```
public T removeAt(int index)
{
    if (index < 0 || index > size-1)
    {
        System.out.println("Error: bad index");
        System.exit(0);
    }

    T tempElem = null;
    if (index == 0) // delete node at head
    {
        if (size == 0 || size == 1)
        {
            //doing this might make a memory leak???
            //but it stops the NullPointerException
        }
        else
        {
            tempElem = head.element;
            head = head.next;
            head.prev.next = null;
            head.prev = null;
        }
    }
}
```

The resources I looked at for this lab were:

<http://www.java2novice.com/data-structures-in-java/linked-list/doubly-linked-list/>

which helped me resolve the issue I ran into with my node file. This was why I made my node class imbedded into the DLList.

I also looked at:

<https://www.geeksforgeeks.org/delete-a-node-in-a-doubly-linked-list/>  
<https://www.geeksforgeeks.org/delete-a-node-in-a-doubly-linked-list/>

to help me resolve the first null pointer exception I had with my removeAt method.

I am in a discord chat with some of my class mates but I did not utilize their help in working on the lab as I wrote in java and I believe they are all using c++.

The test file I was using to write my tests in java was based off the email from Breann Flesch that I believe came from a student named Mike.

I wrote some methods for the Queue and my code compiled but I was not able to test them to see if they worked.