LabReport02

1. **Problem:** Create a doubly linked list of the data type double, which shares similarities with a singly linked list, but with the added feature that each node contains a previous link in addition to its data and next link.

2. **Solution Description:** You would want to write code that implements a doubly-linked list, where each node in the list contains a double value and references to the previous and next nodes in the list. The implementation includes methods for checking if a value is contained in the list, navigating the list using a current node pointer, modifying the value of the current node, and printing the entire list.

3. **Problems Encountered:** My brain tends to think faster than my hands can move and I mostly had syntax errors.

4. When an object in Java becomes unreachable, meaning that no active reference points to it and it cannot be accessed by any part of the program, the Java Virtual Machine (JVM) will mark the object as eligible for garbage collection.

5. Linked Lists compared to Arrays

Advantages

- Dynamic size allows adding and removing of elements
- Efficient adding and removing

Disadvantages

- Must move through all other nodes to get to a certain node

6. Doubly Linked List compared to Singly Linked List

Advantages

- Each node contains pointers in both directions, prev and next
- Circular lists without copying elements
- Efficient at adding and removing on both ends of the list

Disadvantages

- More complex and requires more code

7. This code does not work as intended, It has a syntax error in within and is checking the current node(temp.link) if it is null. It should be checking if temp in its entirety is null. The while loop should also print temp.data not temp. Fixed by changing temp.link in while (temp.link != null) to temp and changing temp in System.out.println(temp); to temp.data and adding temp = temp.link;

LabReport02

8. Yes it does and prints Abc.

9. This code does not work as intended, there is a syntax error, also considered logic error, in temp = rValue; It is trying to put an entire linked list into a String value, instead should be temp.data = rValue;. There is also a runtime error, the line head = head.link; needs to be changed to temp = temp.link;.

10. Tt only advances the "temp" variable five nodes ahead of the head, leaving the original nodes in place and unconnected to the rest of the list. To remove the first five elements, the "head" variable must be updated to point to the sixth node in the list. This can be done by adding the line "head = temp.link;" at the end of the "removeFirst5" method.