**Homework2 Report**

**Purpose**

In this homework, my purpose is implementing interface which given in homework and create algorithm to method which given in interface with use double linked list.

**How did I solve problems**

I define first that show first of linked list and end that show end of linked list using encapsulation.Also, I created LinkedListJava constructor that equals first and end null.

My first problem is determining the special cases in insert method. I decide to linked list is empty, add to start of the linked list and add to end of the linked list are special cases. So, I can add node to the position which I want. Also, position is negative and position larger size of linked list are special case and if position is like that, throw exception.

My second problem is determining the special case in delete method. I decide to special cases are like insertion method. So, I can delete node to the position which I want. Also, this method return deleted element of deleted node.

My third problem is how can I print the elements in Output and ROutput method. I print the elements from start to finish in Output and from finish to start in ROutput with using first and end.

My fourth problem is how can I create new link list which reverse of old linked list in ReverseLinkedList method. I make all right of nodes equal left of nodes and left of nodes equal right of nodes. So, fourth problem is solved.

My fifth problem is how can I get Squashl method requirements. Insert element and how many side by side elements are.

My sixth problem is how can I get Opslashl method requirements. Insert element times element to the right of this element.

My seventh problem is how can I print linked list as string in toString method. All elements to convert string.

My eighth pronlem is how can I create new exception in own package. I create new class which name is HW2Exception and extends exception. Also, throw HW2Exception with a message in LinkedListException method.

I use coding for my algorithm Netbeans with java programming language.

**What complexity I have**

My Output and ROutput method complexity is O(n).

**Conclusion**

Advantages:

-We can navigate through the list as well as end-to-end.

-It is easy to reverse the list.

-When we are in an item, we can return to the previous item. On single linked lists this was not possible.

Disadvantages:

Because we have an extra pointer to the previous item, it requires more memory space per item.Adding and deleting operations take longer than unidirectional linked lists because they require more pointer operations than single linked lists.

**Adem Savaş**

**34435456408**