

**GTU Department of Computer Engineering**

**CSE 222 Spring 2020**

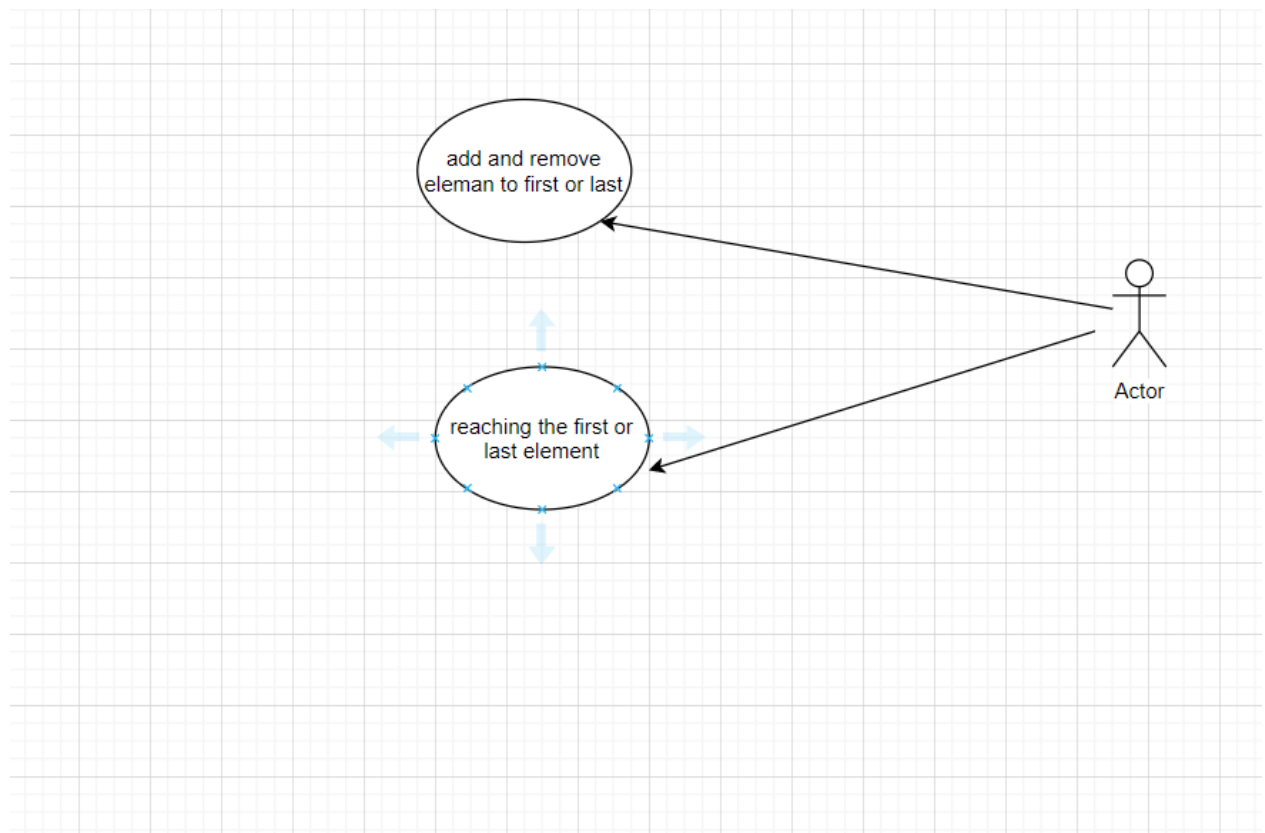
**Homework 4 Report**

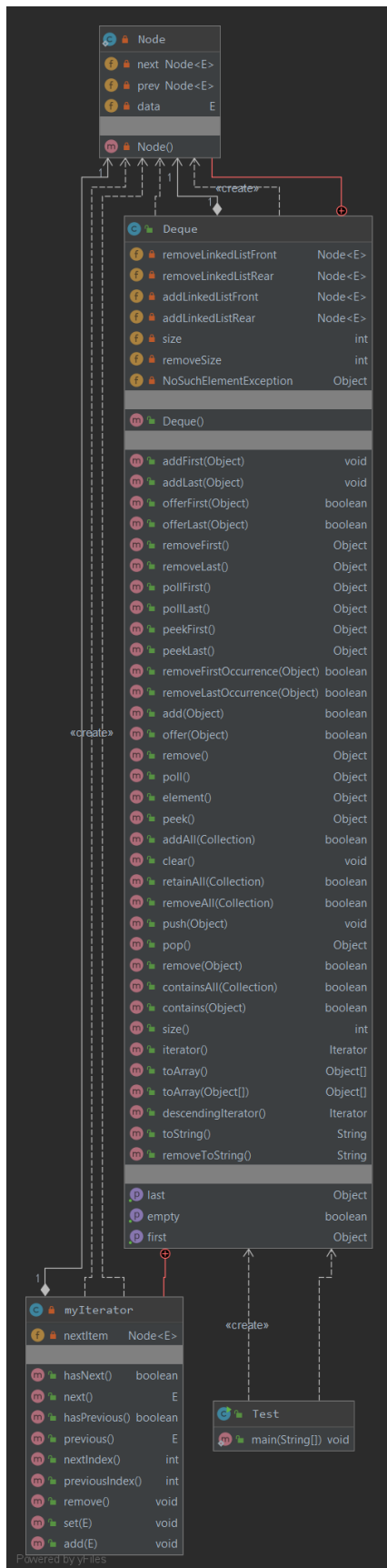
**Fatih OĞUZ**

**151044025**

## PART2

### DIAGRAMS





## Problem Solutions Approach

According to the Deque interface, the list must have a front and rear part, since it will be added to the beginning and end of the list. Also, in order to avoid the use of garbage collector in this project, the deleted items should be gathered in a list. This list called removeLinkedList. removeLinkedList must have front and rear. When add Linked List adds, if it has a remove LinkedList node, it will add its node or create a new node. RemovedLinkedList adds all deleted nodes. In addition, the iterator class has been written but not implemented. Because our list adds and remove from the ends. Front and rear of linked lists kept private, the loop is avoided and iterator is not needed.

## TEST CASES && RUNNING AND RESULTS

All  
methods

```
System.out.println("addFirst ==> 23" );
deque.addFirst( 0: 23);
System.out.println("addFirst ==> 53" );
deque.addFirst( 0: 53);
System.out.println("offerFirst ==> 69");
deque.offerFirst( 0: 69);
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("addLast ==> 100");
deque.addLast( 0: 100);
System.out.println("offerLast ==> 250");
deque.offerLast( 0: 250);
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("removeFirst ");
deque.removeFirst();
System.out.println("removeLast ");
deque.removeLast();
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
```

```
addFirst ==> 23
addFirst ==> 53
offerFirst ==> 69
addLinkedList of deque 69 53 23
removeLinkedList of deque Removed nodes

addLast ==> 100
offerLast ==> 250
addLinkedList of deque 69 53 23 100 250
removeLinkedList of deque Removed nodes

removeFirst
removeLast
addLinkedList of deque 53 23 100
removeLinkedList of deque Removed nodes 69 250
```

```
System.out.println("pollFirst ");
deque.pollFirst();
System.out.println("pollLast ");
deque.pollLast();
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("push 55");
deque.push(0: "55");
System.out.println("push 85");
deque.push(0: "85");
System.out.println("*** use node of removedLinkedList for adding ***");
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("getFirst ==> " + deque.getFirst());
System.out.println("getLast ==> " + deque.getLast());
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("peekFirst " + deque.peekFirst());
System.out.println("peekLast " + deque.peekLast());
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
```

```
pollFirst
pollLast
addLinkedList of deque 23
removeLinkedList of deque Removed nodes 69 250 53 100

push 55
push 85
*** use node of removedLinkedList for adding ***
addLinkedList of deque 85 55 23
removeLinkedList of deque Removed nodes 69 250

getFirst ==> 85
getLast ==> 23
addLinkedList of deque 85 55 23
removeLinkedList of deque Removed nodes 69 250

peekFirst 85
peekLast 23
addLinkedList of deque 85 55 23
removeLinkedList of deque Removed nodes 69 250
```

```
System.out.println("add ==> 99 " + deque.add(99));
System.out.println("*** use node of removedLinkedList for adding ***");
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("offer ==> 44 " + deque.offer(0, 44));
System.out.println("*** use node of removedLinkedList for adding ***");
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("poll==> " + deque.poll());
System.out.println("removeFirst ==> " + deque.removeFirst());
System.out.println("remove ==> " + deque.remove());
System.out.println("pop ==> " + deque.pop());
System.out.println("size ==> " + deque.size());
System.out.println("isEmpty ==> " + deque.isEmpty());
System.out.println("element ==> " + deque.element());
System.out.println("peek ==> " + deque.peek());
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
System.out.println("removeFirst ==>" + deque.removeFirst());
System.out.println("isEmpty ==> " + deque.isEmpty() );
System.out.println("size ==> " + deque.size());
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
```



```
add ==> 99 true
*** use node of removedLinkedList for adding ***
addLinkedList of deque 85 55 23 99
removeLinkedList of deque Removed nodes 69

offer ==> 44 true
*** use node of removedLinkedList for adding ***
addLinkedList of deque 85 55 23 99 44
removeLinkedList of deque Removed nodes

poll==> 85
removeFirst ==> 55
remove ==> 44
pop ==> 23
size ==> 1
isEmpty ==> false
element ==> 99
peek ==> 99
addLinkedList of deque 99
removeLinkedList of deque Removed nodes 85 55 44 23

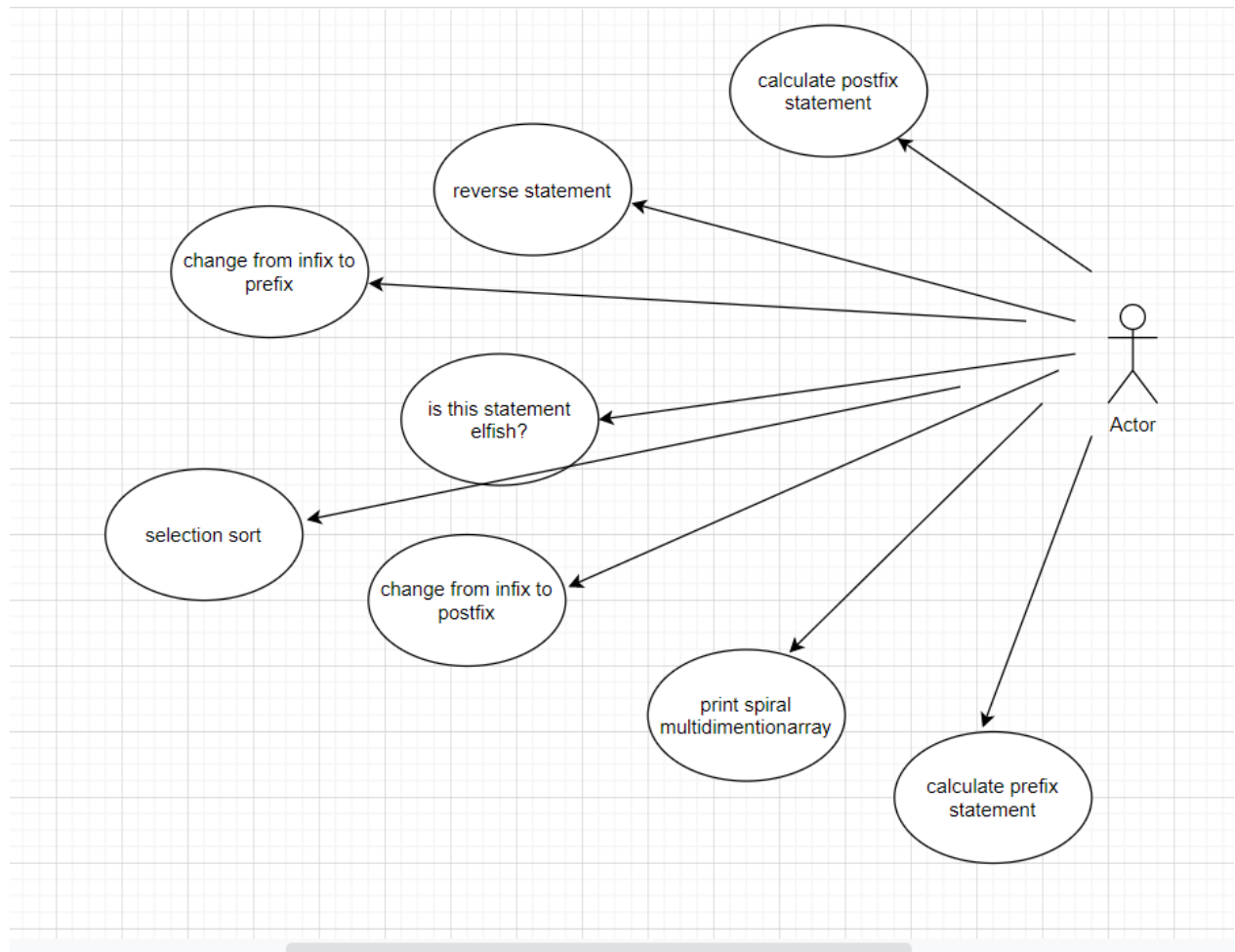
removeFirst ==>99
isEmpty ==> true
size ==> 0
addLinkedList of deque
removeLinkedList of deque Removed nodes 85 55 44 23 99
```

```
System.out.println("removeFirst==>" + deque.removeFirst());
System.out.println("addLinkedList of deque " + deque);
System.out.println("removeLinkedList of deque " + deque.removeToString());
System.out.println("\n");
```

```
*****  
removeFirst==>null  
addLinkedList of deque  
removeLinkedList of deque Removed nodes 85 55 44 23 99
```

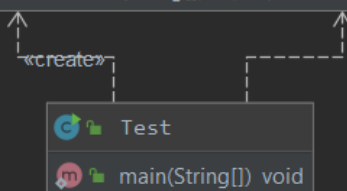
## PART3

## DIAGRAMS



ProblemInterface		
m	split(String)	String[]
m	reverseHelp(String[], int)	String[]
m	reverse(String)	String
m	elfish(String)	boolean
m	elfishHelp(String, int, int, int, int, int)	boolean
m	minIndex(int[], int, int)	int
m	recurSelectionSort(int[], int, int)	void
m	evaluatePostfix(ArrayList, ArrayList)	String
m	evaluatePostfix(ArrayList, ArrayList, ArrayList)	ArrayList
m	evaluatePrefix(ArrayList, ArrayList)	String
m	evaluatePrefix(ArrayList, ArrayList, ArrayList)	ArrayList
m	print(Integer[], int, int, int, int, int, int)	void
m	reverseHelp2(String, String[], int, int)	String
m	evaluatePostfixOrPrefixHelp(String, ArrayList, int, int)	String
m	evaluateResultPostfix(String[], int, int)	void
m	evaluateResultPrefix(String[], int, int)	void

Problem		
m	split(String)	String[]
m	reverseHelp(String[], int)	String[]
m	reverseHelp2(String, String[], int, int)	String
m	reverse(String)	String
m	elfish(String)	boolean
m	elfishHelp(String, int, int, int, int, int)	boolean
m	minIndex(int[], int, int)	int
m	recurSelectionSort(int[], int, int)	void
m	evaluatePostfixOrPrefixHelp(String, ArrayList, int, int)	String
m	evaluatePostfix(ArrayList, ArrayList)	String
m	evaluatePostfix(ArrayList, ArrayList, ArrayList)	ArrayList
m	evaluatePrefix(ArrayList, ArrayList)	String
m	evaluatePrefix(ArrayList, ArrayList, ArrayList)	ArrayList
m	print(Integer[], int, int, int, int, int, int)	void
m	evaluateResultPostfix(String[], int, int)	void
m	evaluateResultPrefix(String[], int, int)	void



## Problem Solutions Approach

Write recursively the methods in this interface into simple parts and to come to a conclusion by combining these parts

## Test Cases & Running Command and Results

```
Problem problem = new Problem();

String str10 = "this function writes the sentence in reverse";
System.out.println("reverse() Function==>> " + str10 + " ==> " + problem.reverse(str10));

*****/

System.out.println("elfish whiteleaf " + problem.elfish(str: "whiteleaf"));
System.out.println("elfish tasteful " + problem.elfish(str: "tasteful"));
System.out.println("elfish unfriendly " + problem.elfish(str: "unfriendly"));
System.out.println("elfish waffles " + problem.elfish(str: "waffles"));
System.out.println("elfish fatih " + problem.elfish(str: "fatih"));

*****/

reverse() Function==>> this function writes the sentence in reverse ==> reverse in sentence the writes function this
elfish whiteleaf true
elfish tasteful true
elfish unfriendly true
elfish waffles true
elfish fatih false
```

```
System.out.print("selecction sort ==> ");
int arr[] = {3, 1, 5, 2, 7, 0};

problem.recurSelectionSort(arr, arr.length, index: 0);

for (int i = 0; i < arr.length; i++)
    System.out.print(arr[i] + " ");
System.out.println("");
```

```
elfish fatih false
selecction sort ==> 0 1 2 3 5 7
```

```

ArrayList<Character> infix = new ArrayList<>();
String str = "A+((B-C*D)/E)+F-G/H";
for(int i = 0;i<str.length();i++){
    infix.add(str.charAt(i));
}
ArrayList<Character> res = new ArrayList<>();
System.out.println("Evaluate postfix  A+((B-C*D)/E)+F-G/H ==> " + problem.evaluatePostfix(res,infix));

```

\*\*\*\*\*

```

ArrayList<Character> infix1 = new ArrayList<>();
for(int i = 0;i<str.length();i++){
    infix1.add(str.charAt(i));
}
ArrayList<Character> res1 = new ArrayList<>();
System.out.println("Evaluate prefix  A+((B-C*D)/E)+F-G/H ==> " + problem.evaluatePrefix(res1,infix1));

```

```

Evaluate postfix  A+((B-C*D)/E)+F-G/H ==> ABCD*-E/+F+GH/-
Evaluate prefix  A+((B-C*D)/E)+F-G/H ==> -++A/-B*CDEF/GH

```

```

String[] strings = {"24","40","6","5","*","-","5","/","+","4","+","10","2","/","-"};
System.out.println("24 40 6 5 * - 5 / + 4 + 10 2 / -");
System.out.println("evaluate postfix");
problem.evaluateResultPostfix(strings, count: 0,strings.length);

```

```

24 40 6 5 * - 5 / + 4 + 10 2 / -
evaluate postfix
6 * 5 = 30
40 - 30 = 10
10 / 5 = 2
24 + 2 = 26
26 + 4 = 30
10 / 2 = 5
30 - 5 = 25

```

\*\*\*\*\*

```

System.out.println("*****");
System.out.println("- + + 24 / - 40 * 6 5 5 4 / 10 2");
System.out.println("evaluate prefix");
String[] strings2 = {"-", "+", "+", "24", "/", "-", "40", "*", "6", "5", "5", "4", "/", "10", "2"};
problem.evaluateResultPrefix(strings2, count: strings2.length-1, strings2.length);

```

```

- + + 24 / - 40 * 6 5 5 4 / 10 2
evaluate prefix
10 / 2 = 5
6 * 5 = 30
40 - 30 = 10
10 / 5 = 2
2 + 24 = 26
4 + 26 = 30
30 - 5 = 25

```

```

System.out.println("*****");
Integer[][] multidimensionalArray = {
    {1 ,2 ,3 ,4,1},
    {5, 6 ,7 ,8,2},
    {9 ,10 ,11 ,12,3},
    {13 ,14 ,15 ,16,4},
    {17 ,18 ,19 ,20,5},
    {17 ,18 ,19 ,20,6},
    {1 ,2 ,3 ,4,7},
    {5, 6 ,7 ,8,8},
};
for(int i=0;i<8;i++){
    for(int j = 0 ;j<5;j++){
        System.out.print(multidimensionalArray[i][j] + " ");
    }
    System.out.println();
}
System.out.println();
problem.print(multidimensionalArray, row_index: 0, column_index: 0, aspect: 0, frekans_row: 5, frekans_column: 7, frekans_control: 0);
System.out.println();
System.out.println("*****");

```

```

*****
1 2 3 4 1
5 6 7 8 2
9 10 11 12 3
13 14 15 16 4
17 18 19 20 5
17 18 19 20 6
1 2 3 4 7
5 6 7 8 8

1 2 3 4 1 2 3 4 5 6 7 8 8 7 6 5 1 17 17 13 9 5 6 7 8 12 16 20 20 4 3 2 18 18 14 10 11 15 19 19

*****

```

```

Integer[][] multidimensionalArray1 = {
    {1, 2, 3, 4},
    {5, 6, 7, 8},
    {9, 10, 11, 12},
    {13, 14, 15, 16},
    {17, 18, 19, 20},
    {17, 18, 19, 20},
};

for(int i=0; i<6; i++){
    for(int j = 0 ;j<4;j++){
        System.out.print(multidimensionalArray[i][j] + " ");
    }
    System.out.println();
}
System.out.println();

problem.print(multidimensionalArray1, row_index: 0, column_index: 0, aspect: 0, frekans_row: 4, frekans_column: 5, frekans_control: 0);
System.out.println();
System.out.println("*****");

```

```

*****
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
17 18 19 20

1 2 3 4 8 12 16 20 20 19 18 17 17 13 9 5 6 7 11 15 19 18 14 10

*****
1 2 3 4

```



```

Integer[][] multidimensionalArray2 ={
    {1 ,2 ,3 ,4},
    {5, 6 ,7 ,8},
    {9 ,10 ,11 ,12},
    {13 ,14 ,15 ,16},
    {17 ,18 ,19 ,20},
};
for(int i=0;i<5;i++){
    for(int j = 0 ;j<4;j++){
        System.out.print(multidimensionalArray[i][j] + " ");
    }
    System.out.println();
}
System.out.println();
problem.print(multidimensionalArray2, 'row_index: 0, 'column_index: 0, 'aspect: 0, 'frekans_row: 4, 'frekans_column: 4, 'frekans_control: 0);
System.out.println();
System.out.println("*****");

```

```

*****
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20

1 2 3 4 8 12 16 20 19 18 17 13 9 5 6 7 11 15 14 10

*****

```