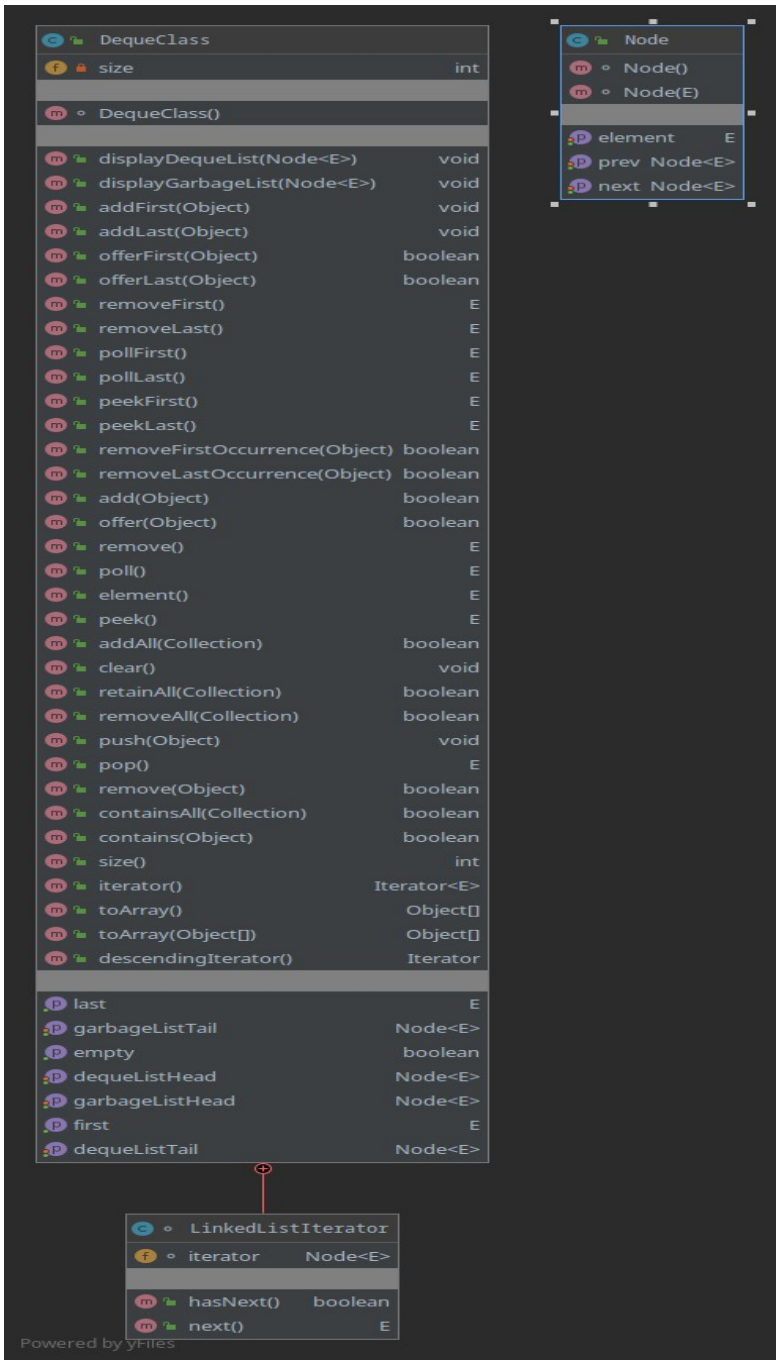
DATA STRUCTURE AND ALGORITHM
HW04

MOHAMMAD ASHRAF YAWAR
161044123

PART02

UML:


DequeClass
size int
DequeClass()
displayDequeList(Node<E>) void
displayGarbageList(Node<E>) void
addFirst(Object) void
addLast(Object) void
offerFirst(Object) boolean
offerLast(Object) boolean
removeFirst() E
removeLast() E
pollFirst() E
pollLast() E
peekFirst() E
peekLast() E
removeFirstOccurrence(Object) boolean
removeLastOccurrence(Object) boolean
add(Object) boolean
offer(Object) boolean
remove() E
poll() E
element() E
peek() E
addAll(Collection) boolean
clear() void
retainAll(Collection) boolean
removeAll(Collection) boolean
push(Object) void
pop() E
remove(Object) boolean
containsAll(Collection) boolean
contains(Object) boolean
size() int
iterator() Iterator<E>
toArray() Object[]
toArray(Object[]) Object[]
descendingIterator() Iterator
last E
garbageListTail Node<E>
empty boolean
dequeListHead Node<E>
garbageListHead Node<E>
first E
dequeListTail Node<E>
Node
Node()
Node(E)
element E
prev Node<E>
next Node<E>
LinkedListIterator
iterator Node<E>
hasNext() boolean
next() E
Powered by yFiles

PROBLEM SOLUTION APPROACH:

since we were asked to implement deque interface and overload desired methods from that interface I have implemented them according to the linkedlist that we created inside over DequeClass.

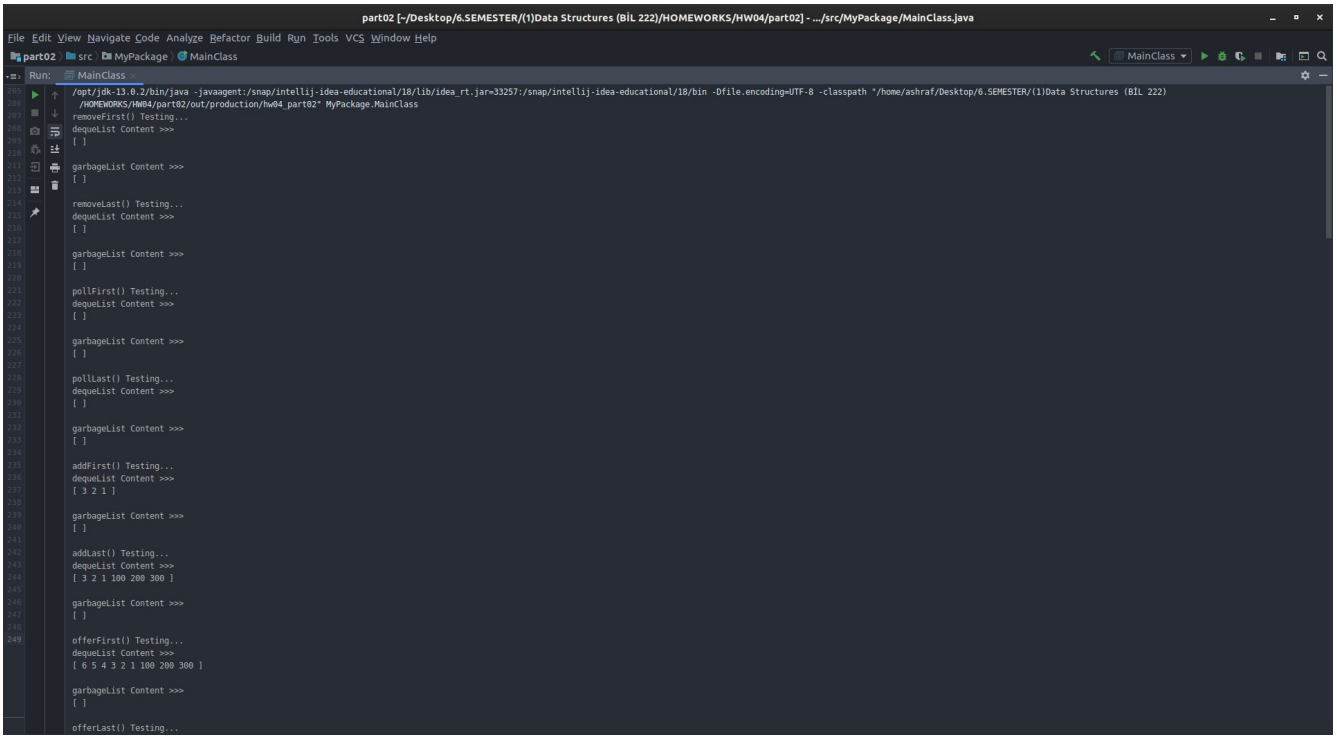
I make use of double linked list with each node having next and prevous references in order to access the head tail and add and remove in constant time.

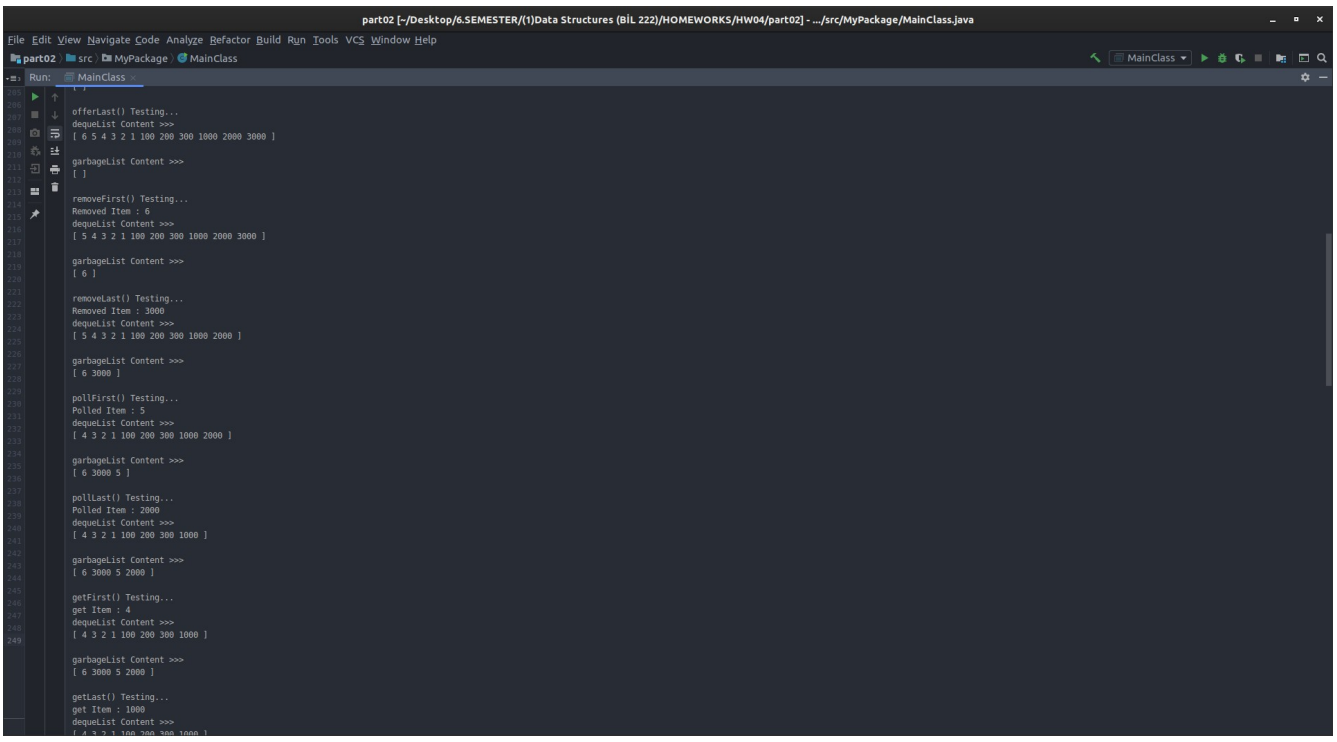
I have two linkedlist one for keeping actual data one for avoiding garbage collection and creating new node when it’s needed as soon as we have at least one node in out garbageLinkedList.

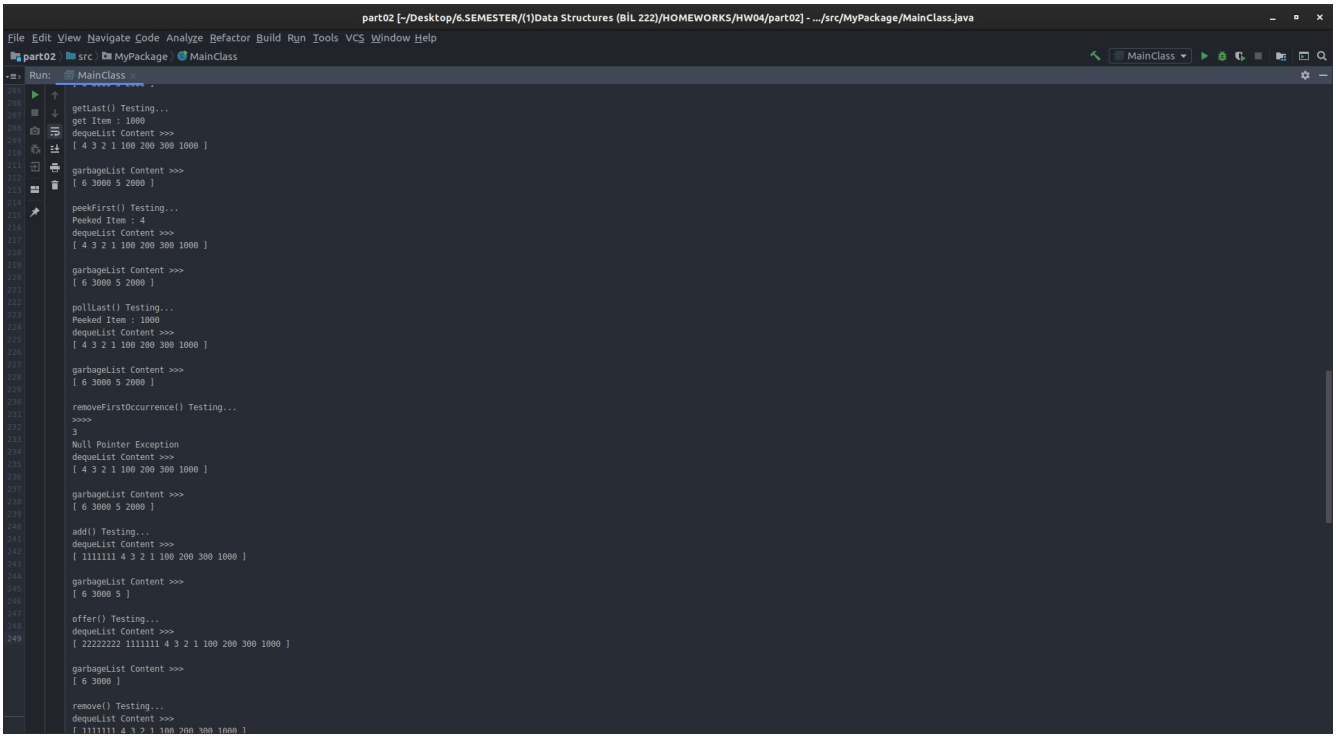
I make use of try catch mechanism in order to avoid any possible Exception and not to crash programs until it’s last input possible.

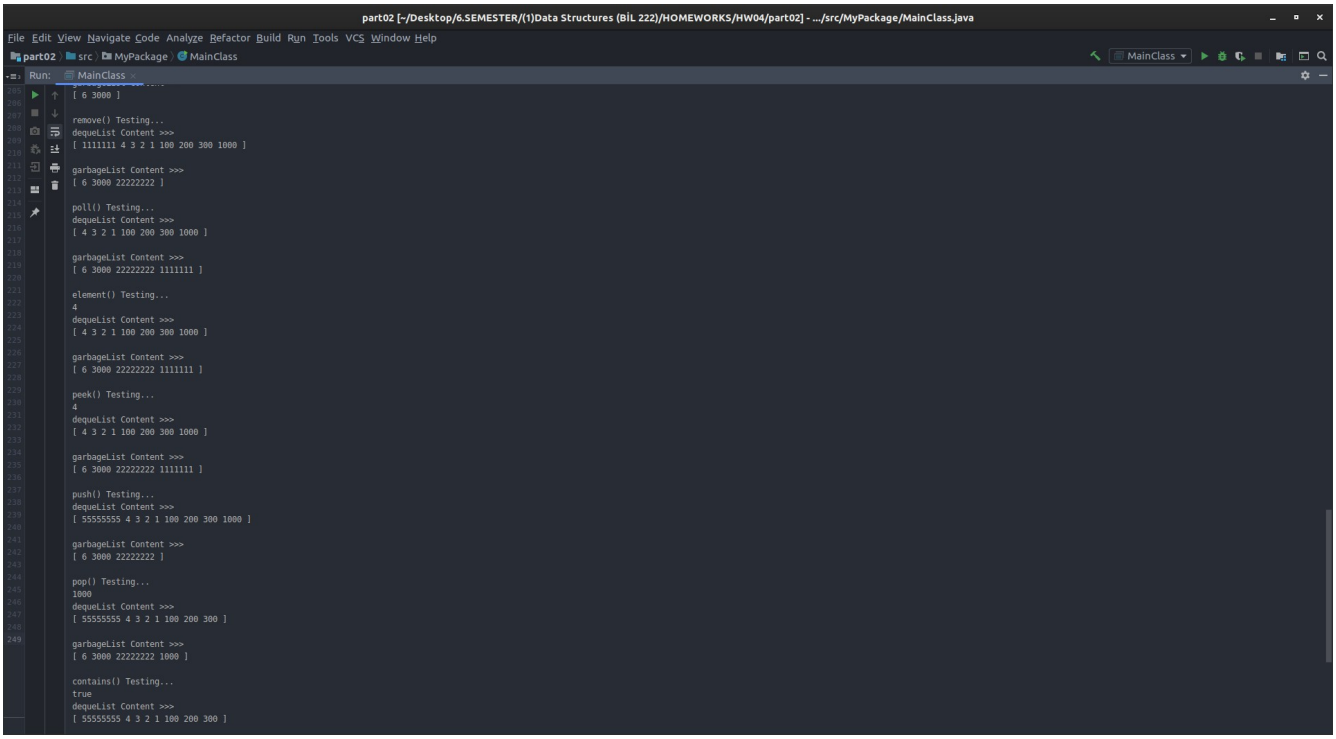
While compiling code I make use of idea called intelli j .
in iterator part I have an inner class for iterator which iterates through DequeLinkedList.

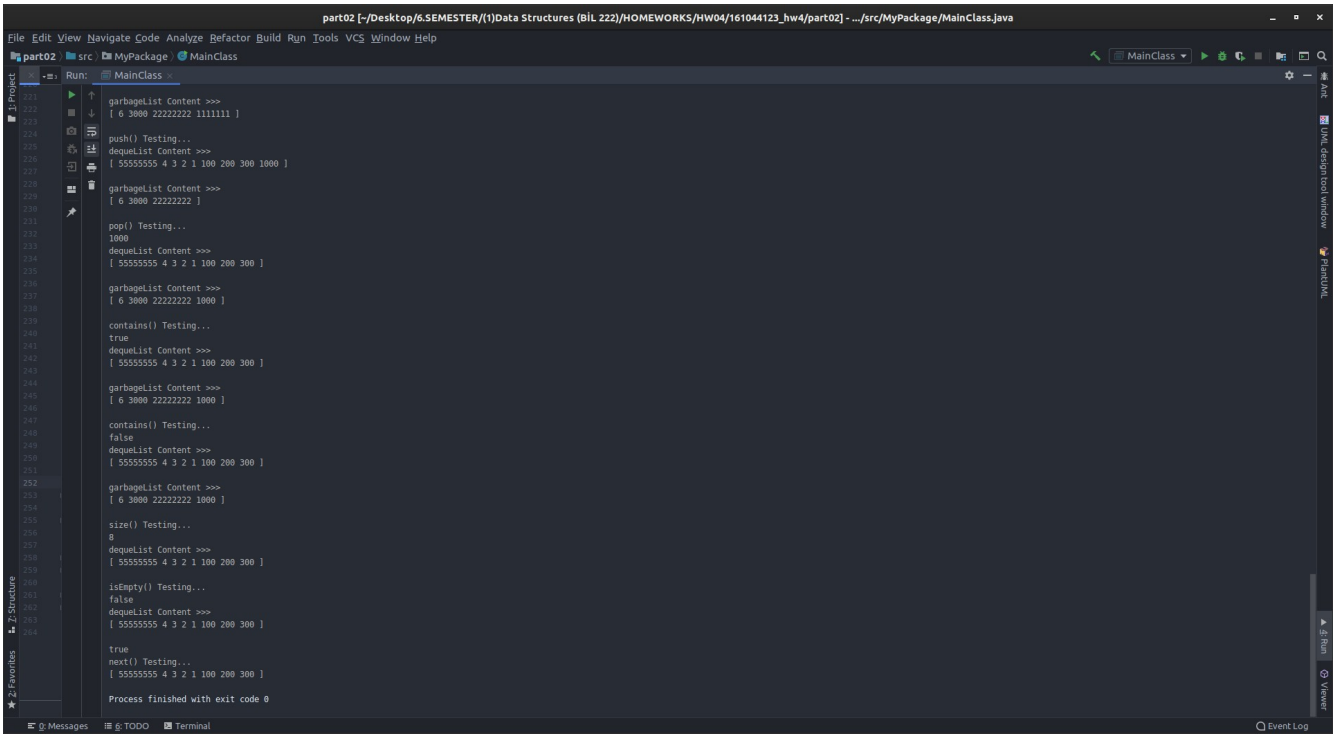
COMMAND LINE RESULTS:






part02 [~/Desktop/6.SEMESTER/(1)Data Structures (BİL 222)/HOMEWORKS/HW04/part02] - .../src/MyPackage/MainClass.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part02 src MyPackage MainClass
Run: MainClass
getLast() Testing...
get Item : 1000
dequeList Content >>>
[4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 5 2000]
peekFirst() Testing...
Peeked Item : 4
dequeList Content >>>
[4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 5 2000]
pollLast() Testing...
Peeked Item : 1000
dequeList Content >>>
[4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 5 2000]
removeFirstOccurrence() Testing...
>>>>
3
Null Pointer Exception
dequeList Content >>>
[4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 5 2000]
add() Testing...
dequeList Content >>>
[1111111 4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 5]
offer() Testing...
dequeList Content >>>
[22222222 1111111 4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000]
remove() Testing...
dequeList Content >>>
[1111111 4 3 2 1 100 200 300 1000]


part02 [~/Desktop/6.SEMESTER/(1)Data Structures (BİL 222)/HOMEWORKS/HW04/part02] - .../src/MyPackage/MainClass.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part02 src MyPackage MainClass
Run: MainClass
[6 3000]
remove() Testing...
dequeList Content >>>
[1111111 4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 22222222]
poll() Testing...
dequeList Content >>>
[4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 22222222 1111111]
element() Testing...
4
dequeList Content >>>
[4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 22222222 1111111]
peek() Testing...
4
dequeList Content >>>
[4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 22222222 1111111]
push() Testing...
dequeList Content >>>
[55555555 4 3 2 1 100 200 300 1000]
garbageList Content >>>
[6 3000 22222222]
pop() Testing...
1000
dequeList Content >>>
[55555555 4 3 2 1 100 200 300]
garbageList Content >>>
[6 3000 22222222 1000]
contains() Testing...
true
dequeList Content >>>
[55555555 4 3 2 1 100 200 300]



TEST CASES:

```
System.out.println("removeFirst() Testing...");
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//removeLast()
System.out.println("removeLast() Testing...");
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());
```

```
//pollFirst()
System.out.println("pollFirst() Testing...");
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//pollLast()
System.out.println("pollLast() Testing...");
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//addFirst()
System.out.println("addFirst() Testing...");
myObject.addFirst(1);
myObject.addFirst(2);
myObject.addFirst(3);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//addLast()
System.out.println("addLast() Testing...");
myObject.addLast(100);
myObject.addLast(200);
myObject.addLast(300);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//offerFirst()
System.out.println("offerFirst() Testing...");
myObject.offerFirst(4);
myObject.offerFirst(5);
myObject.offerFirst(6);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//offerLast()
System.out.println("offerLast() Testing...");
myObject.addLast(1000);
myObject.addLast(2000);
myObject.addLast(3000);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());
```

```
//removeFirst()
  System.out.println("removeFirst() Testing...");
  System.out.println("Removed Item : " + myObject.removeFirst());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead());
  System.out.println("garbageList Content >>> ");
  myObject.displayGarbageList(myObject.getGarbageListTail());


  //removeLast()
  System.out.println("removeLast() Testing...");
  System.out.println("Removed Item : " + myObject.removeLast());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead());
  System.out.println("garbageList Content >>> ");
  myObject.displayGarbageList(myObject.getGarbageListTail());


  //pollFirst()
  System.out.println("pollFirst() Testing...");
  System.out.println("Polled Item : " + myObject.pollFirst());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead());
  System.out.println("garbageList Content >>> ");
  myObject.displayGarbageList(myObject.getGarbageListTail());


  //pollLast()
  System.out.println("pollLast() Testing...");
  System.out.println("Polled Item : " + myObject.pollLast());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead());
  System.out.println("garbageList Content >>> ");
  myObject.displayGarbageList(myObject.getGarbageListTail());


  //getFirst()
  System.out.println("getFirst() Testing...");
  System.out.println("get Item : " + myObject.getFirst());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead());
  System.out.println("garbageList Content >>> ");
  myObject.displayGarbageList(myObject.getGarbageListTail());


  //getLast()
  System.out.println("getLast() Testing...");
  System.out.println("get Item : " + myObject.getLast());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead());
  System.out.println("garbageList Content >>> ");
  myObject.displayGarbageList(myObject.getGarbageListTail());


  //peekFirst()
  System.out.println("peekFirst() Testing...");
  System.out.println("Peeked Item : " + myObject.peekFirst());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead());
  System.out.println("garbageList Content >>> ");
```

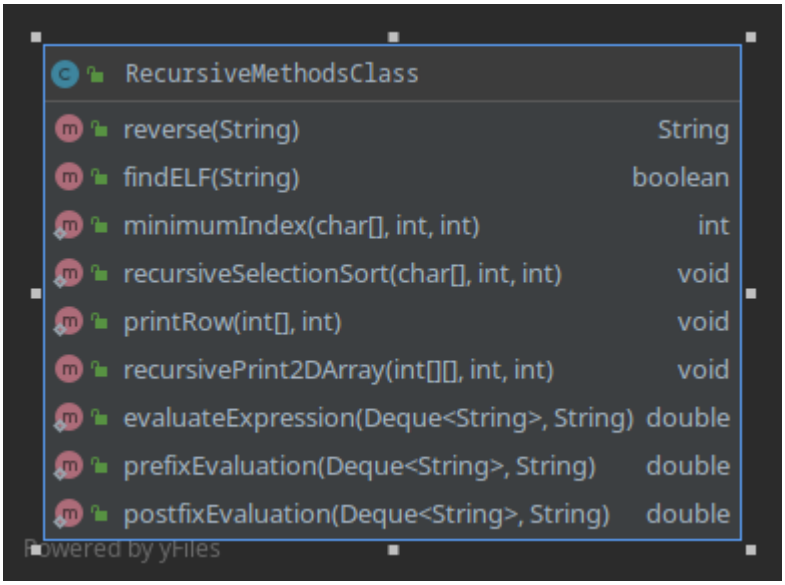
```
myObject.displayGarbageList(myObject.getGarbageListTail());


//peekLast()
System.out.println("pollLast() Testing...");
System.out.println("Peeked Item : " + myObject.peekLast());
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//removeFirstOccurrence()
System.out.println("removeFirstOccurrence() Testing...");
myObject.removeFirstOccurrence(4);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//add()
System.out.println("add() Testing...");
myObject.add(1111111);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//offer()
System.out.println("offer() Testing...");
myObject.offer(22222222);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//remove()
System.out.println("remove() Testing...");
myObject.remove();
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//poll()
System.out.println("poll() Testing...");
myObject.poll();
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//element()
System.out.println("element() Testing...");
```

```
System.out.println(myObject.element());
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//peek()
System.out.println("peek() Testing...");
System.out.println(myObject.peek());
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//push()
System.out.println("push() Testing...");
myObject.push(55555555);
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//pop()
System.out.println("pop() Testing...");
System.out.println(myObject.pop());
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//contains()
System.out.println("contains() Testing...");
System.out.println(myObject.contains(3));
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//contains()
System.out.println("contains() Testing...");
System.out.println(myObject.contains(7777777));
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
System.out.println("garbageList Content >>> ");
myObject.displayGarbageList(myObject.getGarbageListTail());


//size()
System.out.println("size() Testing...");
System.out.println(myObject.size());
System.out.println("dequeList Content >>> ");
myObject.displayDequeList(myObject.getDequeListHead());
```

```
  //isEmpty()
  System.out.println("isEmpty() Testing...");
  System.out.println(myObject.isEmpty());
  System.out.println("dequeList Content >>> ");
  myObject.displayDequeList(myObject.getDequeListHead

//hasNext();
System.out.println(iterator.hasNext());


//next();
System.out.println("next() Testing...");
System.out.print("[");
while (iterator.hasNext()){
    System.out.print(" "+iterator.next());
}
System.out.println(" ]");
```

PART03

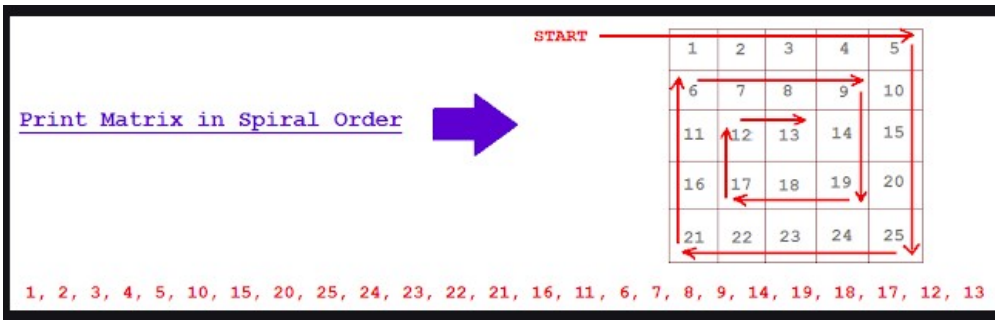
UML:



PROBLEM SOLUTION APPROACH:

>> I have used recursion in all the 6 function as it was desired
>> in prefix and postfix evaluation part I have used a common function to evaluate both type of expression. By the help of a string I indicate the type of the expression to be evaluated while calling the function `evaluateExpression().if st is “pref” then it means that we called the evaluateExpression() function to solve a prefix expression and the function will trade accordingly and same condition applies for postfix expression evaluation in which the functione valuateExpression() acts like possible evaluator since our st is “post”.`

`>> in the last part 6. part where we are spouse to print the content of a 2D arrays in spiral order using recursion I used the classic method where everywhere in internet we can access it is called` **`first row to last column → last column to fist row`** `method which is basically as follows:`

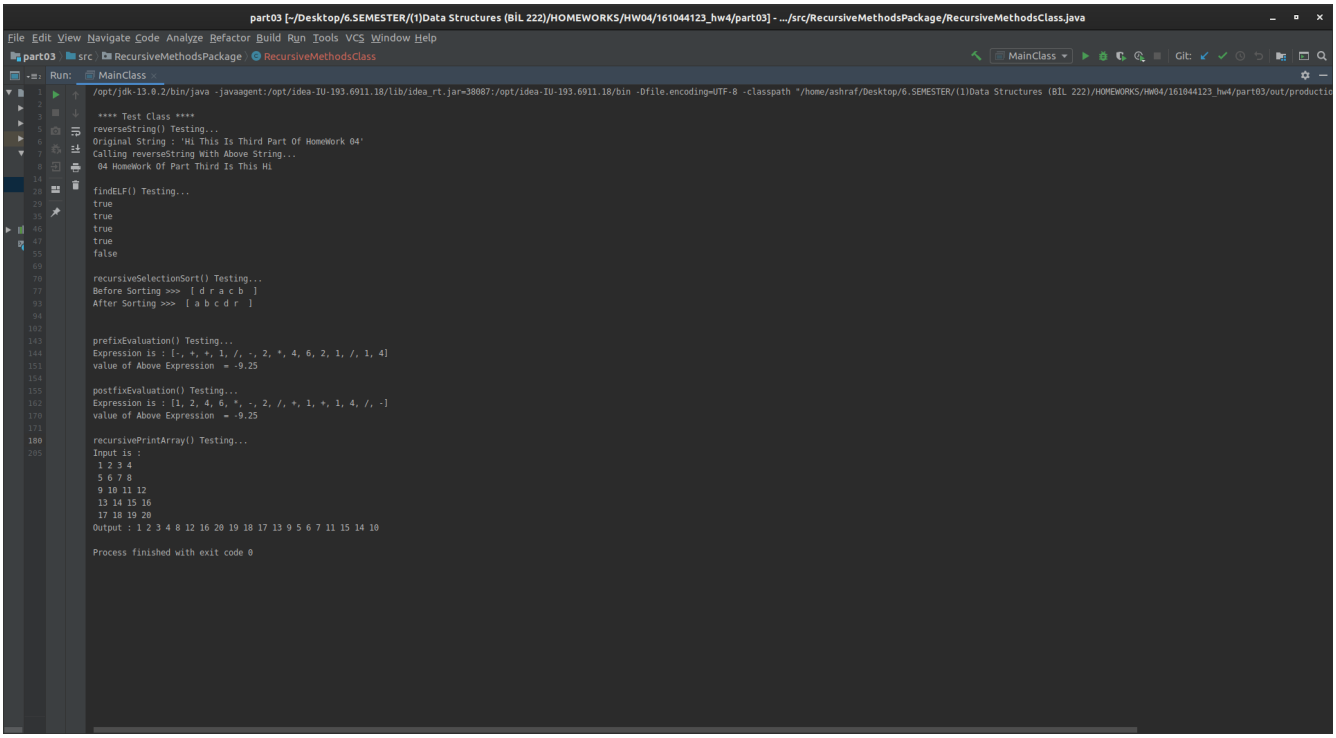


TEST CASES:

```
//reverseString()
System.out.println(recObj.findELF("whiteleaf"));
System.out.println(recObj.findELF("tasteful"));
System.out.println(recObj.findELF("unfriendly"));
System.out.println(recObj.findELF("waffles"));
System.out.println(recObj.findELF("ttt"));


//recursiveSelectionSort()
char charArray[] = new char[]{'d', 'r', 'a', 'c', 'b'};
recObj.recursiveSelectionSort(charArray,charArray.length,0);


//prefixEvaluation()
String input = "- + + 1 / - 2 * 4 6 2 1 / 1 4";
```

```
    String[] ex = input.split ( " " );
    Deque<String> lastEx = new LinkedList<String>();
    for (String element : ex) { lastEx.addLast(element); }
    System.out.println("value of Above Expression "+recObj.prefixEvaluation(lastEx,"pref"));

    //postfixEvaluation()
    String newinput = "1 2 4 6 * - 2 / + 1 + 1 4 / -";
    String[] exe = newinput.split ( " " );
    Deque<String> Ex = new LinkedList();
    for (String element : exe) { Ex.addLast(element); }
    System.out.println("value of Above Expression  = +recObj.postfixEvaluation(Ex,"post"))



    //recursivePrintArray()
    int array[][] = new int [5][4];
    array[0][0] = 1;array[0][1] = 2;array[0][2] = 3;array[0][3] = 4;
    array[1][0] = 5;array[1][1] = 6;array[1][2] = 7;array[1][3] = 8;
    array[2][0] = 9;array[2][1] = 10;array[2][2] = 11;array[2][3] = 12;
    array[3][0] = 13;array[3][1] = 14;array[3][2] = 15;array[3][3] = 16;
    array[4][0] = 17;array[4][1] = 18;array[4][2] = 19;array[4][3] = 20;

    System.out.print("Output : ");
    recObj.print2DArray(array,0,array.length,0,array[0].length);
}
```

COMMAND LINE INPUTS:


part03 [~/Desktop/6.SEMESTER/(1)Data Structures (BİL 222)/HOMEWORKS/HW04/161044123_hw4/part03] - .../src/RecursiveMethodsPackage/RecursiveMethodsClass.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part03 src RecursiveMethodsPackage RecursiveMethodsClass
MainClass
Git:
Run: MainClass
/opt/jdk-13.0.2/bin/java -javaagent:/opt/idea-IU-193.6911.18/lib/idea_rt.jar=38087:/opt/idea-IU-193.6911.18/bin -Dfile.encoding=UTF-8 -classpath "/home/ashraf/Desktop/6.SEMESTER/(1)Data Structures (BİL 222)/HOMEWORKS/HW04/161044123_hw4/part03/out/productio
**** Test Class ****
reverseString() Testing...
Original String : 'Hi This Is Third Part Of HomeWork 04'
Calling reverseString With Above String...
04 HomeWork Of Part Third Is This Hi
findELF() Testing...
true
true
true
true
false
recursiveSelectionSort() Testing...
Before Sorting >>> [d r a c b]
After Sorting >>> [a b c d r]
prefixEvaluation() Testing...
Expression is : [-, +, +, 1, /, -, 2, *, 4, 6, 2, 1, /, 1, 4]
value of Above Expression = -9.25
postfixEvaluation() Testing...
Expression is : [1, 2, 4, 6, *, -, 2, /, +, 1, +, 1, 4, /, -]
value of Above Expression = -9.25
recursivePrintArray() Testing...
Input is :
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20
Output : 1 2 3 4 8 12 16 20 19 18 17 13 9 5 6 7 11 15 14 10
Process finished with exit code 0