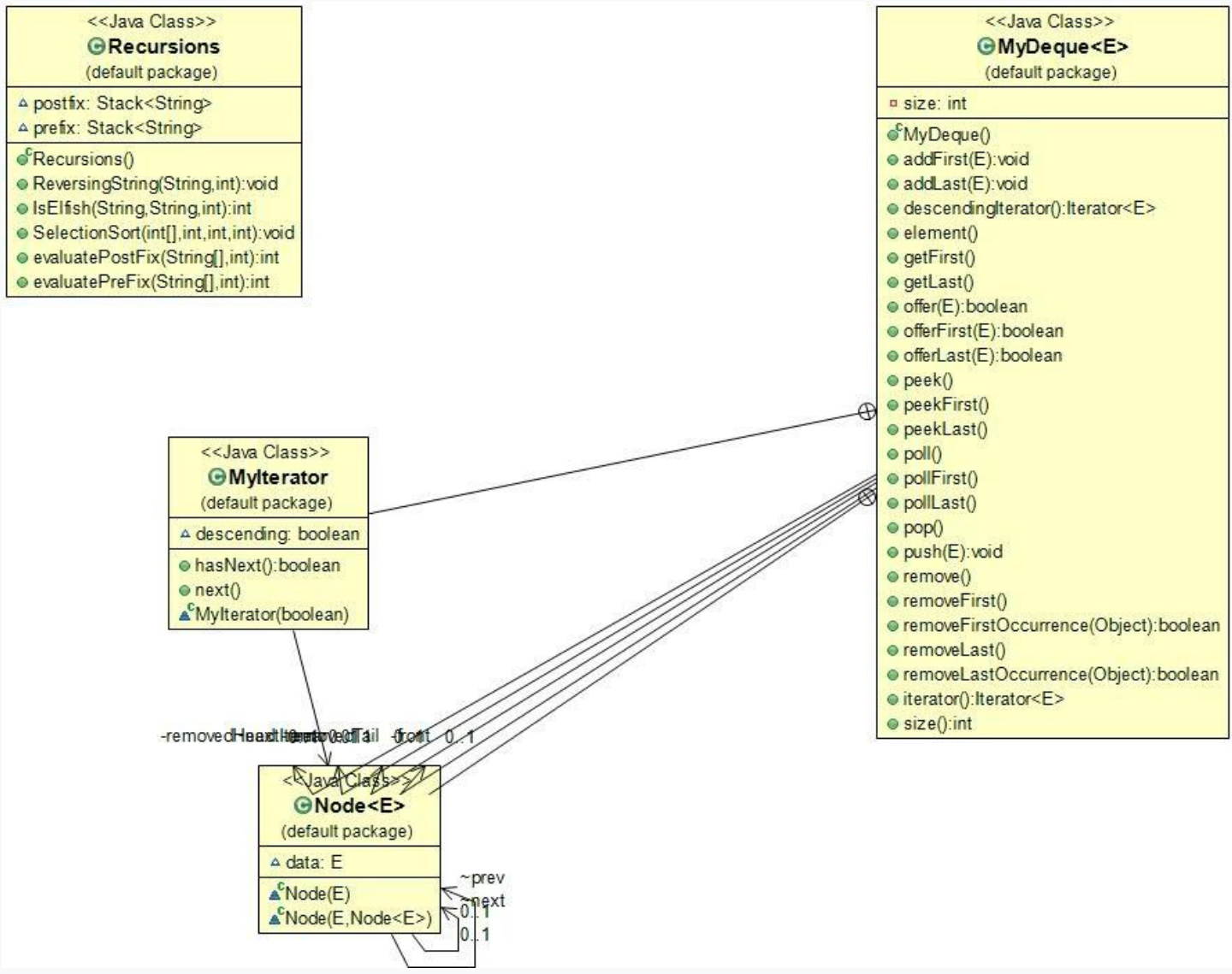
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 4 Report

161044063

ZAFER ALTAY

1-)CLASS DIAGRAM



2-)TEST CASES

For Q2:

- Node Constructor
- Iterator next method
- Iterator hasNext method
- Iterator
- Descending Iterator
- addFirst method of deque
- addLast method of deque
- element method of deque
- getFirst method of deque
- getLast method of deque
- offer method of deque
- offerLast method of deque
- offerFirst method of deque
- peek method of deque
- peekLast method of deque
- peekFirst method of deque
- pool method of deque
- poolFirst method of deque
- poolLast method of deque
- pop method of method of deque
- push method of deque
- remove method of deque
- removeFirst method of deque
- removeLast method of deque
- size method of deque

- iterator method of deque
- descendingIterator method of deque
- removeFirstOccurance method of deque
- removeLastOccurance method of deque

The expected result and output of these methods are shown in the next heading.

For Q2:

- IsElfish method
- ReversingString method
- SelectionSort method
- EvaluatePostfix method
- EvaluatePrefix method

The expected result and output of these methods are shown in the next heading.

3-) Running command and results

For Q1:

```
Which you want to test
1-)Recursions
2-)Deque
2
Now we start testing for MyDeque
We start by creating object of type MyDeque
Now we will test the addLast,add and addFirst methods.We will add 0,1,2 using them
We will check these with peekFirst,peekLast,element and getLast methods
Now we testing peekFirst method ,output must be 0,output :0
Now we testing peekLast method ,output must be 2,output :2
Now we testing element method ,output must be 0,output :0
Now we testing getFirst method ,output must be 0,output :0
Now we testing getLast method ,output must be 2,output
:2
Now we testing offer method ,we will try add 3 after that control it using peekLast
Now we testing offer method ,output must be 3,output :3

Now we testing offerLast method ,we will try add 4 after that control it using peekLast
Now we testing offer method ,output :4

Now we testing offerFirst method ,we will try add -1 and -2 after that control it using peekFisrt
Now we testing offerFirst method ,output must be -2,output :-2

Now we will test poolFirst ,poll and pop methods and control it using peekFirst,output must be 0
Now we testing pollFirst method after that control fisrt element,output :0

Now we will test poolLast method and control it using peekLast,output must be 3
Now we testing pollLast method after that control last element,output  :3

Now we will test removeFirst method and control it using peekFirst,output must be 1
Now we testing removeFirst method after that control fisrt element,output :1

Now we will test removeLast method and control it using peekLast,output must be 2
Now we testing removeLast method after that control fisrt element,output :2
```

```
Now we will test removeLast method and control it using peekLast,output must be 2
Now we testing removeLast method after that control fisrt element,output :2

We will test addFirst again but this time we will add a removed item.--->(0)
Now we test it using peekFirst output must be 0,Output :0

We will test addLast and push again but this time we will add a removed item.--->(3 and 4)
Now we test it using peekLast output must be 4,Output :0

Now we test iterator using hasnext and next methods of iterator but first we will ad some numbers(5,6,7)

Now we testing iterator.Using the iterator we write nums from the front to rear.(Output must be 0-1-2-3-4-5-6-7)
0 1 2 3 4 5 6 7 For the test we will write 3 in the last row and remove with removeLasttOccurance

Now we test descendingiterator(Output must be 3 7 6 5 4 2 1) :
3 7 6 5 4 2 1 0
Now we testing size method.Output must be 8
Output8
```

For Q3:

```
Which you want to test
1-)Recursions
2-)Deque
1
Now we will test all recursion functions
Firstly, we will test ReversingString.We send the Gebze Teknik Üniversitesi Zafer Altay as an input, we will get the reverse version of the input as output.
Altay Zafer Üniversitesi Teknik Gebze

Now we test IsElfish functtion.We will test with 2 words that is  elfish and is not.(Zafer and Waffle)
Zafer is not elfish
Waffle is elfish


Now we will test selectionSort fuction.We will send starter index,starter index+1,stater index for min index and array(8,1,3,10,0)
After that we print my sorting array output must be 0,1,3,8,10
0 1 3 8 10

Now we test evaluate postfix
We create 23 44 3 1 8 * + - +  string,my function evaluate it.(It must be return 56).Output :56

Now we test evaluate prefix.We use split because of learning index num
We create - + 8 / 6 3 2  string,my function evaluate it.(It must be return 8).Output :8
```

4-)Problem Solution Approach

For Q2:

I kept the private static Node class inside the MyDeque class because I want it to be static and private. After all, it is not accessible from the outside to make a deque.

I implemented the Iterator interface to create my own iterator.I override almost all of the methods.I kept a boolean expression to understanding if it is descending or normal according to that expression.If iterator is descendingiterator,our next method goes from rear to front,else next method goes from front to tail. My hasNext method returns true unless nextItem is null.

In addFirst method,if front is null,front and rear show the newly createed element,but if not the next of new element shows front and new element becomes front but before all of these, we are looking to see if the element to be added is among the removed elements.We tie it up if it's inside.

In addLast method,we applies the similar ways for this method but we try for rear.

In descendingIterator method,we send true to the iterator constructor.

In iterator method,we send false.

In element method,we use peek method.If peek method returns null,the element method throws excepiton.

In getFirst, we use element method.In getLast we use peekLast.If peekLast returns null,we throws exception.

In offer,offerFirst,offerLast methods,we use addLast and addFirst methods.If add methods returns null,offer methods throws exception.

In peek,peekLast and peekFirst methods,if front is not null,it returns data of front or rear.

In pool methods,we remove node and add last of removedlist.

We use poll in pop and i use offerlast in push.

In remove methods,we use poll methods.

For Q3:

In ReversingString method,we break the incoming string according to its blanks and print it backwards.Function parameters are input string and the index of scripted string array to be written.Index decreases by 1 each time unless it is equal to 0.It reach the 0 recursive function will stop because it is our base case.

In IsElfish method,We take the elf string and target input and look for the letter in the elf in turn. If the next num reaches the size of elf string(our base case), function return true.Next increases once for each correct letter.

In SelectionSort method,every time we start from the index I get, we look for the others and find the minimum. After finding the minimum, we comparision and we write the

smaller one in the index.After that index and minindex increasing 1.If index(next) reach array size function will stop.(Base case)

In evaluatePostfix method,We take the segmented string according to the spaces and add it to the stack as we get the operand.If we see an operator, we pop two elements from the stack and process the operator processing and push again.If string was end(base case),we pop the element from stack last time because it is result.We created it in class and used it to avoid creating new stacks each time.

In evaluatePrefix methods,we applied the rules in the postfix method, but we read element from the end of string.