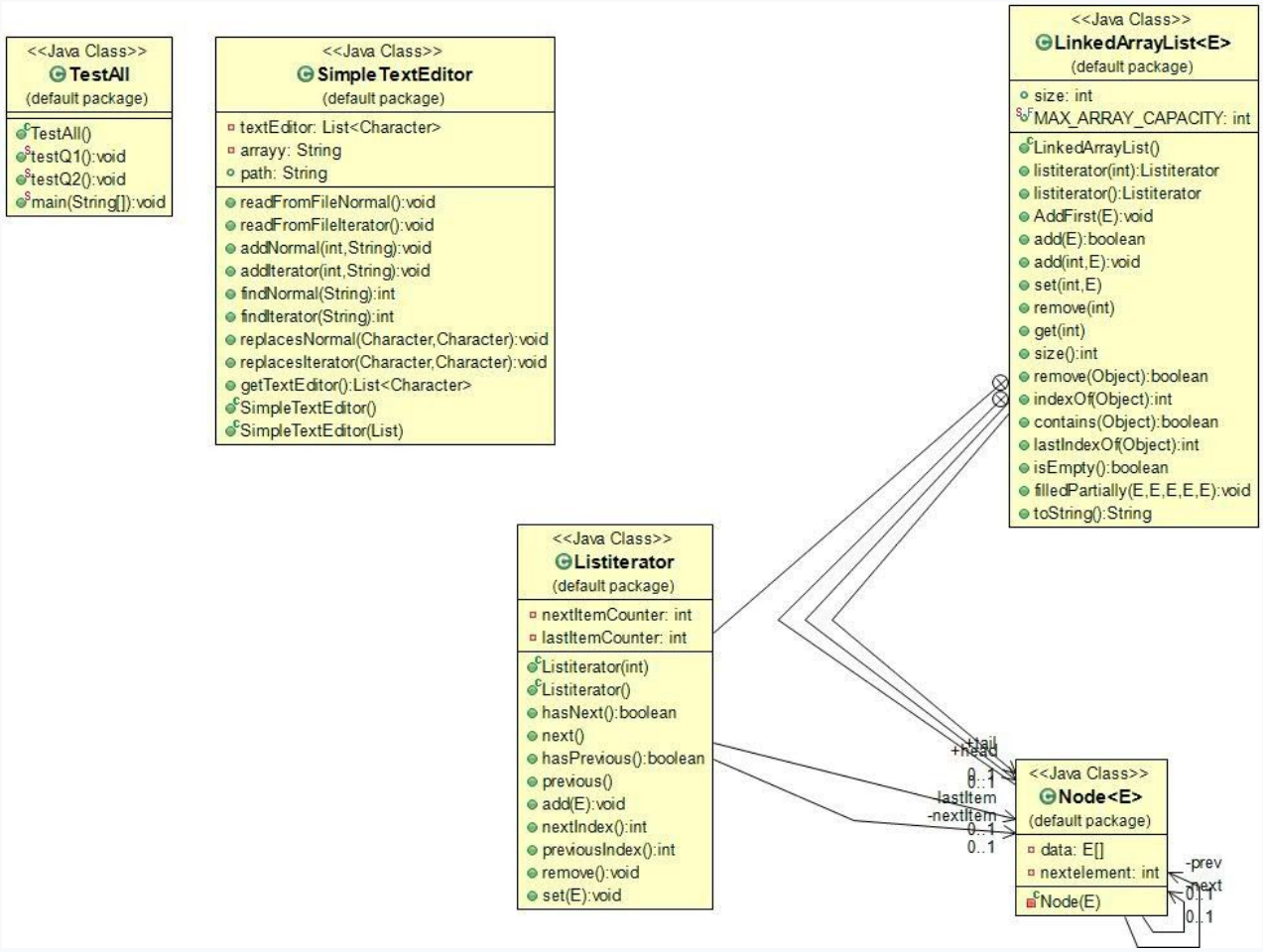
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 3 Report

161044063

ZAFER ALTAY

1-)CLASS DIAGRAM



2-)TEST CASES

For Q1:

- Create LAL Object (*LAL=LinkedArrayList)
- Filled partially array of LAL object
- Add element to end of the LAL
- Add element to anywhere in LAL
- Add element to head of the LAL
- Remove element from LAL
- Set the LAL element
- Get the LAL element
- Control the size of LLA
- Check which index the element is in LAL
- Create My list iterator
- Control index num using iterator
- Add element to LLA using iterator
- Remove element using iterator
- Check next,previous element
- Check that is iterator has next element
- Check that is iterator has previous element
- Set element using iterator

For Q2:

- Create logFile (*STE=SimpleTextEditor)
- Create STE object
- Read text from file using normal method and iterator method
- Add elements to STE object using normal method and iterator method
- Replace element of STE using normal method and iterator method
- Find text in STE using normal method and iterator method
- Measures the executions times

3-) Running command and results

For Q1:

```
Which one do you want to test?
1)LinkedArrayList
2)SimpleTestEditor
1
First we create an LinkedArrayList object that contains integer
To make PFA,we added two elements to the first node,one elemnt to second node and two elements to third node
And now we will use toString method for print the elements
0 1 4 5 6
Now we will use add(int x) method for add elements to last index.We will add two elements because of we test method that if array filled, add method create new node that is tail
0 1 4 5 6 7 8
We added and printed them
And now we test add(int,E) method.We will complete the missing index with adding 2,3 and 9
When adding to missing index, we also tested that we created a new node or shifted when the node was full.
And we will print the list
0 1 2 3 9 4 5 6 7 8
We will remove 9 using remove(int index) method and print
0 1 2 3 4 5 6 7 8
Now We will add 99 and set 99 to -1 using addFirst and  set method and print
99 0 1 2 3 4 5 6 7 8
-1 0 1 2 3 4 5 6 7 8 Now we will control the get method.We get a element that at the 5th index (it must be 4)
5 th index = 4
Now we test the size method.It must be 10
Size = 10Now we test indexOf method for true (5 must be in 6th index) and false(29 no indexed so it must return -1)
5 is in 6 th index
29 is in -1 th index
Now we check lastIndexOf method we try for 8.It mus be return 9
8 was last time occurs in 9 th index
We tested all methods of LinkedArrayList class now we will test Linkediterator class

First of all we create iterator at 4th indexes
Now we check nextIndex and previousIndex method ,first we will use next method 2 times and previous method 2 times(Output must be 4-5-5-4)
0-0-2-2
Now we will try add method of iterator so i will use nexttt method after added 99 after that I use next mehod after added 98
We check the add method results,test next and hasnext method of iterator.We will create new iterator at head of the list and we will go to the tail(Output must be -1 0 1 2 98 3 99 4 5 6 7 8)
-1 0 1 2 98 3 99 4 5 6 7 8
Now we try it for previous
8 7 6 5 4 99 3 98 2 1 0 -1
Our last element is -1 now we try remove it using remove method of iterator after that we use next meethod three times output must be 0 1 2
0 1 2
Last of all we tried set method for convert from 2 to 99 after that we will go from head to tail for check it.(Output must be 0 1 100 98 3 99 4 5 6 7 8)
0 1 100 98 3 99 4 5 6 7 8
```

For Q2:

```
Nis 02, 2020 4:02:21 PM TestAll testQ2
INFO: CSE 222 HW3 LOG FILE
Now we test all methods and measures the execution time for array list after that we test and measures them for linkedlist
Firstly, we will create an onject of arraylist
Now we check readFromFileNormal method after that we will print the text
AAAAAAAA
Now we add the string to test using addNormal method(5. index to ZAFER ALTAY) and print the new text
AAAAAZAFER ALTAYAAA
Now we find the string in text normally.We search to FER AL in text it must be return 7
FER AL starting to 7 th index
We will print the new text after replace all A character to C using replacesNormal(Output must be CCCCCZCFER CLTCYCCC)
CCCCCZCFER CLTCYCCC

We finished normal methods and we will try iterator methods for all with another txt fileNow we check readFromFileIterator method after that we will print the text
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
Now we add the string to test using addNormal method(5. index to ZAFER ALTAY) and print the new text
BBBBBZAFER ALTAYBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
Now we find the string in text normally.We search to FER AL in text it must be return 7
FER AL starting to 7 th index
We will print the new text after replace all A character to C using replacesNormal
(Output must be BBBBBZAFER ALTAYBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB)

BBBBBZAFER ALTAYBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

We finished the all test.Now we try measuring the total time for all methods that without print comment and output.Firstly we will try normal methods for exec. time
Normal methods Execution time in nanoseconds for ArrayList  : 159600
Normal methods Execution time in milliseconds for ArrayList : 0.1

Second we will try ıterator methods for exec. time
Iterator methods Execution time in nanoseconds for ArrayList  : 167200
Iterator methods Execution time in milliseconds for ArrayList : 0.1

Third we will try normal methods for exec. time
Normal methods Execution time in nanoseconds for LinkedList  : 155000
Normal methods Execution time in milliseconds for LinkedList : 0.1

Fourth we will try ıterator methods for exec. time
Iterator methods Execution time in nanoseconds for LinkedList  : 258700
Iterator methods Execution time in milliseconds for LinkedList : 0.2
```

4-)Problem Solution Approach

For Q1:

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

I implemented the ListIterator interface to create my own iterator.I override almost all of the methods because our data is a generic array, but our iterator should keep a generic single element.So for add,I added the element in the current node array and slide other parts in the array. If one of the elements exceeds the array size, I created a new node and added it to its array.I followed the same path in the remove method. If all elements of a node array were deleted, I removed that node.Although most of the other methods are similar to normal, I went through indexes as array elements and not as a node.For the Iterator, I kept not only the next node but also the next index to be able to navigate the array,I applied the same for previous.

For Q2:

I kept a text of list <Character> in STE.I cast in a constructor that bought an object of list type.I did this using polymorphism to get rid of code excess. I took a string and inserted it one by one for add using add methods of list and iterator.In the find method,If the first letter of the string is the same as the letter I checked in the text, I compared the other letters.In the replace method,I traveled in the text and compared the letters,if they match I changed them.

5-)Analysis Of The Performance

1-)List is an ArrayList and iterator is used:

For add method,when it runs for the first time for the first letter, it is O (n) for an iterator that will start from the index and O (1) for other additions and slicing is O(n).In this case, if the addition starts from the first index, the best case is o (n), worst case is o (n*n).

For find method,If the searched word starts with the first letter, the best case is O(InputString.size()).If it doesn't start with the first letter our worst case is O(n*InputString.size()).

For replace method,We don't have the best cases as we will access all the text elements.So worst case is O(n).

For read method,There is no best case because we will read as much as txt size so worst case is O(n).

2-)List is an ArrayList and iterator is not used:

For add method,If we add it to the end of the list, this would be the best case for us.O(1) for all adding but it executes InputString.size() times.If string size is 1,our best case is O(1).WorstCase is O(n*InputString.size()) because O(n) for add and executes InputString.size() times.

For find method,If the searched word starts with the first letter, the best case is O(InputString.size()).If it doesn't start with the first letter our worst case is O(n*InputString.size()).

For replace method,We don't have the best cases as we will access all the text elements.So worst case is O(n).

For read method,There is no best case because we will read as much as txt size so worst case is O(n).

3-)List is an LinkedList and iterator is used:

For add method,If we add it to the end or top of the list,and InputString have only one character,this would be the best case for us and it will be O(1).if not,O (n) is for the iterator to reach the starting index. O (1) is for each add operation. All of these are O (InputString.size()) because they will work InputString.size() times so our worst case is O(n*InputString.size()).

For find method,If the searched word starts with the first letter, the best case is O(InputString.size()).If it doesn't start with the first letter our worst case is O(n*InputString.size()).

For replace method,We don't have the best cases as we will access all the text elements.So worst case is O(n).

For read method,There is no best case because we will read as much as txt size so worst case is O(n).

4-)List is an LinkedList and iterator is not used:

For add method,If we add it to the end or top of the list, this would be the best case for us.O(1) for all adding but it executes InputString.size() times.If string size is 1,our best case is O(1).WorstCase is O(n*InputString.size()) because O(n) for add and executes InputString.size() times.

For find method,If the searched word starts with the first letter, the best case is O(InputString.size()).If it doesn't start with the first letter our worst case is O(n*InputString.size()).

For replace method,We don't have the best cases as we will access all the text elements.So worst case is O(n).

For read method,There is no best case because we will read as much as txt size so worst case is O(n).

6-)Comparision Of Times

According to the measurement results:

LinkedListIterator methods<ArrayListNormal methods<ArrayListIterator methods<LinkedListNormal methods

According to the measurement results is LinkedListIterator the fastets.