Gebze Technical University
Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 2 Report

Alper YAŞAR
151044072

1 - Q1

1. SYSTEM REQUIREMENTS

Class should keep a linked list where each node contains an array of elements with constant capacity. These arrays should be partially filled arrays. Arrays in the nodes of the linked list, may contain different number of elements.

1. Add(E item)
   - Add generic item to linked list. First check size control. If size equal zero;

     (a) addFirst(item)
     - add first element to head array and increase size
     - Every array has MAX_SIZE equal to 5 and if it is full add new node with addAfter link list and increase size
   - If size bigger than zero

     (b) addAfter(item)
     - assign head to iter and go to the last node (empty node) after that assign element and increase size

2. Remove(index)
   - Check index is empty or not empty with get(index)

     a) get(index)
     - check data is exist or not exist

     If index less than 5 ;

     a) removeFirst(index)
     - Remove specific index in first array. If array size is equal to zero than remove first node and assign if has next node to head node.

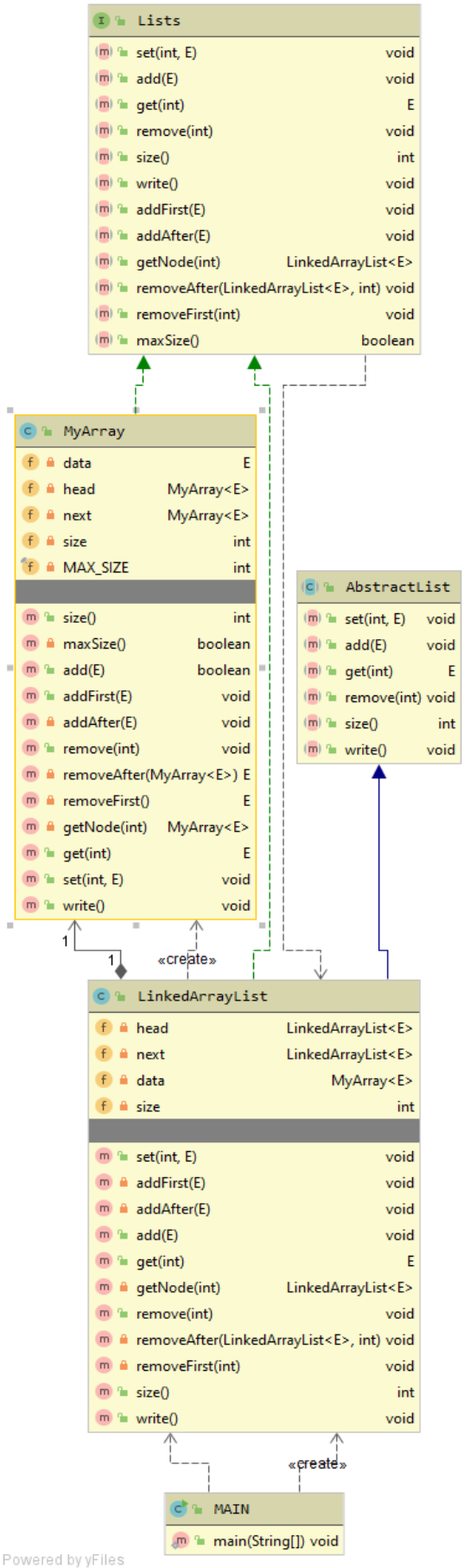
     b) removeAfter(index)
     - check where index which array and get node and remove index in array. If array is empty remove this node.

3. Set(index,newItem)

   - Found where index with getNode and change data with ne

4. Size()
    - Return size
5. write()
    - write all element in link list and in array.

3. CLASS DIAGRAM



4. PROBLEM SOLUTION APPROACH

Sending new elements and change and remove some elements. After that write which element in linked list.

5. TEST CASES

```
LinkedArrayList<Integer> temp = new LinkedArrayList<~>();

temp.add(1);          temp.add(2);
temp.add(3);
temp.remove( index: 1);
temp.add(4);         temp.add(5);
temp.add(11);         temp.add(12);
temp.add(13);         temp.add(14);
temp.add(15);         temp.add(21);
temp.add(22);         temp.add(23);
temp.add(24);         temp.add(35);
temp.add(31);         temp.add(32);
temp.add(33);         temp.add(34);
temp.add(35);
temp.remove( index: 0);

temp.remove( index: 1);

temp.add(123);
temp.set(0,227);        temp.set(6,237);
temp.set(17,247);


temp.write();
```

6. RUNNING AND RESULTS

```
227
5
11
237
13
14
15
21
22
23
24
35
31
32
247
34
35
123
```

1 – Q2

1. SYSTEM REQUIREMENTS

Class should keep a linked list where each node contains an array of elements with constant capacity. These arrays should be partially filled arrays. Arrays in the nodes of the linked list, may contain different number of elements.

1. ReadFile()
   - r read in a text file and construct the text. Open file and read char by char.

2. Add(item,index)
   - adds one or more characters (given as a string) at the specified position (given as an integer index) in text.
3. Search(letter)
   a. returns the start index of the first occurrence of the searched group of characters.

4. replace(newChar,oldchar)
   - replaces all occurrences of a character with another character. Write two implementations:

2. Measure

Measures the running time of your implementation for various text sizes.

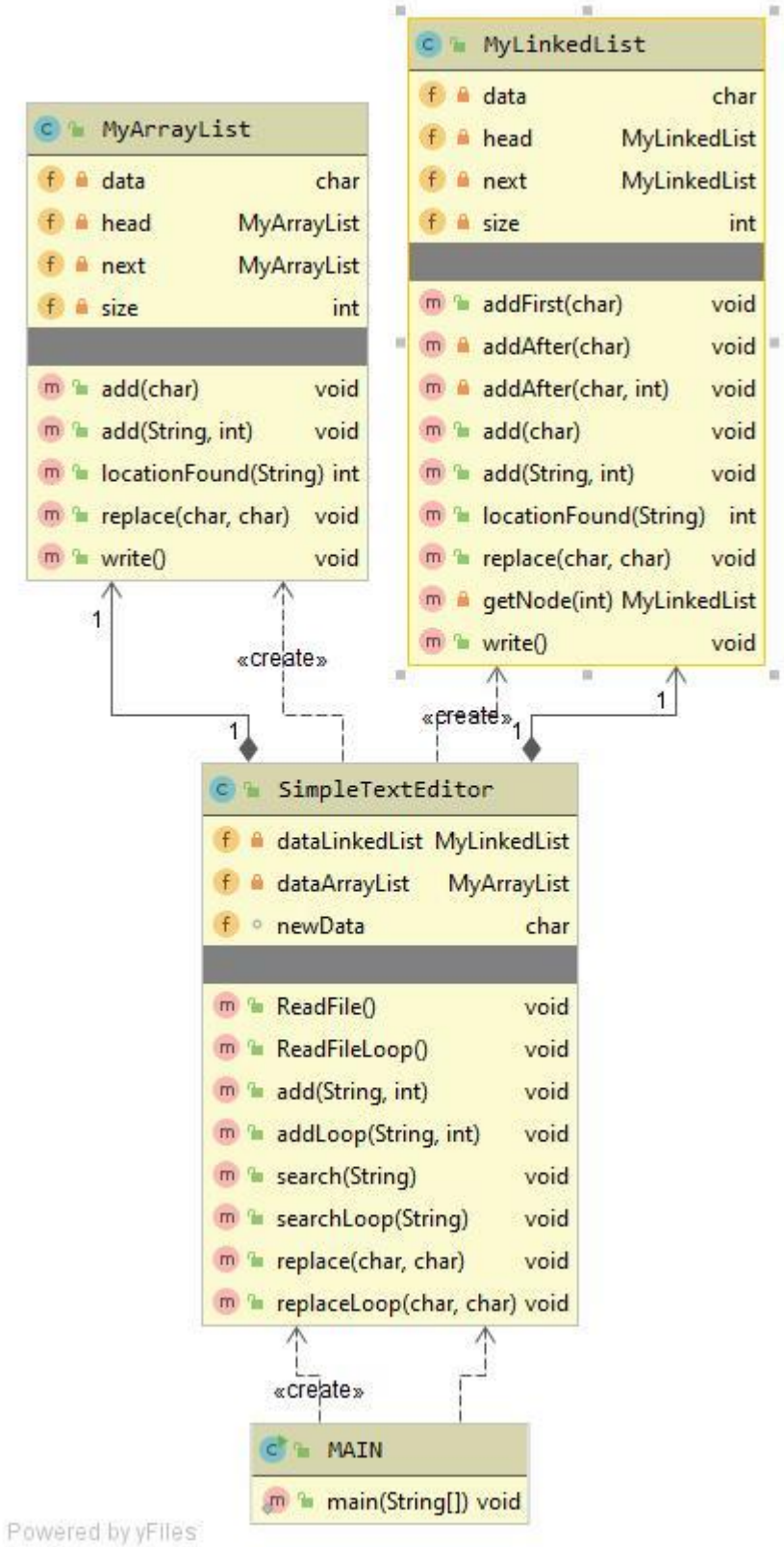
- **asymptotic notation**
- ◦ List is an ArrayList ane iterator is usee
- ▪ read : reading text size = n worst case : O(n), best case : Ω (n) = Q(n)
- ▪ add : add method in iterator use the add(index, object) method so loop is depend on index if index = n => worst case : O(n) , best case : Ω (n) = Q(n)
- ▪ search : wore size = n => worst case : (n), best case : Ω (n) = Q(n)
- ▪ replace : O(1)

- List is an ArrayList and iterator is not used
- ▪ read: reading text size = n worst case : O(n), best case : Ω (n) = Q(n)
- ▪ add: wore size = n , index = m worst case : O(n.m) , best case : Ω (n.m) = Q(n.m)
- ▪ search: list size = n , wore size = m worst case : O(m.n), best case : Ω (n.m) = Q(n.m)
- ▪ replace : text size = n worst case : (n), best case : Ω (n) = Q(n)

◦ List is an ArrayList and iterator is not used
▪ read : readText Size : 97 bytes --- readLoop time : 2ms readText Size : 368 bytes --- readLoop time : 6ms
▪ add : with read method. readText Size : 368 bytes --- aaddLoop time : 3ms readTeit Size : 97 bytes --- addLoop time : 4ms
▪ search : with read method readText Size :97 bytes --- SearchLoop time : 3ms readText Size :368 bytes --- SearchLoop time : 3ms
▪ replace : with read method readText Size : 97 bytes --- replaceLoop time : 3ms readText Size : 368 bytes --- replaceLoop time : 6ms

◦ List is a LinkedList and iterator is used
▪ read: text size : 368 bytes --- readIterator time : 16ms text size : 97 bytes --- readIterator time : 7ms
▪ add: with read method. Text size : 97 bytes --- readIterator time : 5ms ◦ Text size : 368 bytes --- readIterator time : 5ms
▪ search: with read method Text size : 97 bytes --- fneIterator time : 6ms Text size : 368 bytes --- SearchLoop time : 10ms
▪ replace : with read method Text size : 97 bytes --- replaceIterator time : 4ms Teit size : 368 bytes --- replaceIterator time : 6ms

3. CLASS DIAGRAM



4. PROBLEM SOLUTION APPROACH

Sending new elements and change and remove some elements. After that write which element in linked list.

5. TEST CASES

```
SimpleTextEditor deneme = new SimpleTextEditor();
deneme.ReadFile();
deneme.add( item: "alper", index: 0);
deneme.add( item: " yasar" ,  index: 5);
deneme.search( letter: "became");
deneme.replace( newChar: '^', oldChar: ' ');


deneme.ReadFileLoop();
deneme.addLoop( item: "Zeki",  index: 12);
deneme.addLoop( item: "TEKIRDAG",  index: 110);
deneme.searchLoop( letter: "became");
deneme.replace( newChar: '^', oldChar: ' ');
```

,

7. RUNNING AND RESULTS

```
'became' is found in 530
alper^yasarSigmund^Freud^^German:^born^Sigismund^Schlomo^Freud;^6^May^1856^–^23^September^1939)^was^an^Austrian^neurologist
and^the^founder^of^psychoanalysis,^a^clinical^method^for^treating^psychopathology^through^dialogue^between^a
patient^and^a^psychoanalyst.
Freud^was^born^to^Galician^Jewish^parents^in^the^Moravian^town^of^Freiberg,^in^the^Austrian^Empire.
He^qualified^as^a^doctor^of^medicine^in^1881^at^the^University^of^Vienna.Upon^completing^his^habilitation^in^1885,
he^was^appointed^a^docent^in^neuropathology^and^became^an^affiliated^professor^in^1902.^Freud^lived^and^worked^in^Vienna,
having^set^up^his^clinical^practice^there^in^1886.^In^1938,^Freud^left^Austria^to^escape^the^Nazis.
He^died^in^exile^in^the^United^Kingdom^in^1939
--------------------------------------------------------------------------------


'became' is found in 530
alper^yasarSigmund^Freud^^German:^born^Sigismund^Schlomo^Freud;^6^May^1856^–^23^September^1939)^was^an^Austrian^neurologist
and^the^founder^of^psychoanalysis,^a^clinical^method^for^treating^psychopathology^through^dialogue^between^a
patient^and^a^psychoanalyst.
Freud^was^born^to^Galician^Jewish^parents^in^the^Moravian^town^of^Freiberg,^in^the^Austrian^Empire.
He^qualified^as^a^doctor^of^medicine^in^1881^at^the^University^of^Vienna.Upon^completing^his^habilitation^in^1885,
he^was^appointed^a^docent^in^neuropathology^and^became^an^affiliated^professor^in^1902.^Freud^lived^and^worked^in^Vienna,
having^set^up^his^clinical^practice^there^in^1886.^In^1938,^Freud^left^Austria^to^escape^the^Nazis.
He^died^in^exile^in^the^United^Kingdom^in^1939
--------------------------------------------------------------------------------
```

1 – Q2

1. SYSTEM REQUIREMENTS

Each node keeps a character and references to previous and next nodes. To link the words, each node may have a reference to a cross node, as well. Cross node reference for the character ‘U’ in the word “PUZZLES” is the reference of the node containing the character ‘U’ in the word “FUN” ReadFile()

- r read in a text file and construct the text. Open file and read char by char.

1. Add(index,word)

- Connect 2 words to each other.

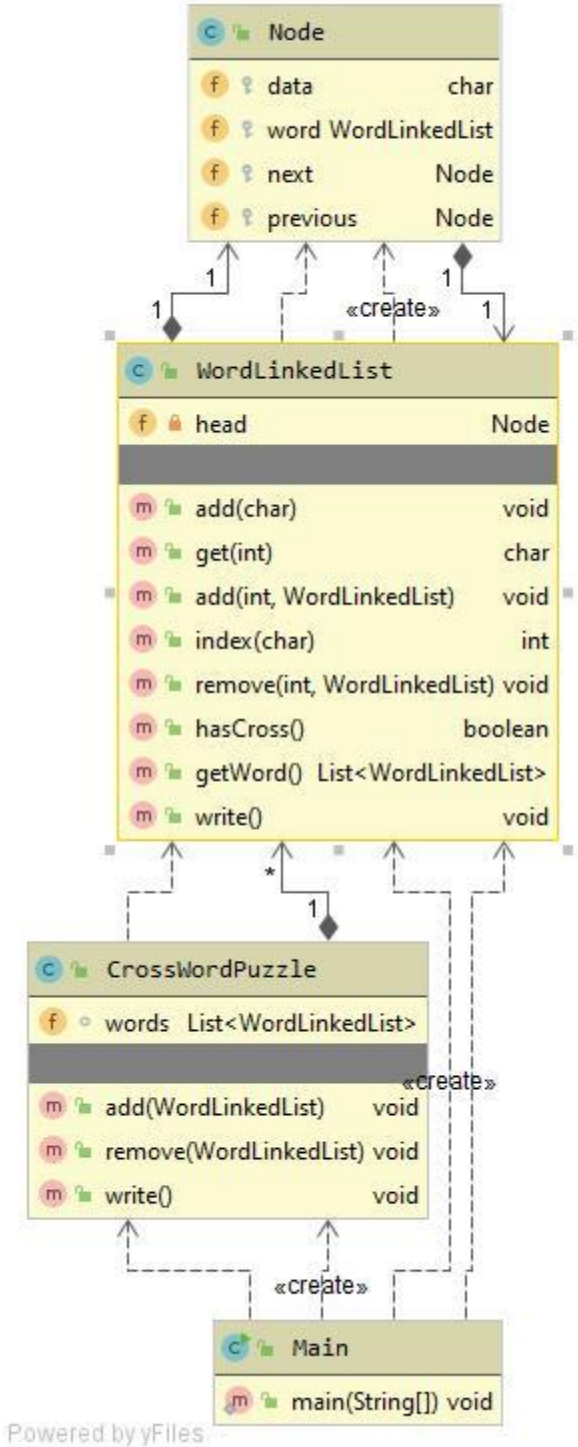
2. crossWord(word)

- cross method that adds a cross between itself and another word at specified indexes. Note that the other word should have the corresponding cross, as well.

3. Write()

- prints the word, its cross words and cross indexes

2. UML DİAGRAM



4. TEST CASES

```
WordLinkedList puz = new WordLinkedList("PUZZLES");
WordLinkedList fun = new WordLinkedList("FUN");
WordLinkedList cw = new WordLinkedList("CROSSWORD");
WordLinkedList are = new WordLinkedList("ARE");

WordLinkedList tek = new WordLinkedList("TEKIRDAG");
WordLinkedList ist = new WordLinkedList("ISTANBUL");
WordLinkedList koe = new WordLinkedList("KOCAELI");
WordLinkedList krk = new WordLinkedList("KIRKLARELİ");

puz.add( index: 1,fun);        puz.add( index: 6,cw);
cw.add( index: 1,are);         cw.add( index: 7,tek);
tek.add( index: 3,ist);        ist.add( index: 3,koe);
koe.add( index: 0,krk);

CrossWordPuzzle crossWords = new CrossWordPuzzle();
crossWords.add(puz);
                               crossWords.write();

System.out.println("After remove :");
  crossWords.remove(are);

crossWords.write();
```

5. Result

```
'PUZZLES' was crossed with 'FUN' in 2. index, 'CROSSWORD' in 7. index.

'FUN' was crossed with 'PUZZLES' in 2. index.

'CROSSWORD' was crossed with 'ARE' in 2. index, 'PUZZLES' in 4. index, 'TEKIRDAG' in

'ARE' was crossed with 'CROSSWORD' in 2. index.

'TEKIRDAG' was crossed with 'ISTANBUL' in 4. index, 'CROSSWORD' in 5. index.

'ISTANBUL' was crossed with 'TEKIRDAG' in 1. index, 'KOCAELI' in 4. index.

'KOCAELI' was crossed with 'KIRKLARELİ' in 1. index, 'ISTANBUL' in 4. index.

'KIRKLARELİ' was crossed with 'KOCAELI' in 1. index.
```