GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 3
Due date: 31 March 2020 – 23:55

SCENARIO:

Q1:

Implement a LinkedArrayList class which implements List interface and extends AbstractList class.

Your 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.

When you remove an element in the list, you should find the array containing the elements, first. During remove operation, you should shift the elements (coming after the removed element) in that array only. The other arrays should not be affected. Of course, if the number of elements in an array becomes zero, the node containing this array should also be removed from the linked list.

When you add a new element, you need to add the elements into the corresponding array. If the capacity of the array is exhausted, you should create a new node (containing an empty array) in the linked list instead of incrementing the size of the array (you are not allowed to reallocate). Of course, you may want to move some of the elements in the exhausted array into the new array.

You are not allowed to use any class in Collection hierarchy as a member. So, you need to define your own Node class to build a linked list and use basic array structure in Java. Note that you need to implement all required methods and ListIterator class.

Write a Main class to test each method in your class.

Q2:

Implement a SimpleTextEditor class which provides the some of the edit functionality of a text editor.

The text is represented by a list of characters in your class. Your class should include a Java List. Your class should provide the following methods:

- Read method to read in a text file and construct the text.

- Add method that adds one or more characters (given as a string) at the specified position (given as an integer index) in your text.
- Find method that returns the start index of the first occurrence of the searched group of characters.
- Replace method that replaces all occurrences of a character with another character. Write two implementations:

You have to write two versions of each method:

- by using the iterator (ListIterator) to navigate on the List
- without using the iterator, using simple for loop with index structure

So, there will be eight methods in total.

Write a Main class which:

- Tests each method of the SimpleTextEditor class.
- Measures the running time of your implementation for various text sizes.
- Creates a readable log file of the test.

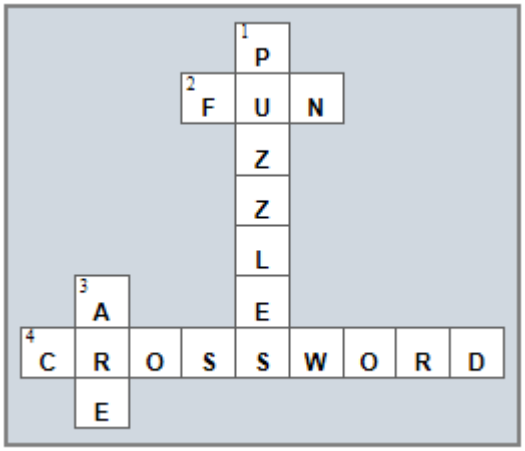
Your should write a report for this question including the following

- Analysis of the performance of each method theoretically using the most appropriate asymptotic notation. Present the analysis for the following cases:
    - List is an ArrayList and iterator is used
    - List is an ArrayList and iterator is not used
    - List is a LinkedList and iterator is used
    - List is a LinkedList and iterator is not used
- Comparison of the experimental performance of each operation when
    - List is an ArrayList and iterator is used
    - List is an ArrayList and iterator is not used
    - List is a LinkedList and iterator is used
    - List is a LinkedList and iterator is not used

Q3:

(Optional for extra credit) Implement a WordLinkedList class to handle words of a crossword puzzle.

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”



Your class will include following methods:

- The constructor which gets a string and constructs itself as WordLinkedList.
- Add 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.
- Remove cross method that removes a cross at specified index.
- Print method that prints the word, its cross words and cross indexes

Implement a CrossWordPuzzle class. Your class will have a list of WordLinkedList and following methods:

- Add word method that adds a word to the puzzle and its crosses if specified
- Remove word method that removes a word from the puzzle and its crosses
- Print method that prints each word in the puzzle with its cross words and indexes

Write a Main class where the user can add/remove words and print the puzzle words with their cross words and indexes

RESTRICTIONS:

- Use only specified data types
- Can be only one main class in each question
- Don't use any other third part library

GENERAL RULES:

- For any question firstly use course news forum in Moodle, and then the contact TA.
- You can submit assignment one day late and will be evaluated over twenty percent (%40).

TECHNICAL RULES:

- Use given CSE222-VM to develop and test your Homeworks (your code must be working on CSE222-VM), CSE222-VM download link will be given on Moodle.
- Implement clean code standards in your code;
    - Classes, methods and variables names must be meaningful and related with the functionality.
    - Your functions and classes must be simple, general, reusable and focus on one topic.
    - Use standard java code name conventions.

REPORT RULES:

- Add all javadoc documentations for classes, methods, variables …etc. All explanation must be meaningful and understandable.
- You should submit your homework code, Javadoc and report to Moodle in a "studentid_hw3.tar.gz" file.
- Use the given homework format including selected parts from the table below:

Detailed system requirements	
Use case diagrams (extra points)	
Class diagrams	X
Other diagrams	
Problem solutions approach	X
Test cases	X
Running command and results	X

GRADING :

- **No OOP design: -100**
- **No interface: -95**
- **No method overriding: -95**
- **No error handling: -50**
- **No inheritance: -95**
- **No polymorphism: -95**
- No javadoc documentation: -50
- No report: -90
- Disobey restrictions: -100
- **Cheating : -200**
- Your solution is evaluated over 100 as your performance.

CONTACT :

- Teaching Assistant : Nur Banu Albayrak