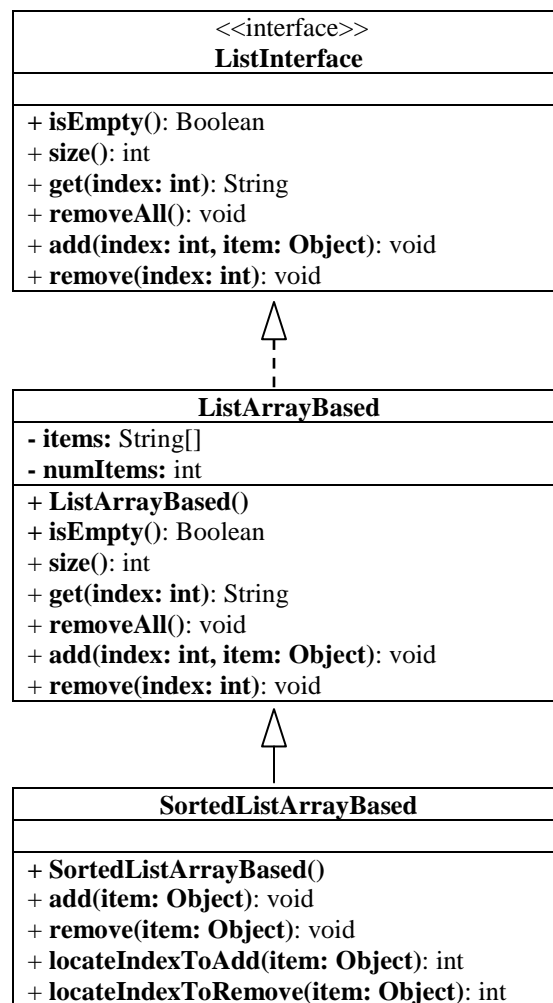## SortedListArrayBased

Topics

- Throughout this lab, we want to practice abstract data type (ADT).

- In particular, this lab asks you to implement a SortedListArrayBased class, which extends ListArrayBased class that implements ListInterface.

- NOTE: After we learn Ch.5, you will be asked to implement another SortedListReferenceBased class, which will extends "ListReferenceBased" class that implements "ListInterface". To be more specific, it will be reference-based (i.e., singly-linked-list-based).

Activities

| <<interface>> |
| **ListInterface** |
| |
| + **isEmpty**(): Boolean
+ **size**(): int
+ **get(index: int)**: String
+ **removeAll**(): void
+ **add(index: int, item: Object)**: void
+ **remove(index: int)**: void |

△

| **ListArrayBased** |
| - **items:** String[]
- **numItems:** int |
| + **ListArrayBased**()
+ **isEmpty**(): Boolean
+ **size**(): int
+ **get(index: int)**: String
+ **removeAll**(): void
+ **add(index: int, item: Object)**: void
+ **remove(index: int)**: void |

△

| **SortedListArrayBased** |
| |
| + **SortedListArrayBased**()
+ **add(item: Object)**: void
+ **remove(item: Object)**: void
+ **locateIndexToAdd(item: Object)**: int
+ **locateIndexToRemove(item: Object)**: int |

- Start with the following supportive files ("ListException.java", "ListIndexOutOfBoundsException.java", "ListInterface.java", "ListArrayBased.java", "SortedListArrayBasedDriver.java", and "SortedListArrayBased.java).

- The first five java files are complete, thus use them as they are provided. You're asked to show your work on the java file, "**SortedListArrayBased.java**".

- Remember that in Chapter 4, "ListArrayBased.java" implements "ListInterface". In this lab, "SortedListArrayBased" extends "ListArrayBased.

- In class, we discussed an example of interface on shopping list (refer to the textbook and the lecture note). Required implementation description/requirement of each method is given in "SortedListArrayBased.java".

- Note that you should keep the sorted (say, non-decreasing) order of items in the array, while adding an item using "add()". Also, make sure you make shift left (or right) properly upon inserting (i.e., adding) (or deleting (i.e., removing)) an item to (or from) the array. Keeping the sorted order in the array is the key in this lab.

- Note that index starts from 0, not 1. Therefore, the valid index ranges from 0 to size()-1.

Running Example

0. apples 1. black beans 2. bread 3. butter 4. chicken 5. eggs 6. flour 7. milk 8. pecans 9. rice 10. sausage numItems is now: 11 Adding juice … item 5 is: eggs 0. apples 1. black beans 2. bread 3. butter 4. chicken 5. eggs 6. flour 7. juice 8. milk 9. pecans 10. rice 11. sausage numItems is now: 12	Removing juice… 0. apples 1. black beans 2. bread 3. butter 4. chicken 5. eggs 6. flour 7. milk 8. pecans 9. rice 10. sausage numItems is now: 11 Adding banana … 0. apples 1. banana 2. black beans 3. bread 4. butter 5. chicken 6. eggs 7. flour 8. milk 9. pecans 10. rice 11. sausage numItems is now: 12	Removing sausage… 0. apples 1. banana 2. black beans 3. bread 4. butter 5. chicken 6. eggs 7. flour 8. milk 9. pecans 10. rice numItems is now: 11 Removing cheese… 0. apples 1. banana 2. black beans 3. bread 4. butter 5. chicken 6. eggs 7. flour 8. milk 9. pecans 10. rice numItems is now: 11 Press any key to continue …

What to Hand in

- Turn in your program, "**SortedListArrayBased.java**", via Blackboard.