Grocery List Program Requirements

Topics: One dimensional arrays, array operations, sorting, searching, loops and selection, Fail safe programming

In this assignment you will write a program that allows a person to create and manipulate a shopping list. At any given time the list may be partially filled, and your program must keep track of how many values are ACTUALLY in the grocery list, versus the size of your array (you must keep track of the current # of values in the array). **It is essential that you practice fail safe programming.** Your program will display a menu that will allow the user to perform the following operations REPEATEDLY until they select quit.

1. **CREATE A NEW EMPTY LIST:** This operation will prompt the user for the maximum number of items the list can hold, and your program will create an array of the specified size. The array will be of type String. **NO OTHER OPERATIONS ARE VALID UNTIL A LIST IS CREATED.** If this option is selected after initially creating a list, it will create a new list overwriting the old list.

2. **ADD A NEW ITEM TO THE END OF THE LIST:** This option will prompt the user for the item they wish to add to the list, and will add it at the end of the existing list.

3. **ADD A NEW ITEM AT A SPECIFIC POSITION IN THE LIST:** This option will add a new value to the list at a specific position specified by the user. This option will prompt the user for the item they wish to add, and the position at which the item is to be inserted.

4. **DELETE AN ITEM FROM THE END OF THE LIST:** This option when selected will delete the value at the end of the list, setting the CORRECT element to **null.**

5. **DELETE AN ITEM FROM A SPECIFIC POSITION IN THE LIST:** This option will remove an item at a specified position in the list. The position given by the user must be valid. Before deleting the value the operation should prompt the user for verification, displaying the value to be deleted. If the user enters yes, this operation will delete the value, otherwise do not delete it.

6. **DISPLAY ELEMENT:** This option will prompt the user for the index of the item to display. This index must be valid.

7. **PRINT THE LIST:** This option will print out the entire contents of the list, three elements per line with 5 spaces in between elements.

8. **SORT THE LIST:** This option will perform a bubble sort to sort the values in the list in alphabetical order .

9. **SEARCH FOR A SPECIFIC ITEM IN THE LIST:** This option will perform a linear search. It will prompt the user to enter the name of the item for which to search. The operation will search the list, and if the item is found it will print out a message that the item was found, and the index of the array element that contains the value. IF the value is not in the list, an appropriate message should be displayed

10. **QUIT**

Additional Requirements:

- For this assignment you will use message and input dialog boxes to request information from the user. Remember that values returned by input dialog boxes are always strings.
    - Any values input that represent numbers will need to be converted to the appropriate numeric type. IT IS ESSENTIAL THAT YOU USE TRY/CATCH BLOCKS SO THAT THE USER MUST ENTER A VALID NUMBER. Hint: see Integer.parseInt
- Any operations that specify element positions, must ensure that the position given is legal and will not cause an array index out of bounds error.
- Since this is a partially filled array, you will need to keep a count of the actual number of elements that contain good values.

For help on input/message dialog boxes see:

A good tutorial on dialog boxes can be found at:

http://www.homeandlearn.co.uk/java/java_option_panes.html