

Assignment#4

Array Lists and Linked Lists

*When to use
3% of course grade*

Submission Requirements

Complete the following exercise and submit electronically in the assignments folder on eLearn as an IntelliJ Project – Zip the entire folder not just the source files in MyCanvas. Please refer the course Calendar for the exact date and time of the submission. This assignment is to be completed individually.

Background

You are to complete the starting code that has been provided for a **SortedLinkedList** that will store a collection of items and maintain the order of the items at all times. You will need to add functionality for the following methods:

- add
- remove
- toString

In case of the add method, the elements must be added in sorted order. When you add the following words in this order [Bob, Carol, Aaron, Alex, Zaphod], the list when printed using the toString method will appear as [Aaron, Alex, Bob, Carol, Zaphod]. You may not call any sort algorithm to achieve this result. Your goal is to ensure that the list is always in a sorted state. With the remove method, the element specified must be removed from the list and the current order maintained.

The class provided has been provided as a generic class. It will work with any class that is Comparable. In your main method you must demonstrate that this works using two different data types that add elements in random order to the list and when toString is called will display the data in sorted order. The sorting order in all cases will be ascending alphabetic order ('A' at the top, lowest integer at the top, etc...).

Once you have demonstrated that your class can handle two different data types (make one of them String), you will need to compare the performance of the class against the standard java ArrayList class. To achieve equivalent functionality with the ArrayList, add each name to the list add call the Collections.sort method after each addition. This will ensure that we are comparing the same functionality in both classes (always sorted).

The main program as provided will read names from a text file (baby names) and place the content into an array. It is suggested you use this data for comparison purposes for the String data type.

Suggested Steps:

1. Create an add method for the SortedLinkedList<T> class
2. Test the add method from main to ensure elements are added in sorted order. To do this you will need to complete the toString method.
3. Create a remove method for the SortedLinkedList class
4. Test the remove method from main to ensure elements are removed from the list and that after removal the list is still in sorted order.
5. Add a Discussion (in a comment) to the top of the file that contains your main method discussing the results obtained. Answer each of the following questions:
 - o Do you notice any significant performance difference between the SortedLinkedList<T> and the ArrayList<T> classes when adding items?
 - o Do you notice any significant performance difference between these two collections when removing items?
 - o When would you choose to use a SortedLinkedList over an ArrayList based on the results of this assignment?

Marking Scheme

Program structure – Comments, follows best programming practices - 20%

Completion of LinkedList Requirements – 30%

Print timing results to screen for both adding and removing items using the provided ArrayList and LinkedList classes – 25%

Performance comparison write-up as a comment at the top of the code. – 25%