Exercises 08

1. **[EX] Lists in action**
    a. Create a list of integers. Add some elements to it.
    b. Print the elements by using
        i. toString() method
        ii. for statement
        iii. enhanced for statement
        iv. iterator
    c. Create a list of Strings. Name it as *shoppingList*.
        i. Add elements: “coffee”, “tea”, “bread”, “butter”, “eggs”
        ii. Use previous methods to list all the elements of it
        iii. Remove the element “bread” from the list
        iv. Print the list elements again
        v. Remove the element at index 2
    d. Create a new list named *newShoppingList* using the values of the *shoppingList*.
        i. Add elements: “apple”, “kiwi”
        ii. Print the elements
        iii. Check if the element “tea” exists in the *newShoppingList* print an appropriate message
    e. Create a new LinkedList named *newLinkedList*.
        i. Add elements: “tomato”, “potato”
        ii. Add all the elements of the newShoppingList to the *newLinkedList*
        iii. Remove the first element of the list and print the list
        iv. Remove the last element of the list and print the list

2. **[EX] Sort the elements of the list.**
        i. Try Collections.sort() method
        ii. Try List.sort() method
        iii. Try Comparator interface
            1. Create instance of an anonymous class
            2. Create a concrete class implementing Comparator and use its instance
    b. Print the elements again.

3. **[EX] List of Accounts**
    a. Define a class Account.
        i. Define fields: firstname and lastname of type String and balance of type double.
        ii. Define constructor taking firstname, lastname and balance as input arguments resp.
        iii. Define toString() method.
    b. Create a list of Accounts and add some elements to it.
    c. Print the elements.
    d. Sort all the elements based on
        i. fullname (firstname + lastname)
        ii. balance
    e. To sort the list of accounts try using

i. Comparable interface.
ii. Comparator object

4. **[PW] Distinct words**
    a. Given a file with names.
    b. Can you make sure to eliminate duplicate word?
        i. How to ensure that the order of the words are retained?
        ii. How about ordering them by default in alphabetically ascending order?

5. **[PW] Queue Demo**
    a. Imagine we have some people in front of the grocery store. All are in queue for some bread.
    b. Given the number of pieces of bread per each person, and the amount in the store. Find how many persons will be able to buy bread.

6. **[PW] PriorityQueue Demo**
    a. Consider the previous example. What about if it required to sell to as many persons as possible? Update your code, so that it finds the number of maximum number of persons that can be sold required amount of bread.

7. **[PW] Stack**
    a. In the previous weeks, we have already implemented Stack ADT. Perform some Stack operations using a Linked List.