PER SCHOLAS

Guided LAB - 303.11.4 - LinkedList Processing

Objective

In this lab, we will explore and demonstrate LinkedList using built-in methods.

Objective

At the end of this lab, learners will have demonstrated the ability to use LinkedList using built-in methods.

Example One: LinkedList Methods

Create a class named "`LinkedListExample`" and add the code below:

```
import java.util.*;
public class LinkedListExample {
     public static void main(String args[]) {

       /* Linked List Declaration */
       LinkedList<String> linkedlist = new LinkedList<String>();

       /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add("Item1");
       linkedlist.add("Item5");
       linkedlist.add("Item3");
       linkedlist.add("Item6");
```

```
        linkedlist.add("Item2");

        /*Display Linked List Content*/
        System.out.println("Linked List Content: " +linkedlist);

        /*Add First and Last Element*/
        linkedlist.addFirst("First Item");
        linkedlist.addLast("Last Item");
        System.out.println("LinkedList Content after addition: " +linkedlist);

        /*This is how to get and set Values*/
        Object firstvar = linkedlist.get(0);
        System.out.println("First element: " +firstvar);
        linkedlist.set(0, "Changed first item");
        Object firstvar2 = linkedlist.get(0);
    System.out.println("First element after update by set method: " +firstvar2);

        /*Remove first and last element*/
        linkedlist.removeFirst();
        linkedlist.removeLast();
        System.out.println("LinkedList after deletion of first and last element:
" +linkedlist);

        /* Add to a Position and remove from a position*/
        linkedlist.add(0, "Newly added item");
        linkedlist.remove(2);
        System.out.println("Final Content: " +linkedlist);
      }
}
```

Output:

```
Linked List Content: [Item1, Item5, Item3, Item6, Item2]
LinkedList Content after addition: [First Item, Item1, Item5,
Item3, Item6, Item2, Last Item]
First element: First Item
First element after update by set method: Changed first item
```

```
LinkedList after deletion of first and last element: [Item1,
Item5, Item3, Item6, Item2]
Final Content: [Newly added item, Item1, Item3, Item6, Item2]
```

Example Two: How to Sort LinkedList

In this example, we will demonstrate how to sort a LinkedList using the **Collections.sort()** method. Please note that for custom sorting of objects, we can use Collections.sort(linkedList, comparator) method.

Create a class named "***LinkedListsortExample***" and add the code below:

```
public class LinkedListsortExample {
   public static void main(String[] args)
   {
       //Create linked list
       LinkedList<String> linkedList = new LinkedList<>();
       linkedList.add("A");
       linkedList.add("C");
       linkedList.add("B");
       linkedList.add("D");

//Unsorted
       System.out.println(linkedList);

//1. Sort the list
       Collections.sort(linkedList);

//Sorted
       System.out.println(linkedList);

//2. Custom sorting
       Collections.sort(linkedList, Collections.reverseOrder());

//Custom sorted
       System.out.println(linkedList);
   }
}
```

Output:

```
[A, C, B, D]
[A, B, C, D]
[D, C, B, A]
```

Submission Instructions:

Include the following deliverables in your submission -

- Submit your source code using the Start Assignment button in the top-right corner of the assignment page in Canvas.

CANVAS STAFF USE ONLY: Canvas Submission Guideline:

Instructions for Canvas Assignment Creation
Assignment Name: **GLAB - 303.11.4 - LinkedList Processing** **Points:** **100** **Assignment Group:** **Module 303: Java SE Review (Not Graded)** **Display Grade As:** **Complete/Incomplete** **Do not count this assignment towards the final grade:** **Checked** **Submission Types:** **File Uploads** **Everything else is the default.**