Java-Linked List Methods

Project Abstract

The purpose of this project is to demonstrate how to perform operations on a `LinkedList` in Java. This task helps you to understand how to use the `LinkedList` class to store elements and perform various operations, such as adding, removing, and retrieving elements from both ends of the list. The project focuses on using methods such as `addFirst()`, `addLast()`, `removeFirst()`, `removeLast()`, `getFirst()`, and `getLast()` to manipulate the LinkedList.

This project focuses on:
 1. Understanding how to create and use a `LinkedList` in Java.
 2. Adding items to a `LinkedList` using methods like `addFirst()` and `addLast()`.
 3. Removing items from a `LinkedList` using `removeFirst()` and `removeLast()`.
 4. Retrieving the first and last items using `getFirst()` and `getLast()`.
 5. Demonstrating the impact of LinkedList operations on its structure.

Tasks Overview

Task 1: Add, Remove, and Get Items in a LinkedList
ObjectiveObjective: Perform various operations on a LinkedList, including adding, removing, and retrieving elements.

Detailed Description: In this task, you will create a `LinkedList` of integers and perform the following operations: adding elements to the front and back of the list using `addFirst()` and `addLast()`, removing elements from the front and back using `removeFirst()` and `removeLast()`, and retrieving the first and last elements using `getFirst()` and `getLast()`.

Steps:
 1. **Create a LinkedList:**
    - Declare a `LinkedList` object named `linkedList` to store `Integer` values.

 2. **Add Items to the LinkedList:**
    - Use the `addFirst()` method to add an item at the front of the list (add 3).
    - Use the `addLast()` method to add items at the end of the list (add 1 & 4).
    - Print the `LinkedList` to show the items after adding them as `"LinkedList after adding items using addFirst() and addLast(): " + linkedList`.

3. **Remove Items from the LinkedList:**
   - Use the `removeFirst()` method to remove the item at the front of the list.
   - Use the `removeLast()` method to remove the item at the end of the list.
   - Print the `LinkedList` again to show the updated list after removing items as `"LinkedList after removing items using removeFirst() and removeLast(): " + linkedList`.

4. **Get the First and Last Items:**
   - Declare a `firstItem` of type `Integer` and use the `getFirst()` method to retrieve and save it.
   - Declare `lastItem` of type `Integer` and use the `getLast()` method to retrieve and save it.

- Print both variables as `"First item: " + firstItem + ", Last item: " + lastItem`.

Execution Steps to Follow:

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**

2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) Terminal New Terminal.**

3. **This editor Auto Saves the code.**

4. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**

5. **To run your project use command:**
   sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn compile exec:java -Dexec.mainClass="com.yaksha.assignment.LinkedListOperations"

   *If it asks for the password, provide password : pass@word1

6. **To test your project test cases, use the command**
   sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn test

   *If it asks for the password, provide password : pass@word1