LAB - Single Linked List Overview

• The goal of this lab session is to familiarise yourself with the functionality of the abstract data type: Single Linked list.

• Understand how to implement a Single Linked list

Task – Part 1

• Illustrate in a drawing the process of adding a new node on the first position. Explain what happens in each step.

Let’s take this linked list

A 🡺 B 🡺 C

1. Create a new element for example (X)
2. Set the pointer of the (X) object to the next element for example (A)
3. Assign the head to the (X)

X 🡺 A 🡺 B 🡺 C

A screenshot of a computer

Description automatically generated

• Illustrate in a drawing the process of adding a new node on the last position. Explain what happens in each step.

Let’s take this linked list

A 🡺 B 🡺 C

1. Create (X) a new element.
2. A loop to go to the end of the elements.
3. Change the pointer of © from null to the (X) element.
4. Set the next pointer of (X) to null.

A 🡺 B 🡺 C 🡺 X

A screenshot of a computer

Description automatically generated

Task – Part 2

• Design the object/class diagram for a single linked list that allows you to reverse the items in the list.

• Included will be a method to reverse all the items in the list.

• Create a NetBeans application that implements a Single Linked List and uses this ADT. Include an interface, a node, and a Single Linked List class.

• Use the app class to demonstrate checking the size, if the list is empty, adding, removing, and reversing the list.

• Add this to a repo on GitHub and copy/paste the link to Moodle.

Task – Part 3

• Design the object/class diagram for a single linked list that implements the queue principle.

• Add items to the linked list implementing your Queue Interface.

• The Single linked list interface will outline your other methods such as size and isEmpty.

• Demonstrate the implementation of enqueue, dequeue, size, isEmpty and display in the app class.

• Add this repo to github and copy/paste the link.