

# INTRODUCTION TO COMPLEX SYSTEMS, JAVA, MVN, AND GIT

Johan Sebastian Arias

August 2020

## 1 Introduction

This is a program developed in Java in conjunction with Apache Maven and Git which calculates the mean and standard deviation given a text file of n real numbers.

## 2 Objectives

1. To Understand the basics of maven and git via command line.
2. To Recognize and implement a Linked List in order to manage collections of data.
3. To Implement basic statistical operations like the mean and the standard deviation and identify its importance.

## 3 Definitions and Context

- **Linked List:** A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in the below image: [1]

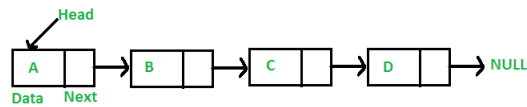


Figure 1: Graphic representation of a Linked list

- **Singly Linked List:** It is the most common. Each node has data and a pointer to the next node. [2]

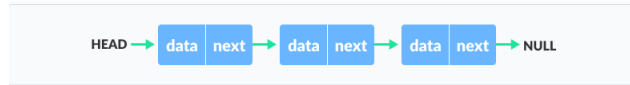


Figure 2: Singly Linked List Representation

- **Doubly Linked List:** We add a pointer to the previous node in a doubly-linked list. Thus, we can go in either direction: forward or backward. [2]



Figure 3: Doubly Linked List Representation

- **Circular Linked List:** A circular linked list is a variation of a linked list in which the last element is linked to the first element. This forms a circular loop. [2]

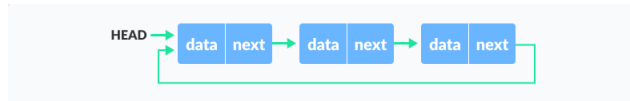


Figure 4: Circular Linked List Representation

#### 4.1 Class diagram:

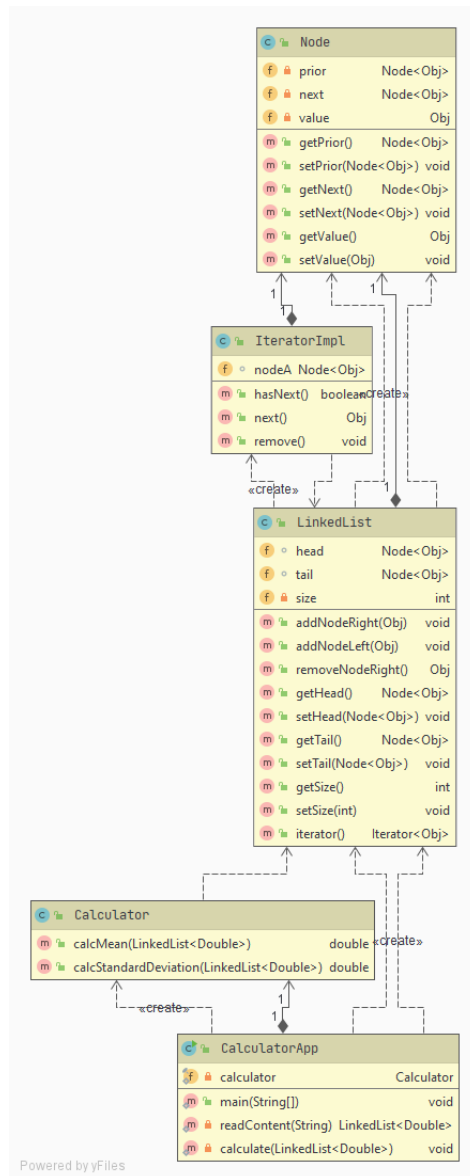


Figure 5: Class Diagram of the design

## 4.2 Design's description:

### 4.2.1 Linked List:

Three classes are involved in the construction of this data structure:

1. **Node:** This class is a generic class and represents the element that will conform the LinkedList, each node has an assigned value and a neighbor node which will be referenced. Within this class we can highlight some methods like:
  - **getValue:** Returns the value assigned to the current node.
  - **setPrior:** Updates the pointer of the previous node with respect to the current node.
  - **setNext:** Updates the pointer of the next node with respect to the current node.
  - **getPrior:** Returns the previous node of the current node.
  - **getNext:** Returns the next node of the current node.
2. **LinkedList:** This class represents the structure that links the nodes. This abstraction contains a head, a tail and a size, and implements the Java **Iterable** interface. Among the methods we can highlight we have:
  - **addNodeRight :** Add a node on the right side of the list.
  - **addNodeLeft :** Add a node on the left side of the list.
  - **removeNodeRight :** Delete a node on the right side of the list.
  - **iterator :** It is the method that must be implemented by the Iterator interface and serves to go through the structure of the LinkedList.
3. **IteratorImpl:** This abstraction serves to go through our LinkedList, that is, to move between nodes. The important methods are:
  - **hasNext:** Returns true if the next node is not empty.
  - **next:** It returns the next node with respect to the current one.

### 4.2.2 Main functionality:

In simple words, this program reads a text file with n numbers separated by comma and calculates the mean and the standard deviation based on the data structure described below.

## 5 Testing the app

### Requisites

- Apache Maven
- Java 8
- git

### Steps to run this project locally

1. Clone this repository:

```
git clone https://github.com/JohanS11/Taller1Arep.git
```

2. Build the project with maven:

```
cd Taller1Arep && mvn package
```

3. Execute the project with maven:

```
mvn exec:java -Dexec.mainClass="edu.escuelaing.arep.calculator.app.CalculatorApp" -Dexec.args="--DataTest\\DataSet.txt"
```

### NOTE:

In order to test with another text file you have to add it to the folder "DataTest" and modify the below command with the name of the txt file: -  
Dexec.args="DataTest \\yourtextfile.txt"

(The values have to be separated by comma)

## 5.1 Proof of concept

```
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ CalculatorApp ---
[+] Mean: 47.71
[+] Standard Deviation: 24.41
[+] Mean: 60.02
[+] Standard Deviation: 62.73
[+] Mean: 46.1
[+] Standard Deviation: 26.94
[+] Mean: 49.04
[+] Standard Deviation: 27.01
[+] Mean: 334.92
[+] Standard Deviation: 437.8
[+] Mean: 60.41
[+] Standard Deviation: 26.4
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 1.365 s
[INFO] Finished at: 2020-08-13T12:32:04-05:00
[INFO] -----
```

Figure 6: POC main class executing

```
-----
T E S T S
-----
Running edu.escuelaing.arep.calculator.app.AppTest
Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.092 sec

Results :

Tests run: 7, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.692 s
[INFO] Finished at: 2020-08-13T12:33:47-05:00
[INFO] -----
```

Figure 7: POC tests passed

## 6 Conclusion

I have recovered some knowledge about tools like Apache Maven, Git, Java and algorithms as well. In the same way I have identified some utilities from the Linked Lists like the response time, its implementation is not that hard.

## References

- [1] Linked list data structure. <https://www.geeksforgeeks.org/data-structures/linked-list/#singlyLinkedList>. Accessed on 2020-08-09.
- [2] Types of linked list - singly linked, doubly linked and circular. <https://www.programiz.com/dsa/linked-list-types#singly>. Accessed on 2020-08-09.