Introduction To Complex Systems, Java, MVN, And Git

Autor:
Carlos Andrés Ramírez Torres

Arquitecturas Empresariales

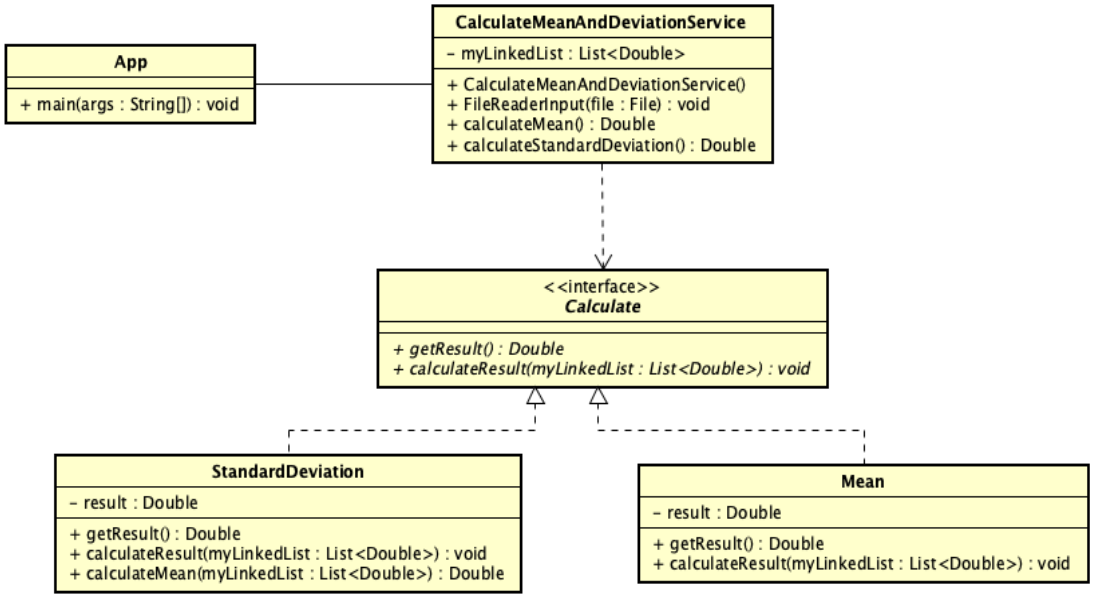
Ingeniería de Sistemas

07 de Agosto del 2020

1 Introduction

In this workshop, we look for a file containing data that allows the calculation of the mean and the standard deviation. We proceed to read line by line and save it in a data structure that allows the corresponding iteration to obtain the appropriate results. Additionally, our own linked list will be implemented based on the Java documentation in order to store the data that is read from an external file. The data structure will allow us to iterate over the data and make the respective calculation.

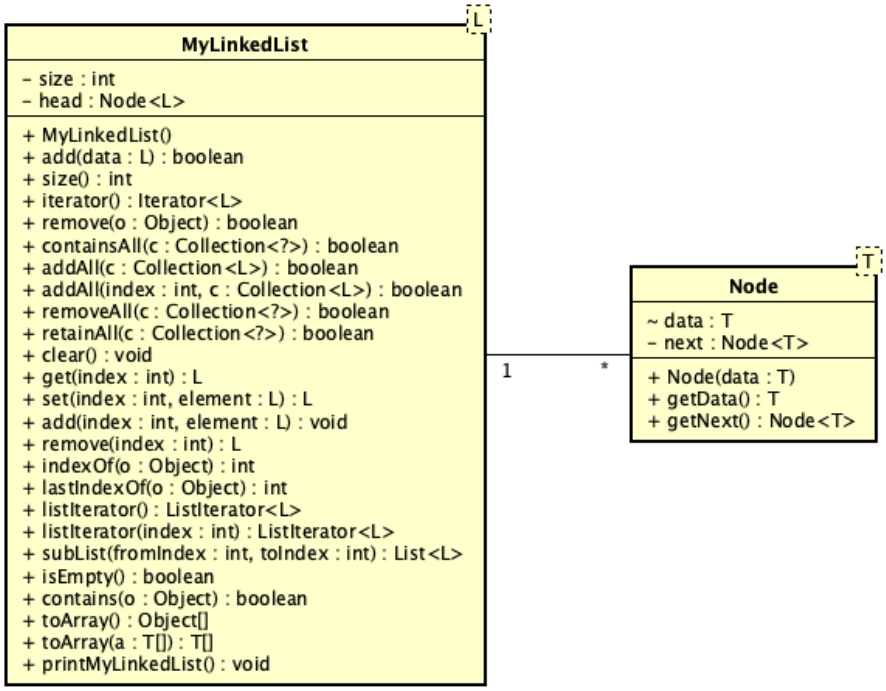
2 Desing



The diagram shows that the App class receives the argument, in this case it is the file path to which the calculation corresponding to the mean and standard deviation will be made, the App class creates an instance of the class **CalculateMeanAndDeviationService** whose purpose is to read the found file and through the methods **calculateMean** and **calculateStandardDeviation** it sends a **MyLinkedList** that contains the data of the file read before, to be used by the **Calculate** interface which with the use of **polymorphism** will obtain the calculation corresponding to the mean or standard deviation, this structure will allow possible changes to be im-

plemented as in the case where the user wants to obtain only one value and in turn the software extension for other calculations.

2.1 Linked List



For the implementation of the Linked List data structure and given the requirement to correspond with the **java api**, we proceed to use two classes **MyLinkedList** which implements List and the class called Node which will allow you to save the corresponding data you want, this is achieved through **Generic Types**, some methods implemented for the use of the structure are

- **Add**: Allows to add objects to the structure.
- **Size**: Allows to obtain the amount of objects in the structure.
- **Iterator**: Allows to use for each facilitating the iteration by the structure.
- **Remove**: Allows to remove a node depending on an indicated position.

3 Conclusion

- The program makes the calculations corresponding to the average and standard deviation in a correct way, making the reading of files or directories containing files with the respective data sample, the calculation is made with a rounding to two decimal places..
- A proprietary implementation of the linked list data structure was made, which implements the java list interface.

4 Test

A total of 16 tests have been performed where different scenarios are contemplated to show that the code is correct, among the tests the following events were tested

- Reading existing files.
- Reading files not found.
- Reading files contained in a directory.
- Calculation of the average of a data sample.
- Calculation of the Standard Deviation of a data sample.
- Linked list method test implemented.

For the development of the tests due to the rounding off of the results, it was chosen to use a 1 percent error delta in some cases.