

INTERPRETERS, COMMUNICATION CHANNELS AND MEMORY

Nicolás Ortega Limas

February 2021

1 Introduction

Some of us have been asking about how was software projects were being managed. Well Apache Spark has basically changed a lot is a software that allows us to creating products and projects worldwide. When creating a Spark project, a very special, very organized and predefined structure that we must follow to ensure the quality of our product.

We will now do our own implementation with Spark, to show it's practical help into web environments. We hope that it will help us with good separation of libraries in our application.

2 Objectives

- Recognizing and implementing everything needed to put our application into a web environment like Heroku.
- Recognizing and implementing everything needed to put our web-application into the continuous integration tool, in our case Circle CI.
- Recognizing and implement a Linked List in order to manage collections
- Implementing basic statistical operations like the mean and the standard deviation and identify its importance.
- Understand the basics of Spark, Maven, Git and Java via command line.

3 Problem proposed

We are required to write a web application that calculates the mean and standard deviation of a set of n real numbers.

It is also a requirement to read the n real numbers from a file and to implement the use a linked list to store the n numbers for the calculations. (Note: You have to write your own implementation of a linked list and it must be compliant with Java's collections API)

$$x_{avg} = \frac{\sum_{i=1}^n x_i}{n}$$

Figure 1: formula for calculating the mean

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - x_{avg})^2}{n-1}}$$

Figure 2: formula for standard deviation

4 Solution

We will develop a full java application that solves this problem by implementing this two formulas presented above. [2]

As it says in the guide that i used to help me build my own LinkedList Class “It’s good to understand how things work, and understanding linked lists is an important step towards understanding more complex data structures, many of which don’t exist in the standard libraries.” [1]

We are going to use the formula for calculating the mean stated in Figure 1 and the formula for standard deviation stated in Figure 2.

- S is the symbol for summation
- i is an index to the n numbers
- x is the data in the set
- n is the number of items in the set

5 Design

To deploy the web app i decide to use Heroku. The app design is decribed by Figure

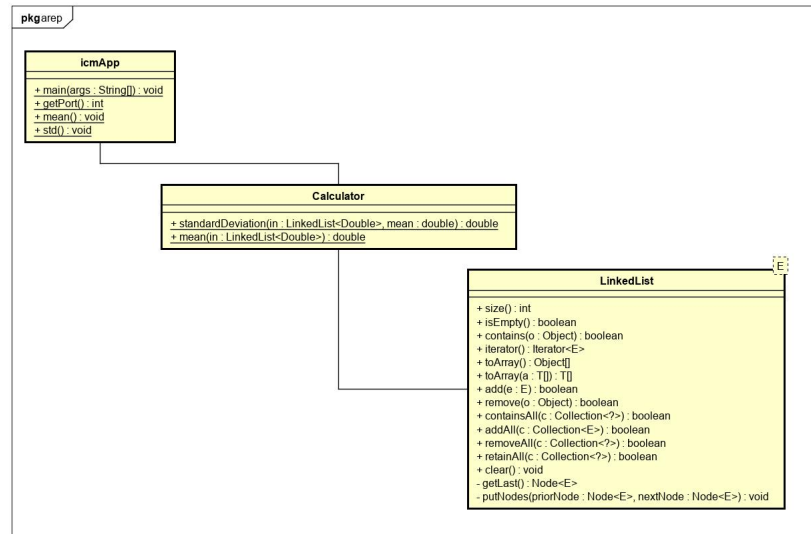


Figure 3: Class Diagram

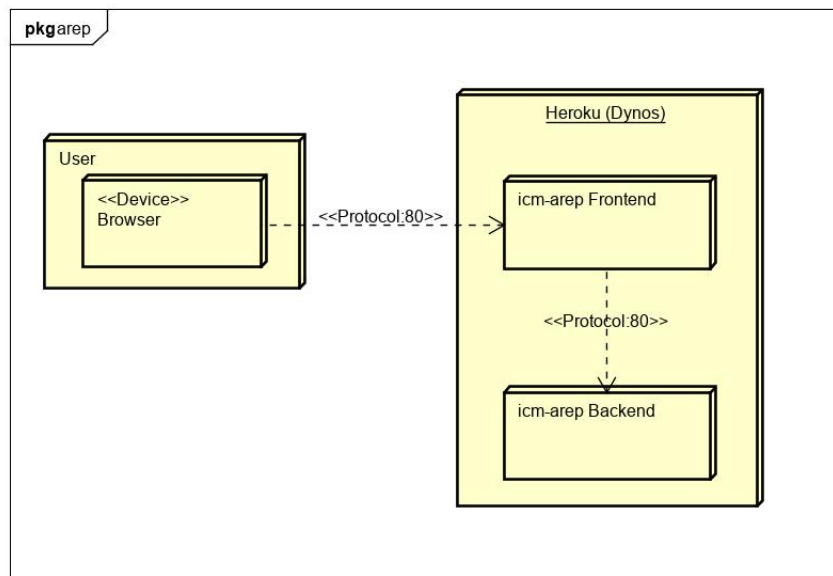


Figure 4: Deploy diagram

DATASET 1
160
591
114
229
230
270
128
1657
624
1503

Figure 5: Data-set one

DATASET 2
15.0
69.9
6.5
22.4
28.4
65.9
19.4
198.7
38.8
138.2

Figure 6: Data-set two

6 Tests

Now we can start to review the six tests done on the project, they should use the presented data sets found on Figure 3, 4 and 5, we should get the results found on Figure 6. With the test executing correctly we can now say we finished.

7 Conclusions

Some of us forget how these basics things are done. Then it is always helpful to remember things like Apache Maven, Apache Spark, Git, Java and general algorithm design as well. In the same way I have identified some utilities from the Linked Lists like the response time, its implementation is really help full at the time of developing an upper performance data structure that allows us to perform high level actions.

Web application development is always useful, due to its complexity and detail oriented focus we need to be in touch really often with this type of practices, sometimes you get to rest a couple of months and tent to forget a lot of the stuff needed to do correct and useful things, so in my case i really appreciate the challenge proposed. Also, spark is a fun framework to learn, in my case, it pushed myself to a complex handling of files and responses, so definitely is a

DATASET 2
15.0
69.9
6.5
22.4
28.4
65.9
19.4
198.7
38.8
138.2

Figure 7: Data-set three

TESTS	EXPECTED VALUES	
	MEAN	STD.DEVIATION
Data set 1	550.6	572.02
Data set 2	60.32	62.255
Data set 3	638.9	625.63

Figure 8: Expected tests results

```
[ INFO] -----
[ INFO] T E S T S
[ INFO] -----
[ INFO] Running edu.escuelaing.arem.icm.arem.icmAppTest
std
572.026844746915
std
62.25583060601187
std
625.6339806770231
mean
550.6
mean
60.32000000000001
mean
638.9
[ INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 3.45 s - in edu.escuelaing.arem.icm.arem.icmAppTest
[ INFO] Results:
[ INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[ INFO]
[ INFO] BUILD SUCCESS
[ INFO]
[ INFO] Total time: 6.375 s
[ INFO] Finished at: 2021-02-05T21:56:18-05:00
[ INFO] -----
C:\Users\User\Desktop\AREP\LAB 2\MINE\icm-arem>
```

Figure 9: Tests executed correctly

obligatory on your career.

References

- [1] How to implement a linkedlist class from scratch in java. <https://crunchify.com/how-to-implement-a-linkedlist-class-from-scratch-in-java/>. accessed: 29.01.2021.
- [2] Read integers from text files in java. <https://www.codegrepper.com/code-examples/java/java+read+integer+from+text+file+into+array+scanner>. Accessed: 29.01.2021.