

Arquitectura Empresarial Laboratorio-01

Johann Steven Bogotá Vélez

Estudiante, Escuela Colombiana de Ingeniería Julio Garavito

Ingeniería de Sistemas

Email: johann.bogota@mail.escuelaing.edu.co

Bogotá, Colombia

January 27, 2021

Abstract

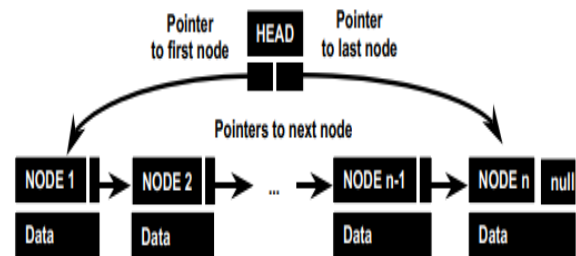
Se va a resolver un problema presentado por la universidad Carnegie Mellon usando las herramientas de git y maven que permite una buena construcción y mantenibilidad del software en el lenguaje de programación de Java, además para resolver este problema se va a trabajar con una implementación de una LinkedList.

1 Introducción

En este artículo se escogieron dos herramientas con las que se trabajará para construir software con un estándar que nos proporciona Maven y el control de versiones que nos ofrece Git, una vez construido el proyecto se empezará a desarrollar un software capaz de resolver un problema presentado por la universidad de Carnegie Mellon, donde nos indica que se quiere obtener la media y la desviación de un conjunto de datos enviados en un archivo, para esto se tiene propuesto resolverlo por medio de una LinkedList, pero en este caso, se implementará basándonos en el api que nos proporciona java, teniendo una nueva solución sobre esta estructura de datos dada por el programador

2 Diseño

Para la implementación de una LinkedList, se tomó el concepto donde se especifica que es una abstracción de un tipo de dato para mantener una colección de datos, además de que se usan apuntadores y están compuesto principalmente de dos componentes, una cabeza y un nodo.



Ya teniendo claro el concepto de LickedList, en la implementación se utilizó una clase llamada Nodo, que está compuesta de dos atributos, el dato (que es de tipo Object para mayor flexibilidad), y el apuntador que indica el siguiente elemento en la lista, por otro lado, la clase LickedList tienen dos atributos, el principal llamado head, que representa la cabeza, este atributo es especial ya que su dato se entiende que es de tipo Nodo, es decir su atributo dato se convierte en un apuntador para indicar el primer elemento de la lista, y su otro atributo, indicando el último elemento de la lista.

En la implementación de la clase, se sobre escribieron los métodos addFirst, addLast, removeFirst, getFirst, getLast, remove, poll, peek, push, pop y get.

Para probar que todo esté funcionando correctamente, se corrieron las pruebas utilizando los comandos de maven, dando como resultado pruebas exitosas.

```

-----
T E S T S
-----
Running edu.escuelaing.arep.app.CalculatorTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.047 sec

Results :

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.233 s
[INFO] Finished at: 2021-01-26T22:37:46-05:00
[INFO] -----
C:\Users\johan\Documents\AREP\LAB-01\miprimera-app>

```

```

@Test
public void testCalculator(){
    Calculator calculator = new Calculator();
    HashMap<String, Double> result = calculator.calculate("./src/test/java/edu/escuelaing/arep/resources/Data1.txt");
    Assert.assertEquals(550.6,result.get("Mean"),0.2);
    Assert.assertEquals(572.03,result.get("Deviation"),0.2);
}

@Test
public void testCalculator2(){
    Calculator calculator = new Calculator();
    HashMap<String, Double> result = calculator.calculate("./src/test/java/edu/escuelaing/arep/resources/Data2.txt");
    Assert.assertEquals(60.32,result.get("Mean"),0.2);
    Assert.assertEquals(62.26,result.get("Deviation"),0.2);
}

@Test
public void testCalculator3(){
    Calculator calculator = new Calculator();
    HashMap<String, Double> result = calculator.calculate("./src/test/java/edu/escuelaing/arep/resources/Data3.txt");
    Assert.assertEquals(638.9,result.get("Mean"),0.2);
    Assert.assertEquals(625.63,result.get("Deviation"),0.2);
}

```

Por ultima, se quiere generar la API del software, por lo que maven nos ofrece este comando si ya colocamos el plugin en las dependencias del archivo .pom, ejecutando mvn javadoc:javadoc, obtendremos el siguiente index.html con toda la documentación.

| OVERVIEW | PACKAGE | CLASS | USE | TREE | DEPRECATED | INDEX | HELP |
|--|-------------|--------|-----------|------|------------|-------|------|
| PREV | NEXT | FRAMES | NO FRAMES | | | | |
| miprimera-app 1.0-SNAPSHOT API | | | | | | | |
| Packages | | | | | | | |
| Package | Description | | | | | | |
| edu.escuelaing.arep.Calculator | | | | | | | |
| edu.escuelaing.arep.Implementation | | | | | | | |
| OVERVIEW | PACKAGE | CLASS | USE | TREE | DEPRECATED | INDEX | HELP |
| PREV | NEXT | FRAMES | NO FRAMES | | | | |
| Copyright © 2021. All rights reserved. | | | | | | | |

3 Conclusión

Como conclusión, se puede decir que utilizar Maven y Git son de gran apoyo a la hora de crear nuestros programas para el desarrollo de software, ya que utilizamos estandares que nos ayudan a tener nuestro trabajo mejor organizados y con una estructura completa, obtuvimos un software de gran calidad que nos da a entender la gran ayuda que nos da estas herramientas.

References

Documentación LinkedList.<https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>