

INGENIERÍA DE SISTEMAS - AREP 2020/-2

Arquitecturas Empresariales

LABORATORIO 1 : EJERCICIO MVN-GIT

Luis Daniel Benavides Navarro

Authors:
Juan Camilo Rojas Ortiz

Índice

1. Introducción	2
2. Lista enlazada	2
2.1. Implementación	2
3. Estadísticas	3
3.1. Media	3
3.2. Desviación estándar	4
3.3. Implementación	4
4. Aplicación principal	4
5. Pruebas	4
6. Resultado	5
7. Referencias	6

Resumen

Este artículo presenta la descripción de la teoría y la arquitectura utilizada para la creación de un sistema capaz de leer un conjunto de datos de un archivo de texto y realizar el cálculo de la media y la desviación estándar de estos, utilizando una implementación propia de lista enlazada. Al final se presentan los datos de prueba (entrada y salida) propuestos para comprobar el correcto funcionamiento del programa y los resultados obtenidos al ejecutar el programa con estos datos, con los que se concluye que la implementación calcula correctamente la media y la desviación estándar de los datos ingresados.

1. Introducción

El objetivo de este laboratorio crear un programa para calcular la media y la desviación estándar de n números reales que se leen a un archivo y se almacenan en una lista enlazada (Linked List) que debe ser implementada.

Con el objetivo de explicar la arquitectura del sistema, esta se dividirá en 3 secciones, donde se explica la implementación y la teoría detrás de la lista enlazada creada y el módulo de estadísticas propuesto, además en la última sección se explica el funcionamiento de la aplicación principal.

2. Lista enlazada

Una lista enlazada es una estructura lineal de datos en la que cada elemento es un objeto separado. Los elementos de la lista enlazada no se almacenan en un lugar contiguo; los elementos se enlazan mediante punteros. Cada nodo de una lista se compone de dos elementos: los datos y una referencia al siguiente nodo. El último nodo tiene una referencia a nulo. [1].

2.1. Implementación

Para este problema se implementó una variación de la lista enlazada, la lista doblemente enlazada, en la que cada elemento además de tener una referencia al siguiente nodo también cuenta con una referencia al nodo anterior y el primer nodo cuenta con una referencia previa nula. La decisión de implementar la lista doblemente enlazada se debe a que esta permite buscar un elemento realizando $\frac{n}{2}$ operaciones con n siendo la longitud de la lista, por lo que si en el futuro se necesitan funcionalidades que requieran de esto, se podrán implementar de forma eficiente. En la figura 1 se puede ver la estructura de una lista doblemente encadenada.

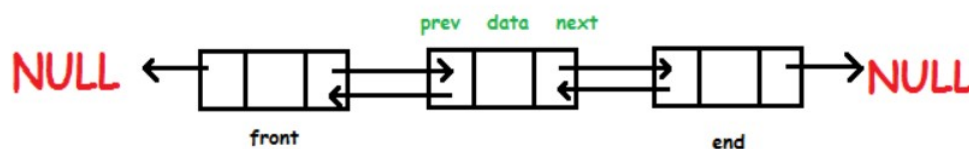


Figura 1: Lista doblemente encadenada [2]

Adicionalmente se realizó una implementación generalizada de la lista, por lo que permite almacenar y hacer operaciones con objetos de cualquier tipo, también se creó la clase implementando la interfaz *Iterable* por lo que la lógica correspondiente a la iteración de la estructura de datos se delega a la clase *LinkedListIterator* y del lado del consumidor se facilita por medio del método *foreach()*, A continuación se presenta el diagrama de clases correspondiente a la implementación realizada

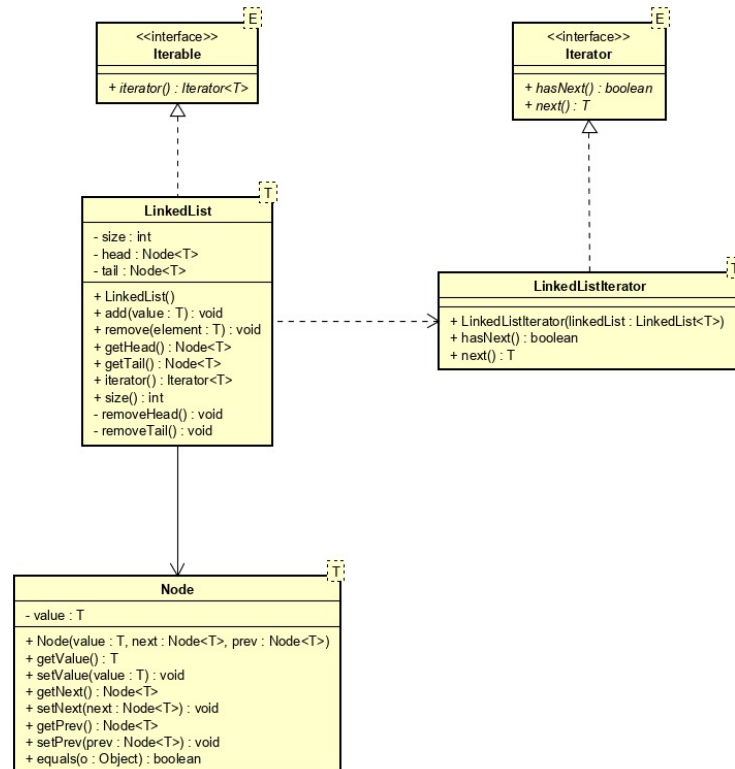


Figura 2: Diseño de la LinkedList implementada

Las operaciones principales que ofrece la estructura de datos implementada son :

- *add* : Permite añadir un nuevo elemento al final de la lista
- *remove*: Permite remover un elemento especificado si este si encuentra en la lista
- *size*: Retorna el tamaño de la lista

3. Estadísticas

3.1. Media

La media es el promedio de un conjunto de datos. La media es la medida más común de la ubicación de un conjunto de números. El promedio localiza el centro de los datos[3].Se calcula de la siguiente

manera:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

3.2. Desviación estándar

La desviación estándar es una medida de la separación o dispersión de un conjunto de datos. Cuanto más ampliamente se separan los valores, mayor es la desviación estándar[3]. Su fórmula es :

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \quad (2)$$

3.3. Implementación

Para el cálculo de la media y la desviación estándar se creó la clase *StatisticsModule* que proporciona los siguientes métodos estáticos:

- *mean*: Calcula la media de un conjunto de datos almacenado en una LinkedList que recibe como parámetro, retorna la respuesta redondeada a 2 decimales
- *standardDeviation*: Calcula la desviación estándar de un conjunto de datos almacenado en una LinkedList que recibe como parámetro, retorna la respuesta redondeada a 2 decimales

Se decidió que los métodos sean estáticos para que estos puedan ser utilizados sin necesidad de instanciar la clase, lo que facilita su uso.

4. Aplicación principal

La lógica principal del programa se encuentra en la clase App, esta presenta dos métodos que se describen a continuación:

- *BuildListFromInputFile* : Construye una LinkedList a partir de los datos que lee de un archivo localizado en una ruta que recibe como parámetro.
- *main* : Inicializa la aplicación, construye las listas y calcula la media y el valor estándar de cada conjunto de datos utilizando los métodos ya especificados.

5. Pruebas

Para verificar el correcto funcionamiento del programa se utilizaron los datos mostrados en la figura 3, para esto, los valores de la columna 1 y la columna 2 se almacenaron en los archivos *input1.txt* e *input2.txt* respectivamente. Los valores esperados tras ejecutar el programa con estos valores se pueden observar en la figura 4 .

Column 1	Column 2
Estimate Proxy Size	Development Hours
160	15.0
591	69.9
114	6.5
229	22.4
230	28.4
270	65.9
128	19.4
1657	198.7
624	38.8
1503	138.2

Figura 3: Valores de entrada

Test	Expected Value		Actual Value	
	<i>Mean</i>	<i>Std. Dev</i>	<i>Mean</i>	<i>Std. Dev</i>
Table 1: Column 1	550.6	572.03		
Table 1: Column 2	60.32	62.26		

Figura 4: Valores esperados para cada columna de entrada

6. Resultado

Tras ejecutar el programa con los casos de prueba propuestos anteriormente se consiguieron los resultados mostrados en la figura 5, que cómo se puede observar corresponden con los valores esperados mencionados en la tabla de la figura 4.

```

C:\Users\usuario\Documents\GitHub\AREP-taller1>mvn exec:java -D "exec.mainClass"="edu.escuelaing.arep.ASE.app.App"
[INFO] Scanning for projects...
[INFO]
[INFO] -----< edu.escuelaing.arep.ASE.app:Lab1Arep >-----
[INFO] Building Lab1Arep 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec-maven-plugin:3.0.0:java (default-cli) @ Lab1Arep ---
La media para los valores del archivo input1 es: 550.6
La media para los valores del archivo input2 es: 60.32
La desviación estándar para los valores del archivo input1 es: 572.03
La desviación estándar para los valores del archivo input2 es: 62.26
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 0.790 s
[INFO] Finished at: 2020-08-07T14:59:05:00
[INFO] -----

```

Figura 5: Resultados obtenidos al ejecutar el programa

7. Referencias

- [1] InterviewBit, *Linked-List*, <https://www.interviewbit.com/courses/programming/topics/linked-lists/#:~:text=A%20linked%20list%20is%20a,has%20a%20reference%20to%20null.>, Accessed on 2020-08-07.
- [2] A. Ahlawat-Studytonight, *Doubly Linked List*, <https://www.studytonight.com/data-structures/doubly-linked-list#:~:text=Doubly%20linked%20list%20is%20a,next%20node%20in%20the%20list.>, Accessed on 2020-08-07.
- [3] http://campusvirtual.escuelaing.edu.co/moodle/pluginfile.php/181210/mod_resource/content/0/EnunciadoTallerEjercicioMVNGit.pdf, Accessed on 2020-08-07.