

Taller de Introducción a Maven y Git

Calculadora Estadística

Davor Cortés, Escuela Colombiana de Ingeniería

Abstract

Este artículo presenta la implementación de una calculadora estadística, con dos operaciones fundamentales como lo son la media de un conjunto de datos y su desviación estándar. Además, como parte del reto se implementó una lista encadenada simple y algunas de sus funcionalidades básicas. Los resultados obtenidos apuntan a que la calculadora cumple con la funcionalidad requerida y de allí también podemos concluir que la estructura de datos implementada cumple bien su papel.

Index Terms

Maven, Git, Dependencia, Clase Genérica, Clase Genérica, Automatización, Lista enlazada.

I. INTRODUCCIÓN

LOS cálculos estadísticos nos ayudan a proveer información importante acerca de un conjunto de datos que por si solos no dicen mucho. La mayor parte del tiempo resultan en operaciones básicas repetitivas y en las que es fácil cometer un error por este motivo podemos delegarle la tarea de hacer estos cálculos a las máquinas.

Se debe implementar una calculadora estadística con dos funcionalidades, calcular la media y la desviación estándar de un conjunto de n números reales, haciendo uso de herramientas tecnológicas como Maven y Git. Los números reales deben ser leídos de un archivo de texto y almacenados en una lista enlazada que cuente con las operaciones básicas correspondientes a la colección de la API de Java.

Los resultados obtenidos serán evaluados por medio de pruebas unitarias, realizadas a la calculadora con datos como el estimado de horas de desarrollo. También se presenta la cobertura de las pruebas en cada uno de los paquetes, clases y métodos usados para la implementación.

II. DESCRIPCIÓN DEL PROBLEMA

El problema de la implementación se ocasiona algunas veces por una mala abstracción de los elementos que componen el sistema es por esto que debemos definir las operaciones y estructuras de datos a utilizar.

A. Lista Enlazada

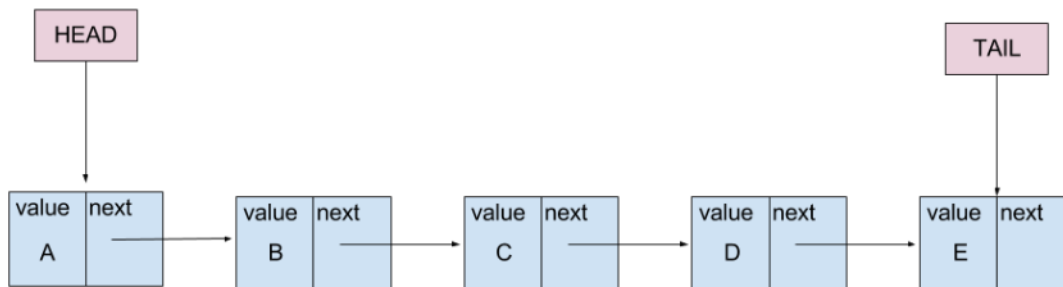


Fig. 1. Lista enlazada simple

Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior.

B. Descripción

- **Node** : Es una clase genérica encargada de encapsular un dato de tipo N y de tener una referencia a un nodo siguiente. Está es la unidad básica con la que se formará la lista enlazada.
- **LinkedList** : Es una clase genérica que representa una abstracción de la lista enlazada simple, esta clase está compuesta de nodos y permite coleccionar datos de tipo N.
 - size(): Devuelve un entero que representa la cantidad de elementos coleccionados.
 - add(N): Colecciona un nuevo elemento.
 - pop(): Remueve y retorna el primer elemento de la colección.
 - remove(): Remueve el primer elemento de la colección
 - iterator(): Devuelve un iterable de la colección para que sea posible iterar sobre ella.
- **StatisticalCalculator** : Es una clase que nos permite instanciar una calculadora estadística para realizar el cálculo correspondiente sobre un conjunto de datos dado.
 - putData(datos): Permite ingresar el conjunto de datos en la calculadora.
 - eraseData(): Permite borrar los datos de la calculadora.
 - mean(): Devuelve el cálculo de la media de los datos.
 - standarDeviation(): Devuelve la desviación estándar de los datos.
- **FileHandler** : Esta clase es la encargada del manejo del archivo que contiene los datos.

IV. EVALUACIÓN

A. Pruebas

La pruebas fueron Automatizadas gracias al framework JUnit y integradas gracias al servicio de CircleCi.

Column 1	Column 2
Estimate Proxy Size	Development
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

Test	Expected Value		Actual Value	
	Mean	Std. Dev	Mean	Std. Dev
Column 1	550.6	572.03	550.6	572.03
Column 2	60.32	62.26	60.32	62.26

Fig. 3. Tabla de Pruebas

B. cobertura

Element	Class, %	Method, %	Line, %
⚡ CalculatorException	100% (1/1)	100% (1/1)	100% (2/2)
🔵 StatisticalCalculator	100% (1/1)	100% (5/5)	100% (23/23)

Fig. 4. Tabla de Cobertura del Paquete Calculator





Element	Class, %	Method, %	Line, %
 calculators	100% (2/2)	100% (6/6)	100% (25/25)
 collections	100% (3/3)	80% (12/15)	75% (27/36)
 file_handling	100% (1/1)	100% (3/3)	83% (15/18)
 App	100% (1/1)	100% (2/2)	83% (15/18)

Fig. 5. Tabla de Cobertura de Paquetes

V. CONCLUSIÓN

La calculadora estadística cumple con las funcionalidades especificadas, el uso de las herramientas tecnológicas como Maven y Git, aceleran el proceso de desarrollo, estructurar bien los directorios y fuentes del proyecto, manejo conciso de las versiones y cambios, además de facilitar el acceso a librerías y frameworks externos. Las clases genéricas son una gran opción a la hora de desarrollar colecciones de datos. Las clases anónimas pueden ser una gran solución para evitar clases innecesarias y sin mucho sentido. Como trabajo futuro se espera poder realizar una versión de la calculadora estadística como web app disponible para el público.

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.