

Design OO

Daniel Felipe Gomez Suarez
Escuela Colombiana De Ingenieria Julio Garavito
AREP

August 2020

1. Introduccion

El paradigma de programacion orientada objetos hace un acercamiento a la realidad, una abstraccion en terminos practicos, es decir, obtener las características de un objeto y como interactua con otros dentro de un sistema. mas alla de las características del objeto, se hace una abstraccion del comportamiento del objeto dentro del sistema.

EJEMPLO DE CLASES Y OBJETOS

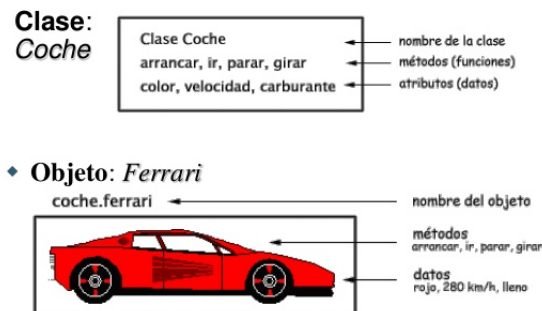


Figura 1: Ejemplo de Clases y Objetos

Un problema desarrollado con técnicas orientadas a objetos requiere, en primer lugar saber cuáles son los objetos del programa. Como tales objetos son instancias de clases, la primera etapa en el desarrollo orientado a objetos requiere de la identificación de dichas clases (atributos y comportamiento), así

como las relaciones entre éstas y su posterior implementación en un lenguaje de programación.

- **Objeto:** un Objeto en programación es una instancia de la clase, es decir un espacio en memoria, Los objetos contienen las características concretas, es decir el color no es general en este caso es rojo, el nombre del objeto es coche ferrari
- **Clase :** Es la abstracción del objeto de la vida real, ese contiene los atributos, que son las características del objeto y métodos que refiere al comportamiento del objeto

2. Desarrollo

El laboratorio consiste en una calculadora estadística con dos funciones básicas, donde los datos son ingresados dado un archivo, el programa guardará los datos en una lista encadenada, es decir cada dato será un nodo de la lista.

este programa calcula la media y la desviación estándar de los datos ingresados por archivo. estos datos deben cargarse a la ruta dispuesta para ello.

- **media=** La media se calcula como un promedio, siendo la suma y la división de la cantidad de datos.

THE FORMULA FOR STANDARD σ

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

Figura 2: Media Estadística

- **Desviación =** la desviación se calcula de acuerdo al valor de la media, esto para saber que tan alejados pueden estar los datos

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

Figura 3: Fórmula Desviación Estándar

2.1. Diseño

La lista encadenada consiste en de forma sencilla enlazar los nodos , como una cola, va agregando los elementos al final, estas conexiones son vertices que permiten enlazar los nodos de las listas encadenadas.

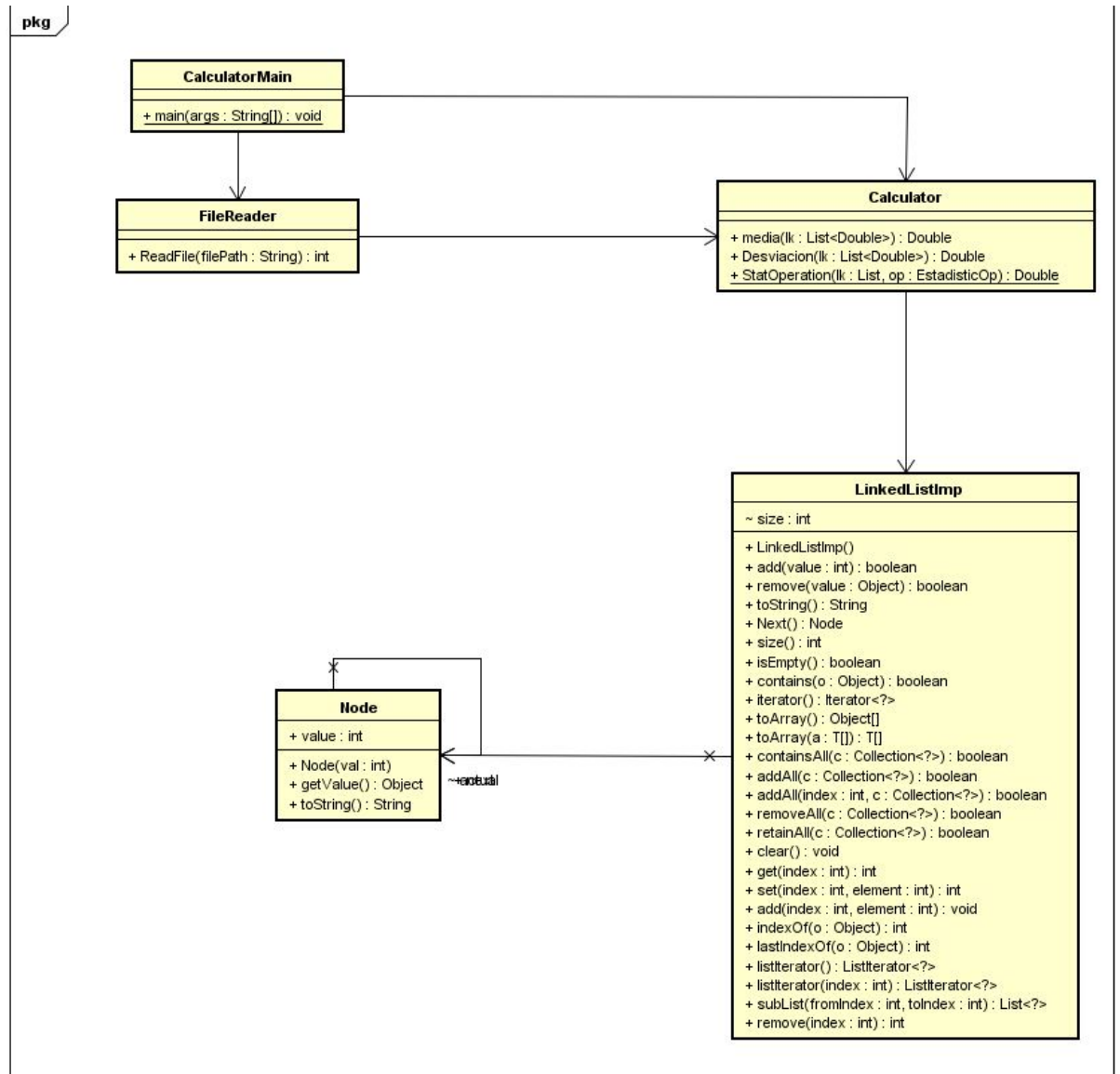


Figura 4: Modelo Linked List

2.2. Elementos y proceso

los elementos del programa son :

- **CalculatorMain** es la clase principal, alli solo se hace la solicitud de las operaciones, recibe el archivo como argumento en la prueba, posteriormente pasa a procesar dato por dato para almacenarlos en la Lista
- **FileReader** esta clase lee linea por linea el archivo el cual contiene los datos para calcular la media y la desviacion, cabe aclarar que se asume la integridad de los datos, ya que deben ser solo datos numericos, pero asi poder hacer los calculos estadisticos.
- **Calculator** esta clase retorna los resultados de los dos calculos de acuerdo a los datos recibidos, esta clase contiene la lista que se implemento para guardar los datos.
- **LinkedListImp** esta clase es la lista la cual consiste en una implementacion de una interfaz "List", y se implementaron los metodos add, remove y el recorrido de la lista.
- **Node** el nodo es la transformacion del dato que ingreso el usuario como objeto, se limito el nodo a tener su valor entero para asi no tener problemas de tipo en la calculadora.

3. Conclusiones

Durante el desarrollo del laboratorio nos dimos cuenta que en esta forma de programar, la tendencia del programador es a realizar metodos los cuales no son propios del comportamiento del objeto, esto forzando comportamientos inadecuados. el anterior problema genera acoplamiento en los metodos y en las clases, causando errores en el funcionamiento del programa. Por esto se debe tener un comportamiento especifico para cada objeto, en este caso el Nodo y La LinkedList debian no compartir comportamientos del otro, ya que lo que se buscaba era implementarla sin usar un contenedor propio de JAVA En cuanto

a eficiencia la mayor parte de los programas son modifcables para mejorar su eficiencia, en cuanto a este caso especifico lo que podria hacerse es agregar un elemento que maneje el ultimo objeto de la lista asi haciendo el metodo ADD mas eficiente sin necesidad de recorrer toda la lista para agregar un dato, reduciendo asi el tiempo de ejecucion de lineal o constante, es decir mejorar su eficiencia

Referencias

- [1] DISEÑO ORIENTADO A OBJETOS
<https://virtual.itca.edu.sv/Mediadores/stis/2diseorientadoaobjetos.html>
2020.
- [2] LINKEDLISTS
<https://www.cs.cmu.edu/adamchik/15-121/lectures/Linked20Lists/linked20lists.html>
2020.
- [3] DESIGN OO
<https://www.tes.com/lessons/gOM9XeejXicEyQ/programacion-orientada-a-objetos-introduccion>
2020.