

Implementación de una Linked list

Introducción a Java, Mvn y GIT

Johan Stiven Guerrero Pineda¹

¹ *Escuela Colombiana de ingeniería Julio Garavito, Bogota D.C, Colombia*

Fecha: 29/11/19

Resumen— Este documento brinda una idea de como fue la implementación de una linked list desde cero, con la especificaciones dadas por las implementaciones abstractas de las clases genéricas de la API de Java.

OBJETIVOS

- Implementar una linked list desde cero, y generar una compatibilidad con las librerías de java alta
- Desarrollar el ejercicio de una calculadora de media y desviación estándar

INTRODUCCIÓN

Java es un lenguaje de programación orientado a objetos, el cual su código es ejecutado por una máquina virtual, la cual se denomina JAVA VIRTUAL MACHINE, esta se encarga de interpretar todas aquellas clases y convertirlas en código de máquina para ser capaz cualquier computador de ejecutar un programa.

El elemento fundamental a la hora de hablar de programación orientada a objetos es el concepto de Objeto, este es común en este lenguaje haciendo que sea una abstracción de un objeto físico de la naturaleza. Estos objetos se pueden modelar mediante clases las cuales nos permiten definir variables y métodos que toman estos objetos.

Uno de los mecanismos mas importantes en este lenguaje se denomina la herencia, este nos permite extender nuevas funcionalidades de clases ya implementadas y finalizadas. Este es uno de los puntos importantes para la implementación de esta laboratorio.

Por otro lado otro termino importante a destacar es la linked list la cual nos permite tener la funcionalidad de una lista, es decir una secuencia de elementos, pero conectados todos ellos entre si de manera circular. Este es el elemento que queremos emprender a implementar, y una apli-

cación. En este documento, comentamos la realización del mismo, por medio del lenguaje de programación de orientado a objetos de Java y su mecanismo de implementación como lo es la herencia.

PROCEDIMIENTO

El primer paso para la implementación de una linked list es determinar su funcionamiento y como actúa con respecto a sus valores. Una linked list es una colección la cual viene implementada en java, esta nos permite una implementación de una cola en Java, las cuales las operaciones de inserción y eliminación son muy eficientes. Estas implementan una clase abstracta para su funcionamiento la cual es `AbstractSequentialList`, esta nos permite generar un comportamiento regular hacia las clases que lo heredan, y he aqui cuando empezamos la implementación de nuestra clase personalizada.

Creamos una clase linked list la cual debería de ser genérica, esto queriendo decir que acepte por dentro los valores que desee, por lo que aceptaría cualquier tipo de objeto. Eso si todos los nodos comparten el mismo tipo. Hablando de los nodos, estos significan como almacenar cada uno de los elementos secuenciados y enlazados en nuestra colección. La creación de los mismos no generaría mas de una clase con dos propiedades importantes anterior y siguiente. Estas propiedades hacen referencia a los nodos vecinos entre ellos. Haciendo asi cada uno de ellos y los nuevos nodos que se creen tengan exclusivamente referencia a un nodo por cada extremo.

Sabiendo la implementación de los nodos, continuemos

con la clase linked list. Esta nos tenia que permitir implementar y realizar cada una de las referencias de estos nodos. Por lo que su implementación debería de lograr esto. Por medio los métodos de push hacemos que un nodo cabeza la cual va a ser el único parámetro de esta clase, refencie vecinos siendo el próximo el siguiente en la lista y en anterior la cabeza de la cola. Es decir el primer elemento de la lista. Con la misma lógica de implementación, este método continuara agregando nuevos elementos a una secuencia de objetos que son referenciados los cuales nos permiten guardar objetos de tipo nodo y a su vez estos guardan el valor específico del tipo de dato que se desea guardar.

Luego de realizar esto la implementación de los demás métodos importantes se hace de la misma manera, haciendo que el código por medio de la implementación de interfaces nos permitan una sencilla, rápida y concisa implementación de una lista enlazada.

A continuación hicimos uso de esta colección personalizada para la resolución de un problema como lo es determinar la media y la varianza de un conjunto de datos. Esta se realizo por medio del la llamada a nuestra ya generada linked list y nos permitió tener un orden entre los elementos y sacarlos en su determinado tiempo para obtener sus valores y generar los resultados a partir de las correspondientes formulas sobre la media y la varianza. Haciendo que el experimento de la implementación de un control genérico sea exitoso y fácil de lograr por medio de la metodología de la programación orientada a objetos como lo es la herencia.

CONCLUSIÓN

- La herencia es un mecanismo muy fuerte el cual nos permite agrandar, etender y permitir trazabilidad a la hora de crear nuevos componentes en un lenguaje orientado a objetos
- Mecanismos como Git y Maven son grandes herramientas a la hora de llevar un desarrollo puesto que nos permiten agilizar esas tareas metódicas las cuales hacen del trabajo de un programador mas largo. El conocimiento de estas logran una mayor eficiencia a la hora de trabajar y además de permitirnoss un trabajo en equipo efectivo

REFERENCIAS

- Class Linked List Documentación, <https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>
- Concepto Herencia, java herencia polimorfismo encapsulamiento
- Git Documentación, <https://git-scm.com>
- Maven Documentación, <http://maven.apache.org>