

Documentación técnica de la aplicación

¿Cual es el objetivo de la aplicación?

Con el desarrollo de esta aplicación lo que buscamos hacer es la elaboración de una estructura de datos como lo es la de árboles binarios en la cual nos basamos para desarrollar el código y el funcionamiento de esta aplicación por lo cual podemos decir que en esta misma se va a poder encontrar la definición de los Nodos y la implementación de funciones como el recorrido en preorden, inorden y postorden los cuales son los diferentes métodos que se pueden observar en el funcionamiento de los árboles binarios.

Explicación de las clases utilizadas en el programa.

Como podemos observar en el código utilizamos 3 clases las cuales son Main, Nodo y ArbolBinario dentro de las cuales su funcionamiento es:

- Main: La clase Main es la clase principal en todo programa, en la cual es esta clase se van a ejecutar todas las acciones que vamos a ejecutar, así que podemos decir que el Main es tanto el punto de entrada como el punto de cierre de nuestra aplicación en el cual vamos a ejecutar todo lo que sea necesario en nuestra aplicación, y en el caso de el desarrollo de mi sistema en el main vamos a encontrar el código con el vamos a darle al usuario un menú indicando los requisitos de la actividad como insertar elementos, imprimir los elementos y eliminar elementos.
- Nodo: En esta clase que ya es creada directamente por nosotros creamos fue la clase Nodo la cual se va a encargar de crear los nodos en los cuales vamos a contener la información de cada valor que se necesita para crear el árbol binario y en donde además vamos a crear el debido constructor para así de esta manera poder darle vida a los valores indicadores que nosotros le pasamos y también le van a dar un valor a los punteros los cuales son de gran importancia para desplazarnos en nuestro árbol

- **ArbolBinario:** En esta clase lo que vamos a hacer es crear todos los métodos que vamos a utilizar para el debido funcionamiento de un árbol binario en la cual incluimos sus funciones de recorrer un árbol con el método Inorden, Postorden y Preorden el las cuales ya más a profundidad le vamos a indicar con código cual es el procedimiento que estas deben de realizar para así de esta manera poder cumplir con el correcto recorrido de cada uno de estos métodos, además en esta clase también vamos a encontrar el debido código para eliminar un nodo con su valor y como los demás nodos deben de reposicionarse.

Explicación de los métodos principales utilizados.

Metodo agregar nodo: En este método lo que estamos haciendo es saber el valor de nuestra raíz para de esta manera empezar a darle sentido a nuestro árbol, además después de esto podemos encontrar diferentes condicionales en los cuales vamos a indicar en qué lugar y posición debería ubicarse el nuevo nodo insertado ya que el sistema de árboles binarios tiene unas reglas las cuales son:

- Cada nodo puede tener como máximo 2 nodos hijos
- El hijo de la izquierda deberá tener un valor menor al de su padre
- El hijo de la derecha deberá tener un valor mayor al de su padre

Así que sabiendo estas condiciones podemos indicar que estas son las mismas condiciones que vamos a encontrar en el método agregar nodo solo que con código Java y de igual manera el código también lo encontramos documentado para un mejor entendimiento.

Método está vacío: Este método es uno de los más sencillos de entender ya que este básicamente va a evaluar el número que tenemos en nuestro nodo raíz que es por donde empieza todo el árbol y si este nos otorga un valor de null significa que no hemos adicionado ningún valor y que no hay nada por mostrar

Método Inorden: Este método en el código lo que va a hacer es representar un tipo de recorrido de árbol binario el cual su nombre es el mismo del método (Inorden) en el cual el funcionamiento de este método se rige bajo unas reglas las cuales son:

- Visita el subárbol izquierdo o también conocido como el nodo hijo izquierdo en inorden
- Visitar el nodo raíz.
- Visita el subárbol derecho o también conocido como el nodo hijo derecho en inorden

Así que en el método vamos a encontrar estas mismas reglas para hacer el debido proceso de recorrido inorden

Método preorden: Este método lo que va a hacer al igual que el pasado es hacer el recorrido que le corresponde a este pero con unas diferentes reglas de recorrido las cuales son:

- Visitar el nodo raíz.
- Visita el subárbol izquierdo o también conocido como el nodo hijo izquierdo en preorden.
- Visita el subárbol derecho o también conocido como el nodo hijo derecho en preorden.

Método postorden: El método postorden tiene la misma funcionalidad de los 2 pasados el cual es realizar un tipo de recorrido al árbol binario cumpliendo con ciertas especificaciones las cuales son:

- Visita el subárbol izquierdo o también conocido como el nodo hijo izquierdo en postorden.
- Visita el subárbol derecho o también conocido como el nodo hijo derecho en postorden.
- Por último visitar el nodo raíz.

Método eliminar: En el método eliminar que encontramos en el código lo que vamos a hacer es eliminar un nodo que nosotros le indicamos, pero al hablar de la eliminación de árboles estos deben de hacer un recorrido para encontrar el valor y así luego eliminarlo y como lo indicamos en el código lo primero que vamos a hacer es:

1. Indicar que mientras el dato que le indicamos a eliminar sea menor al dato auxiliar que hace referencia al dato raíz el recorrido tendrá que ir por el nodo de la izquierda
2. Si al evaluar el dato por la izquierda nos arroja un resultado falso vamos a evaluar el número que tenemos que eliminar pero esta vez por los nodos de la derecha.
3. Si el nodo eliminado tiene 2 hijos vamos a subir como nodo padre al dato que ocupa la posición más próxima.

Sin embargo en el código se encuentra un poco más especificado el paso a paso con la debida documentación dentro de este para entender a mayor profundidad lo que se hace dentro de cada condicional.

Otro caso que podríamos indicar es cuando es un nodo hoja y básicamente un nodo hoja es o son los nodos que no tienen hijos y estos los vamos a encontrar en los extremos de las estructuras.

Y en el código lo que se hace es evaluar si nos encontramos ya en los nodos hoja o simplemente tenemos el nodo que se encuentra en la raíz.

Método obtener nodo reemplazo: Este método es de los más importantes a la hora de eliminar un Nodo ya que obviamente después de eliminar un nodo tenemos que reorganizar la posición de los nodos para seguir cumpliendo con las condiciones principales de los árboles binarios, así que el debido proceso de este método es:

- Cuando hablamos de eliminar los nodos hoja no vamos a tener ningún tipo de inconveniente ya que el nodo hoja como lo indicamos anteriormente es de los últimos nodo de nuestro árbol
- Y cuando vamos a eliminar un nodo que tiene hijos debemos de indicar como raíz del nodo que indicamos al número más próximo del nodo que indicamos