



TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

ING. SISTEMAS COMPUTACIONALES

Lenguajes y autómatas II

Proyecto 1

ALUMNO: Salvador Eliud Carranza de la Rosa

NO. CONTROL: 15480101

CATEDRÁTICO: Juan Pablo Rosas Baldazo

Introducción:

En el siguiente reporte se describe lo realizado para la realización de la aplicación que se pidió como proyecto de la unidad 1 de la materia Lenguajes y autómatas II.

El presente proyecto consiste en realizar el código de un programa en algún lenguaje de programación el cual realizara un árbol binario, el programa a realizar debería de tener una interfaz en la cual se pudiera insertar cualquier expresión aritmética posteriormente ordenar los datos insertados, crear un árbol binario con dichos datos y mostrarlo gráficamente en pantalla.

Para la realización del proyecto se utilizó el Lenguaje de Programación Java su usó el IDE NetBeans, la razón por la que se llegó a la conclusión de utilizar este lenguaje es porque es con el que estoy más familiarizado ya que es este el que hemos venido manejando durante toda la carrera, además que ya tenía un programa parecido y me base de el para la realización de este, este se complementó con un código que está internet (Youtube).

Descripción:

El programa realizado consta de tres clases que son las que tienen los métodos para el funcionamiento de la aplicación, una clase ArbolBin, la clase Nodo y la clase Prueba, en la clase Arbol Bin se le incluyen los métodos que permiten realizar las operaciones como insertar, imprimir en los diferentes órdenes, recorrer a la izquierda, así como borrar los nodos.

La clase Nodo es la que se encarga de leer los valores y asignarlos a las hojas del árbol correspondientes también define cuales son los nodos padres y cuales los hijos o nodos u subnodos.

Y por último la clase Prueba que es la clase main, esta permite que se ejecute el programa, manda los mensajes a pantalla y lee los datos ingresados desde el teclado así como también muestra el resultado.

Pseudocodigo:

Se toma el dato a ingresar X

Partiendo de la raíz preguntamos: $\text{Nodo} == \text{null}$ (o no existe) ?

En caso afirmativo X pasa a ocupar el lugar del nodo y ya hemos ingresado nuestro primer-dato.

En caso negativo preguntamos: $X < \text{Nodo}$

En caso de ser menor pasamos al Nodo de la IZQUIERDA del que acabamos de preguntar y repetimos desde el paso 2 partiendo del Nodo al que acabamos de visitar

En caso de ser mayor pasamos al Nodo de la DERECHA y tal cual hicimos con el caso anterior repetimos desde el paso 2 partiendo de este nuevo Nodo.

Resultados:

Una vez teniendo el código se procedió a la prueba de este, al correr el programa todo va bien se abre la ventana, le damos clic en insertar, ponemos la expresión y nos aparece el árbol gráficamente y acomodado, pero lamentablemente querer hacer que nos aparezcan los datos en los órdenes PreOrden, InOrden, PostOrden, etc, no es posible, ya que hay algún error y no aparece nada en el campo donde se debería imprimir, error el cual por más que intentamos no pudimos solucionar, además que el código esta muy complicado es por esto que decidimos quedarnos con uno mas austero, este solo acepta datos del tipo int ya que no pudimos corregirlo para que aceptara los datos String, pero este si muestra los resultados en el orden que se planteo.

Conclusiones:

Llegamos a la conclusión de los arboles binarios sirven para organizar datos para facilitar su manipulación, ya sea el ingreso, borrado o búsqueda de datos, y precisamente una de las principales ventajas de los árboles binarios es la búsqueda, ya que como en muchos algoritmos de búsqueda necesitamos tener la información ordenada y en los árboles binarios precisamente los datos se van ingresando de forma ordenada.

Este Proyecto me resultó complicado ya que por más que investigué y vi videos sobre este tema no logré hacer que el programa funcionara correctamente.

Referencias:

Programas realizados en clases anteriores

Video tutoriales: <https://www.youtube.com/watch?v=rNgVGkXfNFQ>
Y <https://www.youtube.com/watch?v=6DKyKUg9T-8>