



INSTITUTO TECNOLÓGICO DE NUEVO LEÓN

Ingeniería en sistemas computacionales

Lenguaje y Autómatas 2

Proyecto 1

Catedrático

Juan Pablo Rosas Baldazo

Alumno

Axel Johnary Liñan Estrada

15/02/2018

Introducción

En este proyecto se trató de hacer un árbol de expresiones y que se impriman los recorridos preOrden, inOrden y postOrden. Pero no pude hacer que se formara el árbol cuando corriera el programa, así que el árbol se creara en el Main y los recorridos si se imprimirán. Peor es nada.

La cadena a resolver será: $3 + (8 * 5) + e/2$

Pseudocodigo

Clase Nodo.java

```
Clase Nodo {
    Se crea una cadena llamada "dato"
    Se crea un nodo izquierdo y uno derecho

    El nodo siempre debe tener un dato en sus parámetros

    obtenerNodoIzquierdo() {
        return izq;
    }

    obtenerNodoDerecho() {
        return der;
    }

    asignarNodoIzquierdo(Nodo nodo) {
        izq = nodo;
    }

    asignarNodoDerecho(Nodo nodo) {
        der = nodo;
    }

    obtenerDato() {
        return dato;
    }
}
```

Clase Arbol.java

```
Clase Arbol {

    Método principal{

        Creamos los nodos con operadores.
        La raíz es +
        El nodo 2,3 y 4 serán +, / y *

        En la raíz su nodo izquierdo es un "3"
        En la raíz su nodo derecho es el nodo 2
        En el nodo 2 su nodo derecho es una "e"
        En el nodo 2 su nodo izquierdo es el nodo 3
        En el nodo 3 su nodo derecho es un "2"
        En el nodo 3 su nodo izquierdo es el nodo 4
        En el nodo 4 su nodo derecho es una "8"
        En el nodo 4 su nodo izquierdo es una "5"

        Imprime PreOrden
        Imprime PostOrden
        Imprime InOrden
    }

    PreOrden {
        Si es diferente de nulo {
            Imprime el dato de la raíz
            Obtén su nodo izquierdo y asigna el valor a la raíz
            Obtén su nodo derecho y asigna el valor a la raíz        }
        }

    PostOrden {
        Si es diferente de nulo {
            Obtén su nodo izquierdo y asigna el valor a la raíz
            Obtén su nodo derecho y asigna el valor a la raíz
            Imprime el dato de la raíz
        }
    }

    InOrden {
        Si es diferente de nulo {
            Obtén su nodo izquierdo y asigna el valor a la raíz
            Imprime el dato de la raíz
            Obtén su nodo derecho y asigna el valor a la raíz
        }
    }
}
```

Resultado

Los recorridos del árbol resultaron los siguientes:

- PreOrden: $+3+/*852e$
- PostOrden: $385*2/e++$
- inOrden: $3+8*5/2+e$

Conclusión

Si funcionaria como se esperaba resolvería cadenas de operaciones largas para saber la jerarquía en la que se debe resolver. En el que hice solo ayuda para saber los recorridos de preOrden, postOrden e inOrden.

Referencias

El único tutorial que encontré útil fue el siguiente, en ese me base mi código. El árbol es diferente, lo hice a mano en la libreta luego lo codifique para ver si estaba correcto.

https://www.youtube.com/watch?v=h_8WTkH1OV4