



Universidad Mariano Gálvez de Guatemala
Facultad de Ingeniería
Ingeniería en Sistemas
Curso: Programación III

TAREA:

documentación Proyecto 1

Autor: Wilmer Stuardo Jiménez García
Carné: 7690-22-9822
Guatemala, marzo de 2024

Documentación del Proyecto

Descripción

Este proyecto implementa un sistema basado en árboles de expresión y árboles de búsqueda. Proporciona un menú de opciones para manipular expresiones aritméticas y realizar recorridos en los árboles resultantes.

Funcionalidades

- ****Ingresar Expresión Aritmética****: Permite al usuario ingresar una expresión aritmética y la almacena en memoria.
- ****Validar Expresión y Sustituir Variables****: Valida si existen variables en la expresión y, de ser afirmativo, solicita al usuario el valor de cada variable y sustituye estas en la expresión, creando una nueva expresión.
- ****Realizar Recorridos****: Ofrece opciones para realizar recorridos en el árbol resultante, incluyendo pre-orden, in-orden y post-orden.
- ****Finalizar Programa****: Permite al usuario salir del programa.

Estructura del Main

El archivo principal ("Proyecto1.java") contiene la lógica principal del programa, que se ejecuta en un bucle mientras el usuario no decida salir. Aquí se resumen las principales secciones del archivo:

Instanciación de Objetos

- "Scanner entrada": Se crea un objeto Scanner para la entrada de datos desde el teclado.
- "Ingreso_Datos ingreso_datos": Se instancia un objeto de la clase "Ingreso_Datos" para manejar el ingreso de datos y la validación de expresiones.
- "Arbol arbol": Se crea un objeto de la clase "Arbol" para manipular los árboles de expresión.

Bucle Principal

Se inicia un bucle “while” que se ejecuta hasta que el usuario decida salir (“salir = true”). Dentro de este bucle, se presenta un menú de opciones y se ejecuta la lógica correspondiente a la opción seleccionada por el usuario.

Opciones del Menú

1. ****Ingresar Expresión Aritmética****: Se solicita al usuario ingresar una expresión aritmética y se valida su entrada.
2. ****Realizar Recorrido Pre-orden****: Se realiza un recorrido pre-orden en el árbol de expresión y se muestra por pantalla.
3. ****Realizar Recorrido In-Orden****: Se realiza un recorrido in-orden en el árbol de expresión y se muestra por pantalla.
4. ****Realizar Recorrido Post-Orden****: Se realiza un recorrido post-orden en el árbol de expresión y se muestra por pantalla.
5. ****Salir del Programa****: Se finaliza el bucle y se termina la ejecución del programa.

Clase “Ingreso_Datos “

La clase “Ingreso_Datos” se encarga de gestionar la entrada de datos desde el usuario y realizar la validación de expresiones ingresadas.

Constructor

- “public Ingreso_Datos()”: Constructor de la clase que inicializa un objeto Scanner para la entrada de datos.

Método “imprimir()”

- “public int imprimir()”: Presenta un menú de opciones al usuario y retorna la opción seleccionada.

Método “Opcion1()”

- “public String Opcion1()”: Solicita al usuario ingresar una expresión aritmética y la devuelve como una cadena de texto.

Método "validacion(String expresion) "

- "public String validacion(String expresion) ": Valida si la expresión contiene variables y, de ser así, solicita al usuario el valor de cada variable y sustituye estas en la expresión, devolviendo la nueva expresión válida.

Método "construirArbol(String expresion) "

- "public Arbol construirArbol(String expresion) ": Construye un árbol a partir de una expresión dada, donde cada letra representa un nodo del árbol.

Clase "Nodo"

La clase "Nodo" representa un nodo en el árbol de expresión.

Atributos

- "private String dato": Almacena el dato (valor) del nodo.
- "private Nodo izquierda, derecha": Referencias al hijo izquierdo y derecho del nodo.

Constructor

- "public Nodo(String dato) ": Constructor de la clase que inicializa un nodo con el dato proporcionado y sin hijos.

Métodos Getters y Setters

- Métodos para acceder y modificar los atributos "dato", "izquierda" y "derecha".

Método "imprimirDato()"

- "public void imprimirDato()": Imprime el dato almacenado en el nodo.

Clase "Arbol"

La clase "Arbol" representa un árbol de búsqueda binario.

Atributos

- "private Nodo raiz": Referencia a la raíz del árbol.

Constructor

- "public Arbol()": Constructor de la clase que inicializa el árbol sin raíz.

Métodos

- "public boolean existe(String busqueda)": Verifica si un dato dado está presente en el árbol.
- "public void insertar(String dato)": Inserta un nuevo dato en el árbol.
- "private void insertar(Nodo padre, String dato)": Método auxiliar para insertar un dato en el árbol.
- "private void preorden(Nodo n)": Realiza un recorrido pre-orden en el árbol y muestra los datos.
- "private void inorden(Nodo n)": Realiza un recorrido in-orden en el árbol y muestra los datos.
- "private void postorden(Nodo n)": Realiza un recorrido post-orden en el árbol y muestra los datos.
- "public void preorden()": Método público para iniciar un recorrido pre-orden.
- "public void inorden()": Método público para iniciar un recorrido in-orden.
- "public void postorden()": Método público para iniciar un recorrido post-orden.