Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

**Звіт**

з лабораторної роботи № 6 з дисципліни

Алгоритми та структури даних 2. Структури даних»

«Деревовидні структури даних»

Варіант **3**

**Виконав студент:** ІП-з21 Гавриленко Даяна Юріївна

**Перевірив:** Халус Олена Андріївна

Київ 2023

**6.1 Мета роботи**  
 Мета роботи – вивчити основні підходи формалізації та імплементації алгоритмів побудови та обробки базових деревовидних структур даних.

**6.2 Постановка задачі**

Изображение выглядит как текст, Шрифт, белый, алгебра

Автоматически созданное описание

**6.3 Псевдокод алгоритму** Програма

Отримати 4 числа від користувача для формули (a\*(b+c))/d

Створити FormulaTree з '/' як кореневим значенням

Вставити значення '\*', 'd', '+', 'a', 'b' та 'c' у дерево в такому порядку

Надрукувати дерево за допомогою методу LevelTraverse класу FormulaTree

FormulaTree

Властивості:

Корінь: Вузол

Методи:

Вставити(значення)

Вставити значення у дерево за допомогою методу InsertHelper

LevelTraverse()

Виконати обхід дерева по рівнях та повернути рядкове представлення дерева

InsertHelper(значення, корінь)

Якщо правий нащадок кореня порожній

Створити новий вузол зі значенням та призначити його правим нащадком кореня

Інакше, якщо лівий нащадок кореня порожній

Створити новий вузол зі значенням та призначити його лівим нащадком кореня

Інакше

Рекурсивно викликати InsertHelper для лівого нащадка кореня

LevelTraverseHelper(корінь, результат)

Якщо лівий нащадок кореня не порожній

Додати лівого нащадка до результату

Якщо правий нащадок кореня не порожній

Додати правого нащадка до результату

Якщо лівий нащадок кореня не порожній

Рекурсивно викликати LevelTraverseHelper для лівого нащадка кореня

Повернути результат

Вузол

Властивості:

Значення: об'єкт

Лівий: Вузол

Правий: Вузол

**6.4 Програмна реалізація алгоритму**

**6.4.1 Вихідний код на мові C#**

using System.Text;  
  
namespace Lab\_3;  
  
public class Program  
{  
 static void Main(string[] args)  
 {  
 int[] nums = GetConsoleInput();  
 FormulaTree tree = new FormulaTree('/');  
 object[] values = { nums[3], '\*', nums[0], '+', nums[2], nums[1] };  
 foreach (object value in values)  
 {  
 tree.Insert(value);  
 }  
 Console.WriteLine($"\nTree:\n{tree.LevelTraverse()}");  
 Console.ReadLine();  
 }  
  
 static int[] GetConsoleInput()  
 {  
 Console.WriteLine("Enter 4 numbers in formula (a\*(b+c))/d:");  
 int[] nums = new int[4];  
 for (int i = 0; i < nums.Length; i++)  
 {  
 string? input = Console.ReadLine();  
 while (!int.TryParse(input, out \_))  
 {  
 Console.WriteLine("Wrong input format");  
 input = Console.ReadLine();  
 }  
 nums[i] = int.Parse(input);  
 }  
 return nums;  
 }  
}  
  
public class FormulaTree  
{  
 public Node Root { get; private set; }  
  
 public FormulaTree(object value) => Root = new Node(value);  
  
 public FormulaTree(Node node) => Root = node;  
  
 public void Insert(object value) => InsertHelper(value, Root);  
  
 public void Insert(Node node) => InsertHelper(node, Root);  
  
 public string LevelTraverse() => LevelTraverseHelper(Root, new StringBuilder(Root.ToString().PadLeft(5) + "\n"));  
  
 private void InsertHelper(object value, Node root)  
 {  
 if (root.Right == null)  
 root.Right = new Node(value);  
 else if (root.Left == null)  
 root.Left = new Node(value);  
 else  
 InsertHelper(value, root.Left);  
 }  
  
 private void InsertHelper(Node node, Node root)  
 {  
 if (root.Right == null)  
 root.Right = node;  
 else if (root.Left == null)  
 root.Left = node;  
 else  
 InsertHelper(node, root.Left);  
 }  
  
 private string LevelTraverseHelper(Node root, StringBuilder result)  
 {  
 if (root.Left != null)  
 result.Append(root.Left);  
 if (root.Right != null)  
 result.Append(root.Right + "\n");  
 if (root.Left != null)  
 LevelTraverseHelper(root.Left, result);  
 return result.ToString();  
 }  
}  
  
public class Node  
{  
 public Node(object value)  
 {  
 Value = value;  
 Left = Right = null;  
 }  
  
 public object Value { get; private set; }  
  
 public Node? Left { get; set; }  
  
 public Node? Right { get; set; }  
  
 public override string ToString() => $"[{Value}]";  
}

**6.4.2 Приклад роботи**

На рисунках 6.1 і 6.2 показані приклади роботи програми:

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рис.6.1 Приклад роботи програми

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рис.6.2 Приклад роботи програми

**6.5 Висновок**

Під час виконання лабораторної роботи на тему "Деревовидні структури даних" я отримала цінний досвід роботи з побудовою та використанням деревовидних структур для відображення складних формул. Застосувавши цей підхід до побудови дерева формули "(a\*(b+с))/d", змогла на практиці навчитися працювати з вузлами, коренями та листками дерева, а також вивести його графічне представлення. Ця лабораторна робота дала мені змогу краще зрозуміти принципи роботи деревовидних структур та їх застосування в реальних завданнях.