Министерство образования Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий

Кафедра технологий программирования

**Отчёт по лабораторной работе № 4 по курсу «Программирование сетевых приложений»»**

Выполнил: Шиковец Е.А.

Группа 21-ИТ-1

Проверила: преподаватель

Виноградова А.Д.

Полоцк 2022

**ЦЕЛЬ РАБОТЫ:** Деревья. Помеченное дерево. Дерево выражений. Обход дерева.

**ТЕОРИТИЧЕСКИЕ СВЕДЕНИЯ**

1. **Определение понятия дерево.**

Дерево – это нелинейная иерархическая структура данных.

1. **Из каких частей состоит дерево?**

Она состоит из узлов и ребер, которые соединяют узлы.

1. **Определение понятия лес.**

Лес — множество непересекающихся деревьев. Например, если «срезать» корень, получится лес.

1. **Определение понятия высота дерева.**

Высота дерева — это высота корневого узла или глубина самого

глубокого узла.

1. **Определение понятия обход дерева.**

В программировании часто возникает потребность перебрать один за другим все узлы некоторого дерева и предпринять определенные действия для каждого из них. Такой перебор и называется обходом дерева.

1. **Перечислите способы обхода дерева в глубину.**

Существуют различные варианты обхода в глубину. Рассмотрим некоторые из них. Предположим, что у нас есть двоичное дерево.

Вариант 1.

Обойти левое поддерево

Посетить корень

Обойти правое поддерево

Вариант 2.

Посетить корень

Обойти левое поддерево

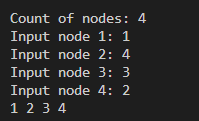
Обойти правое поддерево

**ОПИСНИЕ ПРОДЕЛАНОЙ РАБОТЫ:**

1.Создал проект в MCVST;

2. Описал необходимые действия;

**РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ:**

****

**ЛИСТИНГ ПРИЛОЖЕНИЯ:**

#include <iostream>

using namespace std;

struct tnode

{

    int data;

    struct tnode \*left;  // левое поддерево

    struct tnode \*right; // правое поддерево

};

// вывод в сортированном виде

void treeprint(tnode \*tree)

{

    if (tree != NULL)

    {

        treeprint(tree->left);     // вывод левого поддерева

        cout << tree->data << " "; // вывод кореня дерева

        treeprint(tree->right);    // вывод правого поддерева

    }

}

// добавление узлов в дерево

struct tnode \*addnode(int x, tnode \*tree)

{

    if (tree == NULL) // создание корня

    {

        tree = new tnode;

        tree->data = x;

        tree->left = NULL;

        tree->right = NULL;

    }

    else                                           // если корень есть

        if (x < tree->data)                        // x меньше корневого - уходим влево

            tree->left = addnode(x, tree->left);   // добавление элемента рекурсией

        else                                       // во всех остальных случаях - уходим вправо

            tree->right = addnode(x, tree->right); // тоже добавление элемента рекурсией

    return (tree);

}

// освобождение памяти дерева

void cleartrea(tnode \*tree)

{

    if (tree != NULL)

    {

        cleartrea(tree->left); // удаление левой и правой ветки с помощью рекурсии

        cleartrea(tree->right);

        delete tree; // удаление корня

    }

}

int main()

{

    struct tnode \*root = 0;

    system("CLS");

    int node, n;

    cout << "Count of nodes: ";

    cin >> n;

    for (int i = 0; i < n; i++)

    {

        cout << "Input node " << i + 1 << ": ";

        cin >> node;

        root = addnode(node, root); // добавление узла в дерево

    }

    treeprint(root); // вывод элементов дерева в отсортированном виде

    cleartrea(root); // очистка памяти

    return 0;

}

**ВЫВОД:** В результате лабораторной работы я ознакомился с понятиями «дерево», «помеченное дерево», «дерево выражение», «обход дерева» и основными алгоритмами их реализации и обработки, научился применять полученные знания на практике.