

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 № 5
з дисципліни «Основи програмування –
2. Методології програмування»

« Дерева»

Варіант 11

Виконав студент ПІ-13 Дем'янчук Олександр Петрович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота 5

Варіант 11

Мета роботи - вивчити особливості організації та обробки дерев.

Завдання: Побудувати дерево, що відображає формулу $(a+b)*c-d$, де коренем дерева та його підкореннями є операції "+, -, *, /", а листками є змінні a, b, c, d.

Вивести значення дерева-формули.

Надрукувати відповідні піддерева $y1=a+b$, $y2=y1*c$, $y3=y2-d$.

Код програми

C++

main.cpp

```
#include "func.h"

int main() {
    int num[4];
    cout << "ok, throw me some numbers: \n";
    for (int i = 0; i < 4; i++) {
        cout << (char)(97 + i) << ": ";
        cin >> num[i];
    }
    Node* root = 0;
    buildTree(root, num);
    vector<int> base;
    cout << "tree value is " << root->value << "\n";
    string y3 = "y3=y2-d";
    printTree(root, y3);
    string y2 = "y2=y1*c";
    printTree(root->left, y2);
    string y1 = "a+b";
    printTree(root->left->left, y1);
}
```

func.h

```
#pragma once
#include <iostream>
#include <vector>
#include <string>
using namespace std;

struct Node {
    int value;
    Node* left;
    Node* right;
};

void buildTree(Node*&, int*);
void addNode(Node*&, int v = 0);
void addRight(Node*& node, int v = 0);
void arithmeticNode(Node*&, int, char);
```

```
void printNode(Node* root, int space);
void printTree(Node* root, string formula);
```

func.cpp

```
#include "func.h"

void buildTree(Node*& y3, int* num) {
    for (int i = 0; i < 4; i++) {
        addNode(y3);
    }
    y3->left->left->left->value = num[0];
    y3->left->left->left->right = 0;
    arithmeticNode(y3->left->left, num[1], '+');
    arithmeticNode(y3->left, num[2], '*');
    arithmeticNode(y3, num[3], '-');
}

void addNode(Node*& node, int v) {
    int leaves = 0;
    if (!node)
    {
        node = new Node;
        node->value = v;
        node->left = 0;
        addRight(node->right);
        return;
    }
    else
    {
        addNode(node->left, v);
    }
}

void addRight(Node*& node, int v) {
    node = new Node;
    node->value = v;
    node->left = 0;
    node->right = 0;
}

void arithmeticNode(Node*& node, int num, char oper) {
    node->right->value = num;
    switch (oper) {
        case('+'):
            node->value = node->left->value + node->right->value;
            break;
        case('-'):
            node->value = node->left->value - node->right->value;
            break;
        case('*'):
            node->value = node->left->value * node->right->value;
            break;
        default:
            node->value = node->left->value / node->right->value;
            break;
    }
}
```

```

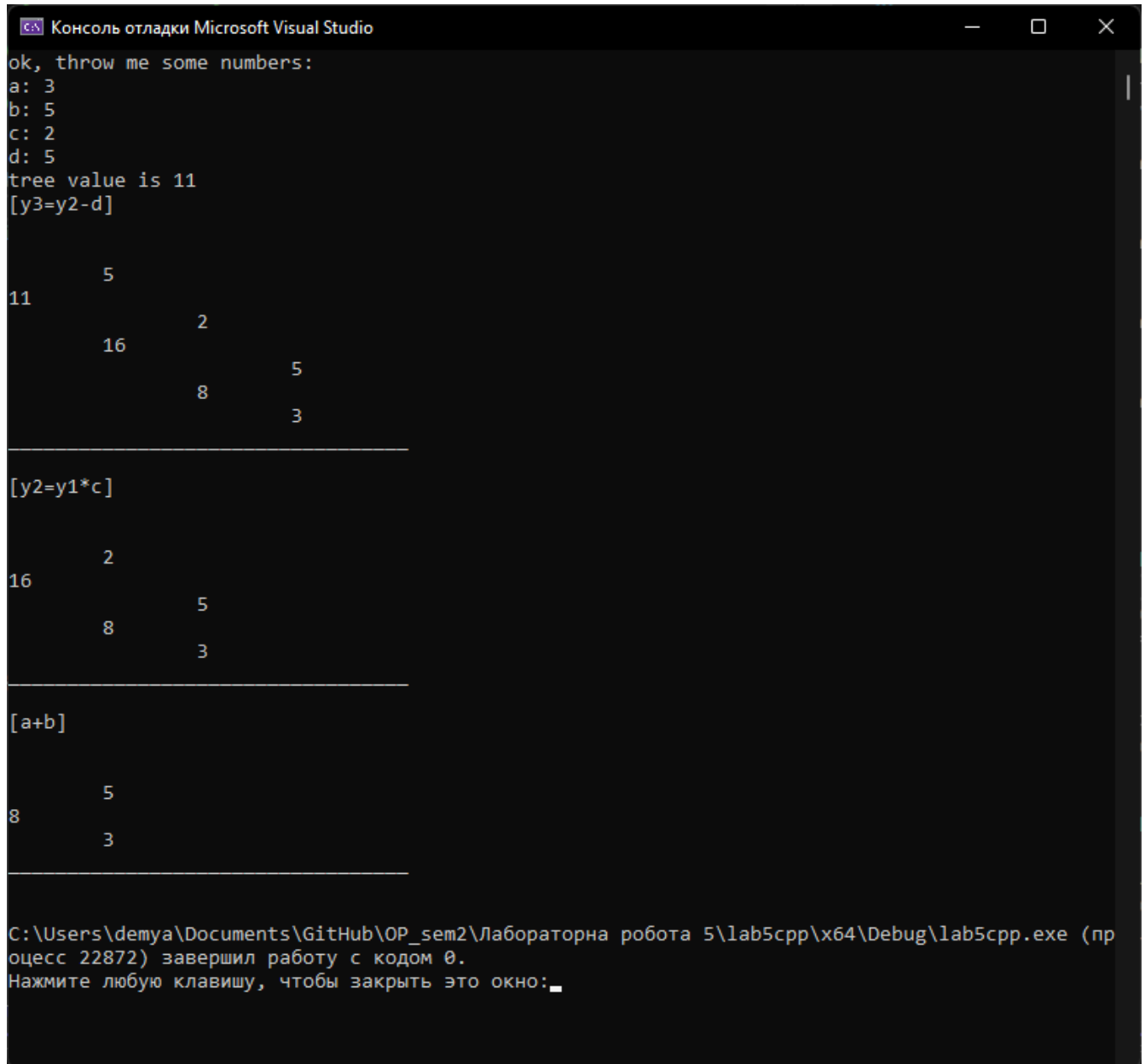
void printNode(Node* root, int space) {
    if (root == NULL) return;
    printNode(root->right, ++space);
    for (int i = 1; i < space; i++) {
        cout << '\t';
    }
    cout << root->value << endl;
    printNode(root->left, space);
}

void printTree(Node* root, string formula) {
    cout << "[" << formula << "]" \n";
    cout << endl << endl;
    printNode(root, 0);
    cout << "----- \n \n";
}

```

Тестування

C++



```
Консоль отладки Microsoft Visual Studio
ok, throw me some numbers:
a: 3
b: 5
c: 2
d: 5
tree value is 11
[y3=y2-d]

      5
11
      2
      16
          8
          5
          3

[y2=y1*c]

      2
16
      5
      8
      3

[a+b]

      5
8
      3

C:\Users\demya\Documents\GitHub\OP_sem2\Лабораторна робота 5\lab5cpp\x64\Debug\lab5cpp.exe (процесс 22872) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно: _
```

Висновки:

На роботі вивчив і використав особливості організації та обробки дерев, навчився заповнювати гілки дерева значеннями, обходити дерево і виводити піддерева.