

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 «Основи програмування – 2.
Метидології програмування»

«Дерева»

Варіант 24

Виконав студент ПІ-13 Нещерет Віталій Олександрович
(шифр, прізвище, ім'я, по батькові)

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

Лабораторна робота 6

Дерева

Варіант 24

24. Побудувати дерево, елементами якого є дійсні числа. Поміняти місцями найбільше та найменше значення дерева.

Код програми

C++

lab_cpp_6.cpp

```
#include "Tree.h"

int main()
{
    Branch* root = NULL;
    int size;
    double element;
    cout << "Enter the number of elements in tree: ";
    cin >> size;
    cout << "Enter tree elements: ";
    for (int i = 0; i < size; i++)
    {
        cin >> element;
        addElement(root, element);
    }
    printTree(root, 0);

    swapMaxAndMin(root);
    cout << "Tree with swaped elements: " << endl;
    printTree(root, 0);
    return 0;
}
```

Tree.h

```
#pragma once
#include <iostream>

using namespace std;

struct Branch {
    double data;
    Branch* leftBranch;
    Branch* rightBranch;
};

void addElement(Branch*&, double);
void printTree(Branch*, int);
void swapMaxAndMin(Branch*&);
double getMax(Branch*);
double getMin(Branch*);
void setOnMaxPos(Branch*&, double);
void setOnMinPos(Branch*&, double);
```

Tree.cpp

```
#include "Tree.h"

void addElement(Branch*& branch, double element) {
    if (!branch) {
        branch = new Branch;
        branch->leftBranch = NULL;
        branch->rightBranch = NULL;
        branch->data = element;
        return;
    }

    if (element >= branch->data) {
        addElement(branch->rightBranch, element);
    }
    else {
        addElement(branch->leftBranch, element);
    }
}

void printTree(Branch* branch, int tabs) {
    if (!branch) {
        return;
    }
    printTree(branch->rightBranch, ++tabs);

    for (int i = 1; i < tabs; i++) {
        cout << "\t";
    }
    cout << branch->data << endl;

    printTree(branch->leftBranch, tabs);
}

void swapMaxAndMin(Branch*& branch) {
    if (!branch) {
        return;
    }
    double max = getMax(branch);
    double min = getMin(branch);
    setOnMaxPos(branch, min);
    setOnMinPos(branch, max);
}

double getMin(Branch* branch) {
    if (branch->leftBranch != NULL) {
        return getMin(branch->leftBranch);
    }
    else {
        return branch->data;
    }
}

double getMax(Branch* branch) {
    if (branch->rightBranch != NULL) {
        return getMax(branch->rightBranch);
    }
    else {
        return branch->data;
    }
}
```

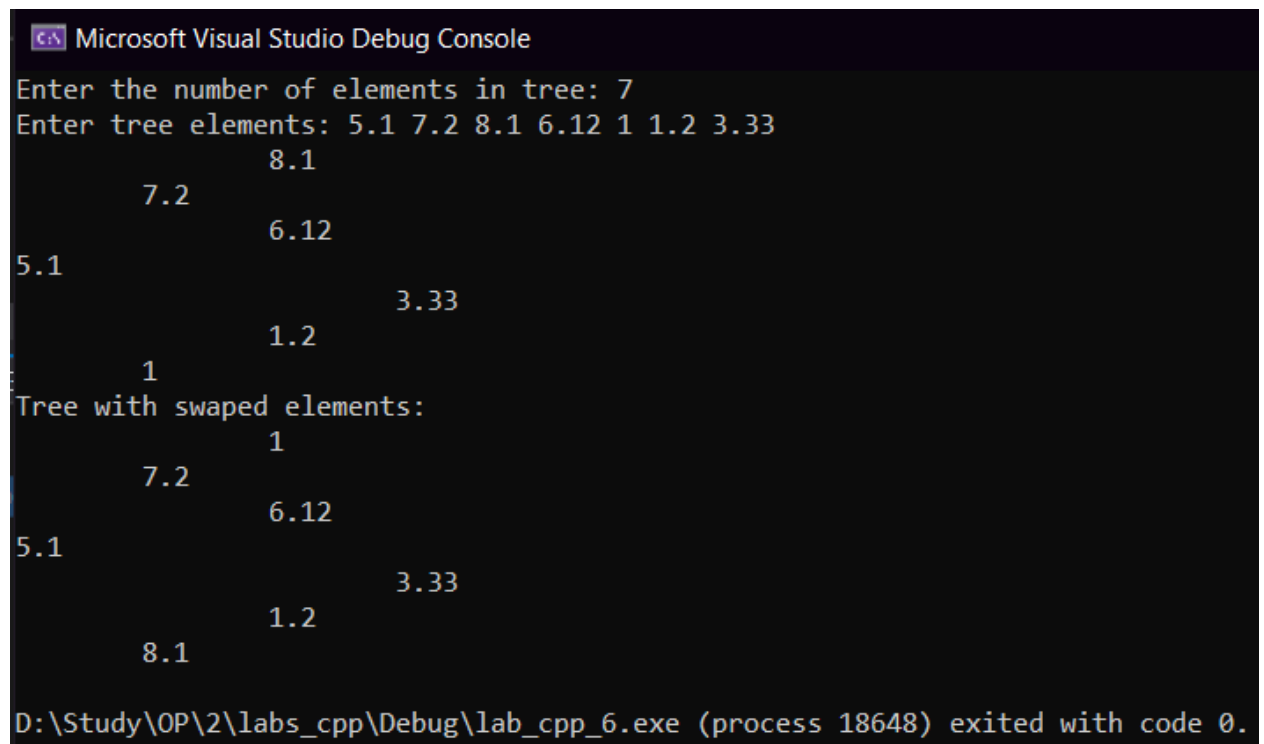
```

void setOnMaxPos(Branch*& branch, double val) {
    if (branch->rightBranch != NULL) {
        setOnMaxPos(branch->rightBranch, val);
    }
    else {
        branch->data = val;
        return;
    }
}

void setOnMinPos(Branch*& branch, double val) {
    if (branch->leftBranch != NULL) {
        setOnMinPos(branch->leftBranch, val);
    }
    else {
        branch->data = val;
        return;
    }
}

```

Тестування:



```

Microsoft Visual Studio Debug Console
Enter the number of elements in tree: 7
Enter tree elements: 5.1 7.2 8.1 6.12 1 1.2 3.33
      8.1
    7.2
  6.12
5.1      3.33
      1.2
    1
Tree with swaped elements:
      1
    7.2
  6.12
5.1      3.33
      1.2
    8.1
D:\Study\OP\2\labs_cpp\Debug\lab_cpp_6.exe (process 18648) exited with code 0.

```

Висновки:

Я вивчив особливості організації та обробки дерев. Застосував ці навички на практиці.