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. Модульне програмування»

«Дерева»

Варіант 24

Виконав студент ІП-11 Печковський Олександр Костянтинович

(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

(прізвище, ім'я, по батькові)

Київ 2022

**Умова задачі:**

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

**Код на С++:**

**Lab6C++.cpp**

#include "Header.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);

}

TreeOutput(root, 0);

SwapMaxAndMin(root);

cout << "Tree with swaped elements: " << endl;

TreeOutput(root, 0);

}

**Header.h**

#pragma once

#pragma once

#include <iostream>

using namespace std;

struct Branch {

double data;

Branch\* leftBranch;

Branch\* rightBranch;

};

void AddElement(Branch\*&, double);

void TreeOutput(Branch\*, int);

void SwapMaxAndMin(Branch\*&);

double getMax(Branch\*);

double getMin(Branch\*);

void setMinOnMaxPosition(Branch\*&, double);

void setMaxOnMinPosition(Branch\*&, double);

**Source.cpp**

#include "Header.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 TreeOutput(Branch\* branch, int tabs) {

if (!branch) {

return;

}

TreeOutput(branch->rightBranch, ++tabs);

for (int i = 1; i < tabs; i++) {

cout << "\t";

}

cout << branch->data << endl;

TreeOutput(branch->leftBranch, tabs);

}

void SwapMaxAndMin(Branch\*& branch) {

if (!branch) {

return;

}

double max = getMax(branch);

double min = getMin(branch);

setMinOnMaxPosition(branch, min);

setMaxOnMinPosition(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 setMinOnMaxPosition(Branch\*& branch, double value) {

if (branch->rightBranch != NULL) {

setMinOnMaxPosition(branch->rightBranch, value);

}

else {

branch->data = value;

return;

}

}

void setMaxOnMinPosition(Branch\*& branch, double value) {

if (branch->leftBranch != NULL) {

setMaxOnMinPosition(branch->leftBranch, value);

}

else {

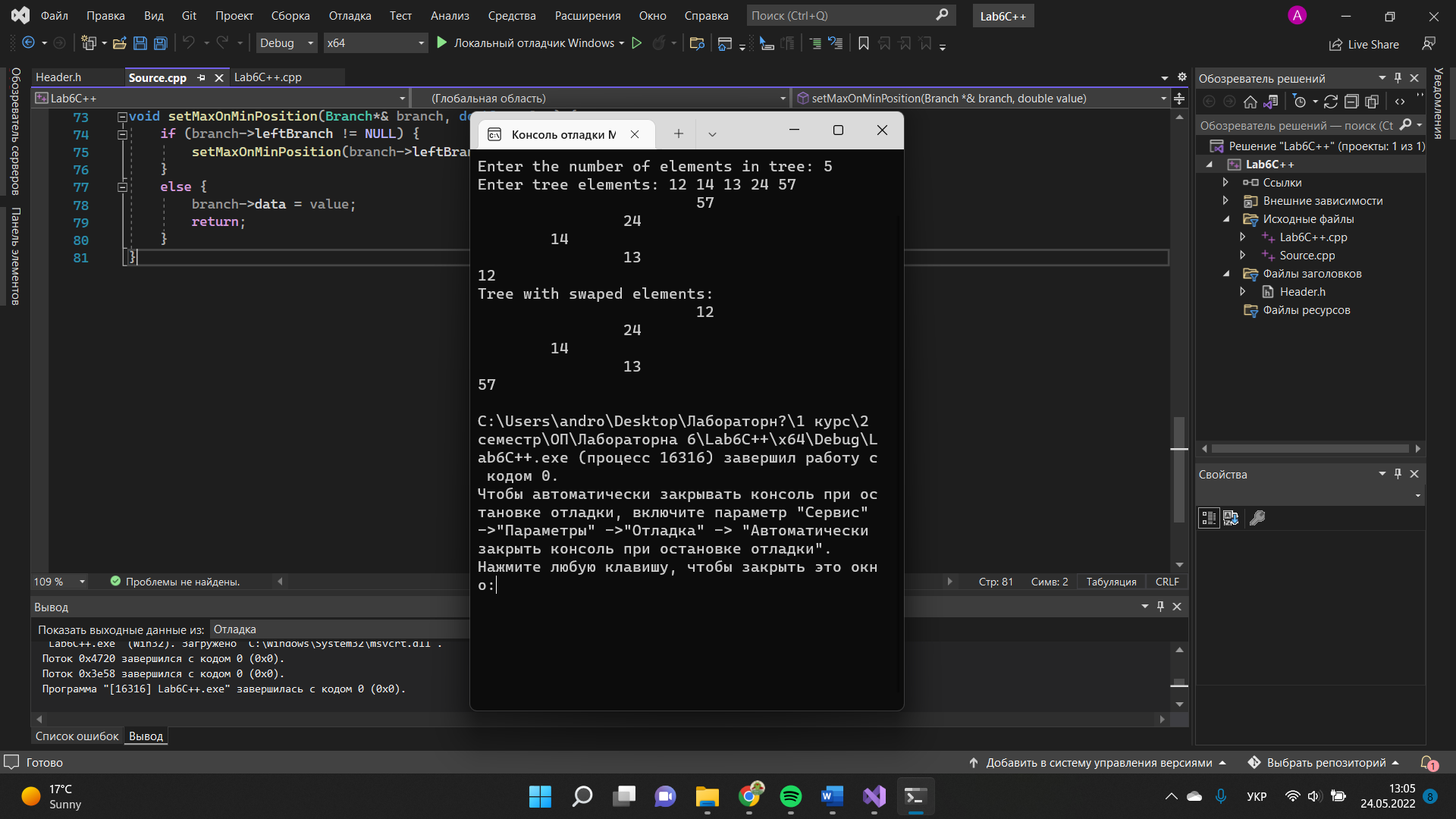
branch->data = value;

return;

}

}

**Скріншот роботи програми на C++:**

****

**Висновок:** під час виконання лабораторної роботи я вивчив особливості організації і обробки дерев.