
Основи програмування – 2

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.
Методології програмування»

«Дерева»

Варіант 29

Виконав студент ПІ-13 Романюк Діана Олексіївна
(шифр, прізвище, ім'я, по батькові)

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

Лабораторна робота 6

Мета – вивчити особливості організації і обробки дерева.

Варіант 29

Постановка задачі:

29. Побудувати дерево, елементами якого є символи. Визначити і вивести на друк усі термінальні вершини (листя) цього дерева.

Код програми на C++:

HeaderLab6.h

```
#pragma once
#include <iostream>

using namespace std;

class Node
{
public:
    char symbol;
    Node* LEFT;
    Node* RIGHT;
};

class Tree
{
public:
    Node* root;
    Tree() : root(0) {};
    void AddElement(char s);
    void printTree();
    void printLeafs();
private:
    int getLevels(Node*);
    void printTree(Node*, int);
    void AddElement(Node*&, char);
    void printLeafs(Node*);
};
```

Lab6.cpp

```
#include "HeaderLab6.h"

int main() {
    Tree tree;

    int n;
    cout << "Enter amount of tree elements: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        char s;
        cout << "Enter symbol: ";
        cin >> s;
        tree.AddElement(s);
    }
```

```
}

cout << "Entered tree: " << endl;
tree.printTree();

cout << "Its terminal nodes are: " << endl;
tree.printLeafs();

}
```

HeaderLab6.cpp

```
#include "HeaderLab6.h"

void Tree::AddElement(Node*& node, char symbol) {
    if (!node) {
        node = new Node;
        node->symbol = symbol;
        node->LEFT = 0;
        node->RIGHT = 0;
    }
    else {
        if (symbol < node->symbol) {
            AddElement(node->LEFT, symbol);
        }
        else if (symbol > node->symbol) {
            AddElement(node->RIGHT, symbol);
        }
    }
}

void Tree::AddElement(char s) {
    this->AddElement(this->root, s);
}

void Tree::printLeafs() {
    this->printLeafs(this->root);
}

void Tree::printLeafs(Node* root)
{
    if (root)
    {
        printLeafs(root->LEFT);
        if (!root->LEFT && !root->RIGHT)
            cout << root->symbol << endl;
        printLeafs(root->RIGHT);
    }
}

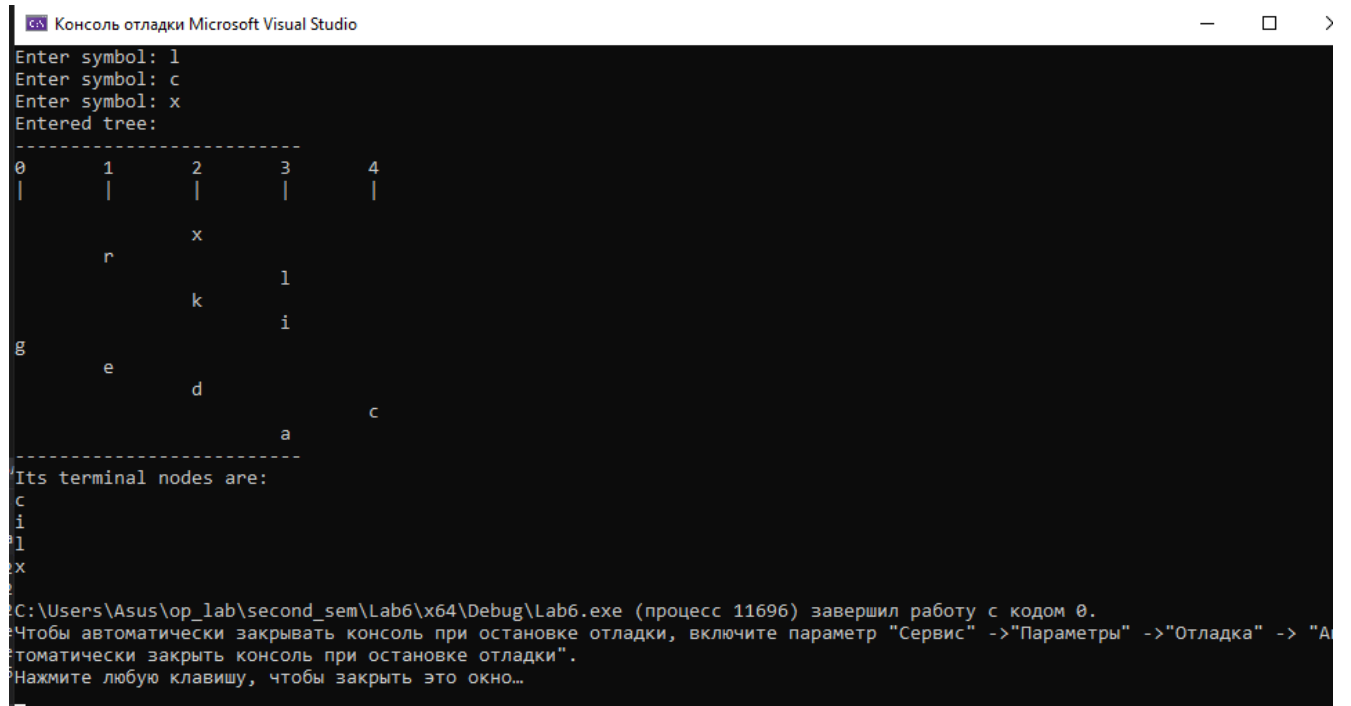
void Tree::printTree(Node* root, int space) {
    if (root == NULL) return;
    printTree(root->RIGHT, ++space);
    for (int i = 1; i < space; i++) {
        cout << '\t';
    }
    cout << root->symbol << endl;
    printTree(root->LEFT, space);
}

void Tree::printTree() {
    cout << "-----\n";
    int l = this->getLevels(root);
    for (int i = 0; i < l; i++) cout << i << '\t';
    cout << endl;
    for (int i = 0; i < l; i++) cout << "| \t";
    cout << endl << endl;
    printTree(root, 0);
}
```

Основи програмування – 2

```
cout << "-----\n";
}

int Tree::getLevels(Node* root) {
    if (root == NULL) return 0;
    return max(getLevels(root->LEFT), getLevels(root->RIGHT)) + 1;
}
```



Консоль отладки Microsoft Visual Studio

```
Enter symbol: l
Enter symbol: c
Enter symbol: x
Entered tree:
-----
0      1      2      3      4
|      |      |      |      |
      r      x      l      c
      |      |      |
g      k      i
      |      |
      e      d
          a
-----
Its terminal nodes are:
c
i
l
x
2
C:\Users\Asus\op_lab\second_sem\Lab6\x64\Debug\Lab6.exe (процесс 11696) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "А
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

Висновки: під час лабораторної роботи ми вивчили особливості організації і обробки дерева.