

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №3**  
**по дисциплине «Алгоритмы и структуры данных»**  
**Тема: Бинарные деревья**

Студент гр. 9303

\_\_\_\_\_

Хафеева Н. Л.

Преподаватель

\_\_\_\_\_

Филатов А. Ю.

Санкт-Петербург

2020

### Цель работы.

Ознакомиться с понятием «бинарное дерево», изучить его особенности и реализовать программу, решающую поставленную задачу с помощью бинарного дерева.

### Задание.

Вариант 4д

Для заданного бинарного дерева  $b$  типа  $BT$  с произвольным типом элементов определить, есть ли в дереве  $b$  хотя бы два одинаковых элемента.

### Основные теоретические положения.

Бинарное дерево – конечное множество узлов, которое либо пусто, либо состоит из корня и двух непересекающихся бинарных деревьев, называемых правым поддеревом и левым поддеревом. Так определенное бинарное дерево *не* является частным случаем дерева. Например, бинарные деревья, изображенные на рис. 3.3, различны между собой, так как в одном случае корень имеет пустое правое поддерево, а в другом случае правое поддерево не пусто. Если же их рассматривать как деревья, то они идентичны.



Рис. 3.3. Бинарные деревья из двух узлов

Определим скобочное представление бинарного дерева (БД):

$\langle \text{БД} \rangle ::= \langle \text{пусто} \rangle \mid \langle \text{непустое БД} \rangle,$

$\langle \text{пусто} \rangle ::= \Lambda,$

$\langle \text{непустое БД} \rangle ::= ( \langle \text{корень} \rangle \langle \text{БД} \rangle \langle \text{БД} \rangle ).$

### Выполнение работы.

Для выполнения программы были реализована функция `CreatBinTree`.

В функции `void CreatBinTree(BinTree* b, string& str, unsigned int& n, bool& cond)` считывается заданное бинарное дерево, каждый последующий символ сравнивается с предыдущими и, если символ совпадает с каким-либо предыдущим символом, то переменная `cond` изменяется на `true`. Исходный код программы представлен в приложении А. Результаты тестирования включены в приложение Б.

### **Выводы.**

Ознакомилась с понятием «бинарное дерево», изучила его особенности и реализовала программу, решающую поставленную задачу с помощью бинарного дерева.

Была реализована программа, которая определяет, есть ли в бинарном дереве два одинаковых узла. Была создана структура `BinTree`, которая описывает бинарное дерево. Так же была реализована функция `CreatBinTree`, которая считывает дерево и осуществляет проверку.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <vector>
#include <fstream>
#include <string>

using namespace std;

struct BinTree {
    BinTree* left = nullptr;
    BinTree* right = nullptr;
    char data = '0';
};

void CreatBinTree(BinTree* b, string& str, unsigned int& n, bool& cond) {
    if (n >= str.length()) return;

    if (str[n] == '#') {
        n++;
        return;
    } else {
        b->data = str[n];
        n++;
        for (int i = 0; i < n - 1; i++) {
            if (str[i] == b->data && cond == false) {
                cond = true;
            }
        }

        b->left = new BinTree;
        CreatBinTree(b->left, str, n, cond);
        b->right = new BinTree;
        CreatBinTree(b->right, str, n, cond);
    }
}

void DeleteBinTree(BinTree* b) {
    if (b == nullptr) {
        return;
    }

    DeleteBinTree(b->left);
    DeleteBinTree(b->right);
    delete b;
}
```

```

        return;
    }

int main() {
    ifstream fin;
    unsigned int n = 0;
    bool cond = false;
    fin.open("inputfile.txt");
    string str;

    if (!fin.is_open()) {
        cout << "Permission denied or wrong path.";
        return 0;
    }

    while (getline(fin, str)) {
        BinTree* b = new BinTree;
        cout << "Input data: " << str << endl;
        CreatBinTree(b, str, n, cond);

        if (cond) {
            cout << "TRUE" << endl;
        } else {
            cout << "FALSE" << endl;
        }

        cond = false;
        n = 0;
        DeleteBinTree(b);
    }
    fin.close();
    return 0;
}

```

## ПРИЛОЖЕНИЕ Б

### ТЕСТИРОВАНИЕ

Таблица Б.1 — Примеры тестовых случаев

№	Входные данные	Выходные данные	Комментарий
1.	a abm##nefpl##k## anmuachassakda# asbndb sdfghjk asdhdgas	Input data: a FALSE Input data: abm##nefpl##k## FALSE Input data: anmuachassakda# TRUE Input data: asbndb TRUE Input data: sdfghjk FALSE Input data: asdhdgas TRUE	Программа работает корректно
2.	afgh##usi bnanak	Input data: afgh##usi FALSE Input data: bnanak TRUE	Программа работает корректно
3.	try##usdka t uytjgd ff	Input data: try##usdka FALSE Input data: t FALSE Input data: uytjgd FALSE Input data: ff TRUE	Программа работает корректно