



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
**"МИРЭА - Российский технологический университет"**

## **РТУ МИРЭА**

---

Институт информационных технологий (ИТ)  
Кафедра математического обеспечения и стандартизации  
информационных технологий (МОСИТ)

### **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4** **по дисциплине** **«Структуры и алгоритмы обработки данных»**

Тема. Бинарные деревья

Выполнил студент группы ИКБО-25-22

Ракитин В.А.

Принял доцент

Бузыкова Ю.С.

Москва 2023

**Цель работы:** изучить теорию и научиться программировать бинарные деревья.

**Задание на лабораторную работу:**

№ варианта	Задание на программирование бинарного дерева
1	Написать программу копирования одного бинарного дерева в другое

**Ход работы:**

```
#include <iostream>
#include <sstream>

// Структура для представления узла бинарного дерева
struct Node {
    int data;
    Node* left;
    Node* right;

    Node(int val) : data(val), left(nullptr), right(nullptr) {}
};

// Функция для копирования бинарного дерева
Node* copyTree(Node* root) {
    if (root == nullptr) {
        return nullptr;
    }

    // Создаем новый узел с тем же значением, рекурсивно копируем
    // левое и правое поддерево
    Node* newNode = new Node(root->data);
    newNode->left = copyTree(root->left);
    newNode->right = copyTree(root->right);

    return newNode;
}

// Функция для вставки значения в бинарное дерево
Node* insert(Node* root, int value) {
    if (root == nullptr) {
        return new Node(value);
    }

    if (value < root->data) {
        root->left = insert(root->left, value);
    }
    else if (value > root->data) {
        root->right = insert(root->right, value);
    }

    return root;
}
```

```

}

// Функция для вывода бинарного дерева в виде дерева
void printTree(Node* root, std::string prefix = "", bool isLeft =
false) {
    if (root != nullptr) {
        std::stringstream ss; // Используем std::stringstream для
конвертации int в строку
        ss << root->data;
        std::cout << (prefix + (isLeft ? "|-- " : "'-- ") + ss.str())
<< std::endl;

        printTree(root->left, prefix + (isLeft ? "|  " : "  "),
true);
        printTree(root->right, prefix + (isLeft ? "|  " : "  "),
false);
    }
}

int main() {

    setlocale(LC_ALL, "ru");

    Node* root = nullptr;

    // Пользовательский ввод значений узлов дерева
    int value;
    while (true) {
        std::cout << "Введите значение (или -1 для завершения ввода):
";
        std::cin >> value;

        if (value == -1) {
            break;
        }

        root = insert(root, value);
    }

    // Копируем исходное дерево
    Node* copiedTree = copyTree(root);

    // Выводим исходное дерево в виде дерева
    std::cout << "Исходное дерево:" << std::endl;
    printTree(root);

    // Выводим скопированное дерево в виде дерева
    std::cout << "Скопированное дерево:" << std::endl;
    printTree(copiedTree);

    return 0;
}

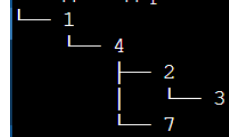
```

## Тестирование:

Проверим правильность выполнения программы:

```
Введите значение (или -1 для завершения ввода): 1
Введите значение (или -1 для завершения ввода): 4
Введите значение (или -1 для завершения ввода): 2
Введите значение (или -1 для завершения ввода): 7
Введите значение (или -1 для завершения ввода): 3
Введите значение (или -1 для завершения ввода): -1
```

Исходное дерево:



Скопированное дерево:

