

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский политехнический
университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные
системы» направление подготовки: 09.03.01– «Информатика и
вычислительная техника»

Лабораторная работа

**по дисциплине «Основы алгоритмизации и программирования» на
тему «Бинарные деревья»**

Выполнила студентка группы

ИВТ-23-16 Долганова Диана Евгеньевна

Проверил:

доцент кафедры ИТАС

Яруллин Денис Владимирович

(оценка)

(дата)

(подпись)

г. Пермь, 202

Постановка задачи

1. Самостоятельно придумать вид Дерева и реализовать алгоритмы для этого собственного варианта бинарного дерева поиска, имеющего не менее трёх уровней .
2. Алгоритмы:
 - 1) Необходимо реализовать функции для редактирования дерева: - Вставка узла, - Удаление узла, - Поиск элемента по ключу.
 - 2) Реализовать алгоритмы обхода дерева:
 - 3) Прямой.
 - 4) Симметричный.
 - 5) Обратный.
 - 6) Выполнить задание своего варианта из методички: Laby_Chast_3.docx
3. Реализовать алгоритм балансировки дерева.
4. Реализовать вертикальную и горизонтальную печать.
5. Визуализацию дерева выполнить с использованием любой доступной графической библиотеки – SFML, SDL, OpenGL...

Анализ задач

Структуры и методы класса Tree:

Структура Node: Описывает узел бинарного дерева с полями: data (значение узла), left и right (указатели на левого и правого потомков), num (номер узла), x и y (координаты узла на экране).

Поля класса Tree:

1. int arMean: переменная для хранения суммы значений узлов дерева (используется для расчета среднего арифметического).
2. int count: общее количество узлов в дереве.
3. Node *root: указатель на корневой узел дерева.

Метод init(int x): Инициализирует корневой узел дерева с заданным значением x и добавляет его данные к arMean.

Метод `addNode(int x, Node *tree, int z, int y, int h)`:

1. Рекурсивно добавляет узел со значением `x` в дерево, с учетом переданных координат `z`, `y` и шага `h`.
2. Увеличивает значение `arMean` на `x`.

Метод `deletNodes(Node *tree)`: Рекурсивно удаляет все узлы дерева, освобождая выделенную для них память.

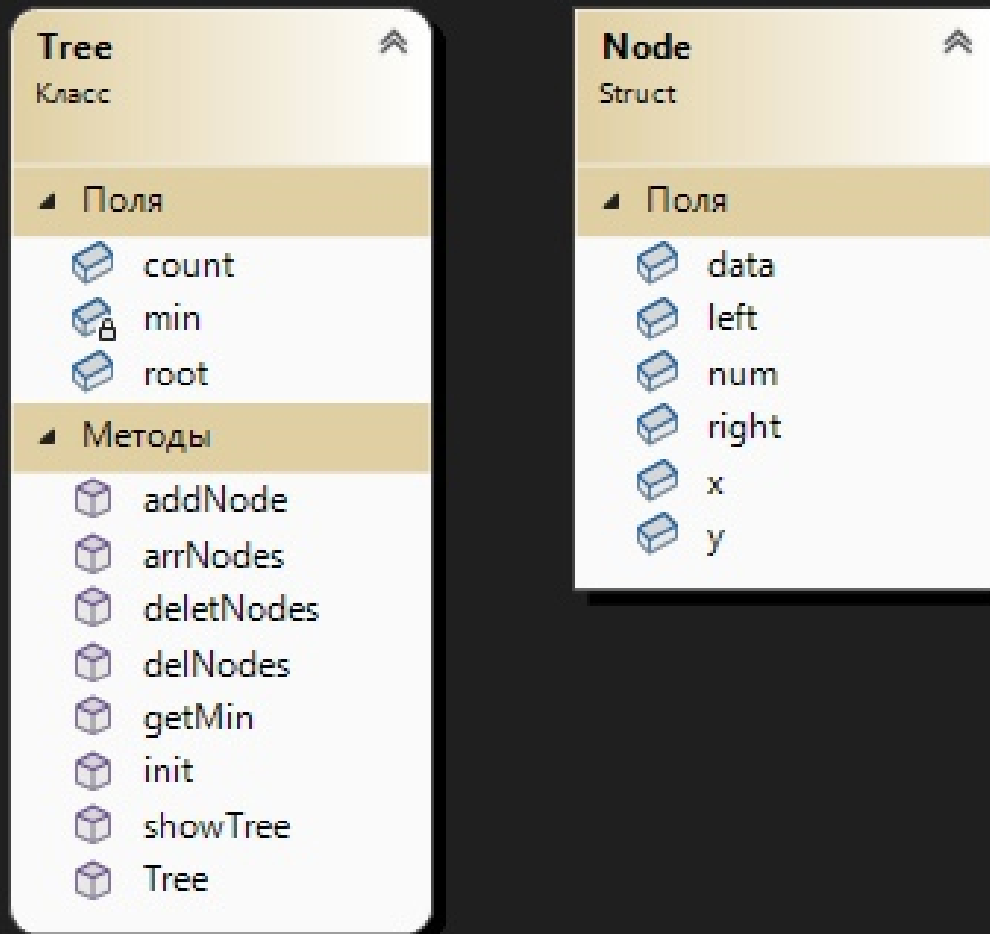
Метод `delNodes(int data, Node *tree)`: Находит и удаляет узел с данным значением `data` из дерева, переназначая родительские указатели.

Метод `showTree(Node *tree)`: Рекурсивно обходит дерево в порядке "левый-корень-правый" и выводит на экран номер узла и его данные.

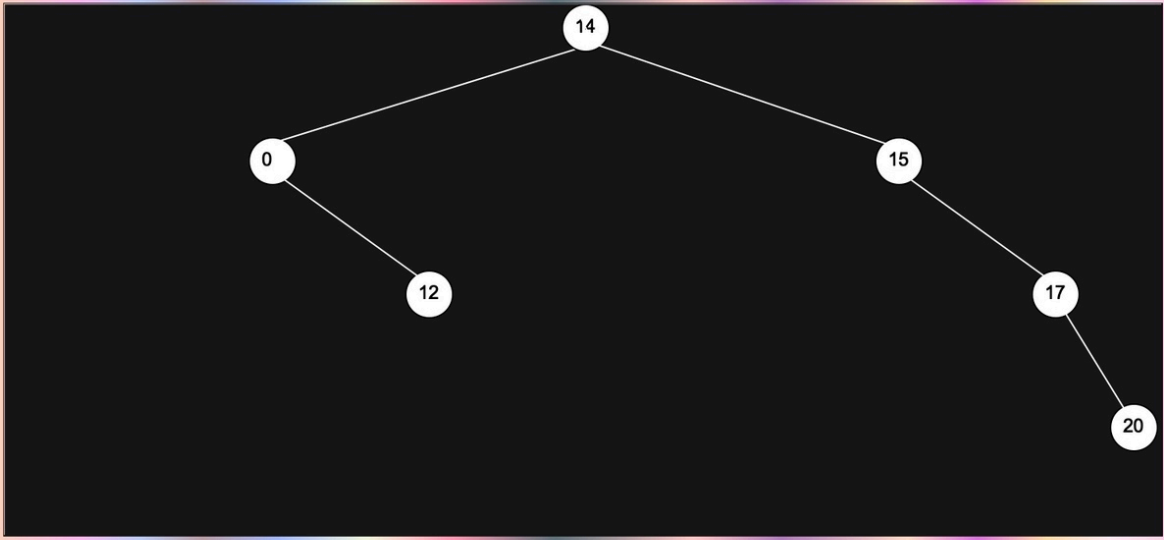
Метод `arithMean()`: Возвращает среднее арифметическое всех значений узлов дерева.

Метод `arrNodes(QString a, Node *tree)`: Добавляет узлы в дерево, значения которых перечислены в строке `a` через запятую.

UML - диаграмма



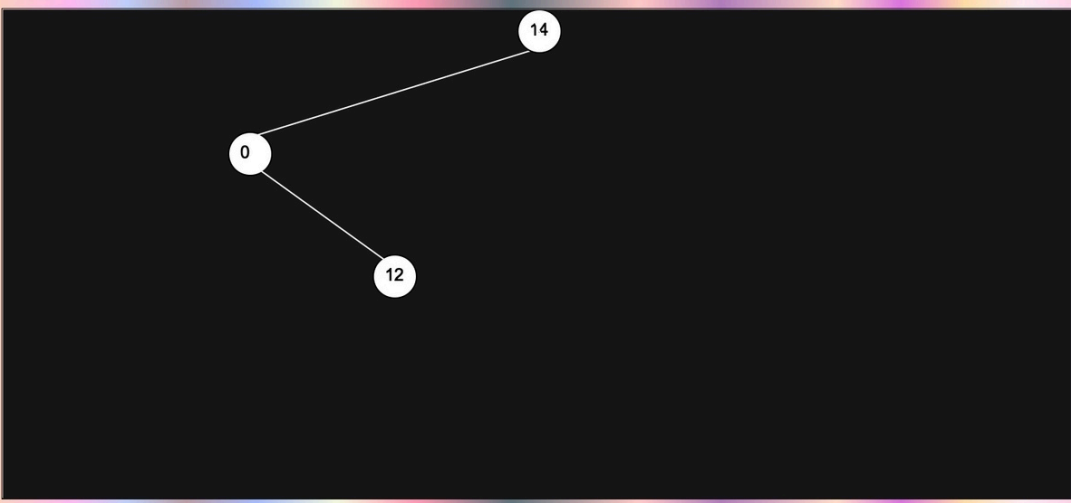
Демонстрация работы



● Удалить узел
● Инициализация
● Добавить узел
● Последовательность узлов

Применить Показать

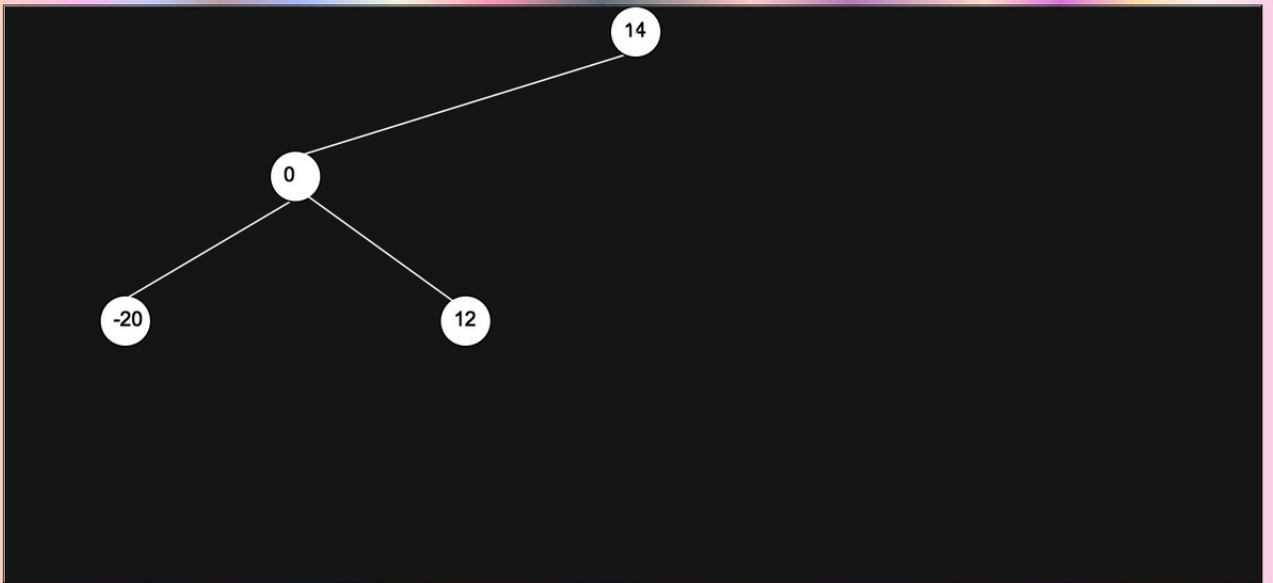
0



● Удалить узел
● Инициализация
● Добавить узел
● Последовательность узлов

15

Применить Показать



- Удалить узел
- Инициализация
- Добавить узел
- Последовательность узлов

-20

Применить

Показать

-20

Ссылка на GitHub

https://github.com/dolganovadd/LABS_PSTU/tree/main/sem_2.gitkeep/labs.gitkeep/binary%20trees.gitkeep