

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 42

ОТЧЁТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

старший преподаватель
должность, уч. степень, звание

подпись, дата

С.Ю. Гуков
инициалы, фамилия

ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №4

Бинарные деревья поиска

по курсу:

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. № 4329

подпись, дата

Д.С. Шаповалова
инициалы, фамилия

Санкт-Петербург 2025

ОГЛАВЛЕНИЕ

| | |
|--|---|
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №4..... | 1 |
| Цель работы..... | 3 |
| Постановка задачи..... | 3 |
| Краткое описание хода разработки и назначение используемых технологий..... | 4 |
| Назначение используемых технологий..... | 4 |
| Исходный код программы (с комментариями в необходимых местах)..... | 5 |
| Результаты работы программы с примерами разных сценариев (скриншоты)..... | 5 |
| Выводы..... | 8 |

Цель работы

Изучить структуру данных бинарного дерева поиска, разработать расширенный класс для работы с ним и получить практический навыки использования и визуализации этой структуры данных.

Постановка задачи

Необходимо разработать программу, которая генерирует на вход случайную числовую последовательность случайного размера и строит из этой последовательности бинарное дерево.

Программа также должна иметь следующий функционал:

- удаление / добавление узла с выводом статуса операции (например, успешное добавление, ошибка валидации, узел не найден);
- поиск узла в дереве;
- в консольном варианте интерфейса хоть какое-то наглядное и понятное отображение родителя и левого и правого потомков (например, через ASCII графику в консоли);

Необязательно (отсутствует в разработанной программе):

- графическое представление дерева (опционально);
- поиск узла в заданном поддереве (опционально);
- балансировка дерева (опционально, божественный уровень).

Проект может быть выполнен либо в качестве консольного приложения (тогда обязателен командно-текстовый интерфейс), либо иметь графический пользовательский интерфейс (User Interface, UI), а также может быть написан на любом языке программирования.

Требования к структуре проекта

- ☐ Применение принципов ООП (наследования, инкапсуляции, полиморфизма, абстракции) и SOLID
- ☐ Дружелюбный командно-текстовый либо графический пользовательский интерфейс

КРАТКОЕ ОПИСАНИЕ ХОДА РАЗРАБОТКИ И НАЗНАЧЕНИЕ ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ

Ход разработки:

1. Генерируем случайную числовую последовательность случайного размера.
2. Функция построения бинарного дерева (тут же функция вставки узла).
3. Командно-текстовый дружелюбный интерфейс (user friendly).
4. Реализуем отрисовку дерева в консоль.
5. Реализуем поиск узла.
6. Реализуем удаление узла, трансплантацию дерева.

НАЗНАЧЕНИЕ ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ

- Модуль random - для генерации случайной последовательности, включая генерацию её размера.

Разработано 2 класса: для представления узла и для дерева целиком. Классы связаны отношением композиции.

Исходный код программы (с комментариями в необходимых местах)

Представлен на платформе гитхаб: [SUAI_09.03.02/4_semester/ТП/ТП-лр4-5/4_lr.py](https://github.com/MyataEtoki/SUAI_09.03.02/4_semester/ТП/ТП-лр4-5/4_lr.py) at main · MyataEtoki/SUAI_09.03.02

РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ С ПРИМЕРАМИ РАЗНЫХ СЦЕНАРИЕВ (СКРИНШОТЫ)

```
Сгенерированная последовательность: [38, 99, 42, 11, 6]

Бинарное дерево:
|
|  ┌ 99
|  └ 42
└ 38
   └ 11
      └ 6

Выберите действие:
1. Добавить узел
2. Удалить узел
3. Найти узел
4. Показать дерево
5. Выход
Введите номер действия:
```

Рисунок 1 - Старт программы

```
Введите номер действия: 1
Введите значение для добавления: 23
Находимся в узле 38
23 < 38 → идём влево
Находимся в узле 11
23 > 11 → идём вправо
Дошли до пустого узла – значение не найдено.
Узел со значением 23 успешно добавлен.

Выберите действие:
1. Добавить узел
2. Удалить узел
3. Найти узел
4. Показать дерево
5. Выход
Введите номер действия: 4

Бинарное дерево:
|
|  ┌ 99
|  └ 42
└ 38
   └ ┌ 23
      └ 11
         └ 6
```

Рисунок 2 - Добавили узел “23”.

```
Бинарное дерево:
|
|  ┌ 99
|  └ 42
└ 38
   ┌ 23
   └ 11
      └ 6

Выберите действие:
1. Добавить узел
2. Удалить узел
3. Найти узел
4. Показать дерево
5. Выход
Введите номер действия: 3
Введите значение для поиска: 42
Находимся в узле 38
42 > 38 → идём вправо
Находимся в узле 99
42 < 99 → идём влево
Находимся в узле 42
Значение 42 найдено!
```

Рисунок 3 - Нашли узел “42”

```
Выберите действие:
1. Добавить узел
2. Удалить узел
3. Найти узел
4. Показать дерево
5. Выход
Введите номер действия: 1
Введите значение для добавления: 42
Находимся в узле 38
42 > 38 → идём вправо
Находимся в узле 99
42 < 99 → идём влево
Находимся в узле 42
Значение 42 найдено!
Узел со значением 42 уже существует.
```

Рисунок 4 - Не можем добавить существующий узел

```
Бинарное дерево:
|
|  ┌ 99
|  └ 42
└ 38
   |
   └ 23
      └ 11
         └ 6

Выберите действие:
1. Добавить узел
2. Удалить узел
3. Найти узел
4. Показать дерево
5. Выход
Введите номер действия: 2
Введите значение для удаления: 11
Находимся в узле 38
11 < 38 → идём влево
Находимся в узле 11
Значение 11 найдено!
Узел со значением 11 успешно удалён.
```

Рисунок 5 - Удаляем узел “11”

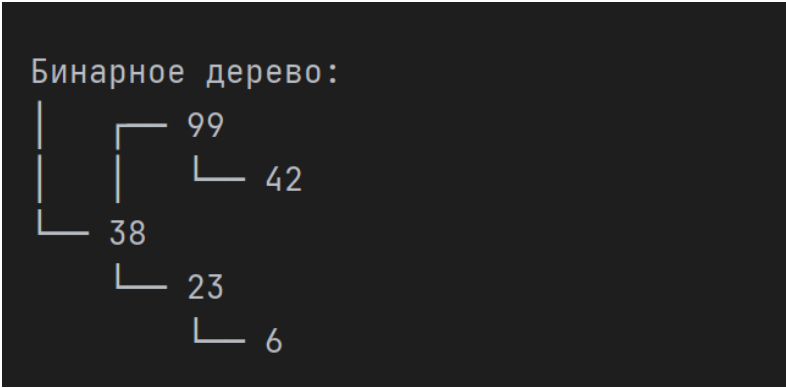


Рисунок 6 - Последствия удаления узла “11”

Выводы

В ходе выполнения лабораторной работы мной были освоены и изучены: алгоритм построения бинарного дерева, поиска, добавление и удаления(самое интересное) значения. Написанная программа была протестирована, полученный результат соответствует ожиданиям.