

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

Лабораторна робота №5

з дисципліни «Алгоритми та структури даних»

на тему: «Дерева»

Виконав:
студент гр.ПЗ1911
Сафонов Д. Є.
Прийняла:
Куруп'ятник О. С.

Дніпро, 2021

Тема. Дерева.

Мета. Ознайомитися з поняттям дерева. Отримати практичні навички реалізації бінарного дерева пошуку.

Завдання. Розробити програму мовою java для реалізації бінарного дерева та операцій додавання, видалення, пошуку та перегляду інформації про певний об'єкт(вивід інформаційної частини), симетричного обходу. Клас дерева та вузла представити двома різними класами. Атрибути класу вузла повинні мати модифікатори доступу private або вузол має бути представлений як внутрішній клас класу дерева. Вузол повинен містити посилання на лівий та правий нащадки, поле ключа, поле інформаційної частини, не містити посилання на батьківський вузол. Значення ключа повинно обраховуватися на основі значення інформаційної частини. При додаванні нового вузла зі значенням ключа, яке вже є в дереві, виконати заміну інформаційної частини. Тип даних інформаційної частини визначити за параметризованим типом з індивідуального завдання до лабораторної роботи 4.

В програмі реалізувати текстове меню для виконання усіх операцій над деревом в довільному порядку. Для кожного пункту меню передбачити зворотній зв'язок у вигляді результату виконання операції та/або повідомлення про успішне виконання операції.

Індивідуальне завдання.

Варіант	Ситуація	Значення параметризованого типу
6	Снігуронька записує бажання дітей, які шикуються у чергу. Бажання можуть бути таких видів: іграшки, гаджети, одяг, смаколики. Дід Мороз виконує бажання по одному у порядку надходження. Порахувати скільки бажань кожного виду вже виконав Дід.	Бажання

Текст програми. [github](https://github.com)

Висновки.

Розроблений клас повністю ітеративний(використовується програмний стек замість фізичного). Хоч ітеративні алгоритми дещо складніше вони кращі за свої рекурсивні аналоги тому, що більшість мов програмування використовує фізичний стек, який досить швидко переповнюється. Але рекурсивні алгоритми набагато простіші для написання та розуміння(вони набагато коротші та не потребують маніпуляцій зі стеком).