МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ І СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

КАФЕДРА КОНСТРУЮВАННЯ ЕОА

**ЗВІТ**

з лабораторної роботи №5

по курсу «Алгоритмічні мови та програмування – 2»

на тему «Дерева.»

Виконав:

студент гр. ДК-92

Баскакова Олександра

Перевірив:

ст. викладач

Губар В.Г.

Київ – 2020

**Лабораторна робота №5**

*Дерева*

*Тема роботи*: Бінарні дерева.

*Мета роботи*: Складання алгоритмів з використанням бінарних дерев.

*Завдання:*

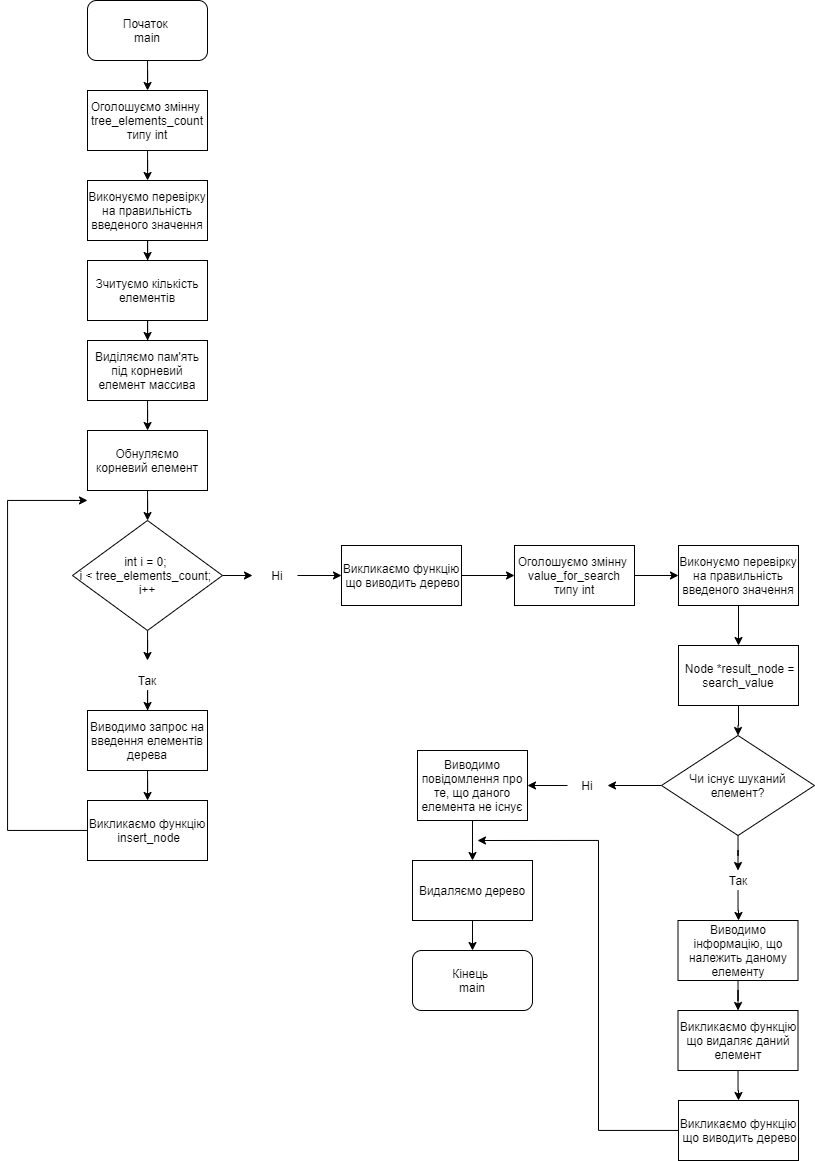
Створити бінарне дерево (виберіть згідно Вашого завдання), вивести його. Реалізувати функції: друку дерева (тип обходу згідно варіанту), створення/видалення дерева, вставки елементу, видалення елементу, пошуку елементу. Дерево містить елементи типу int, тип обходу – симетричний.

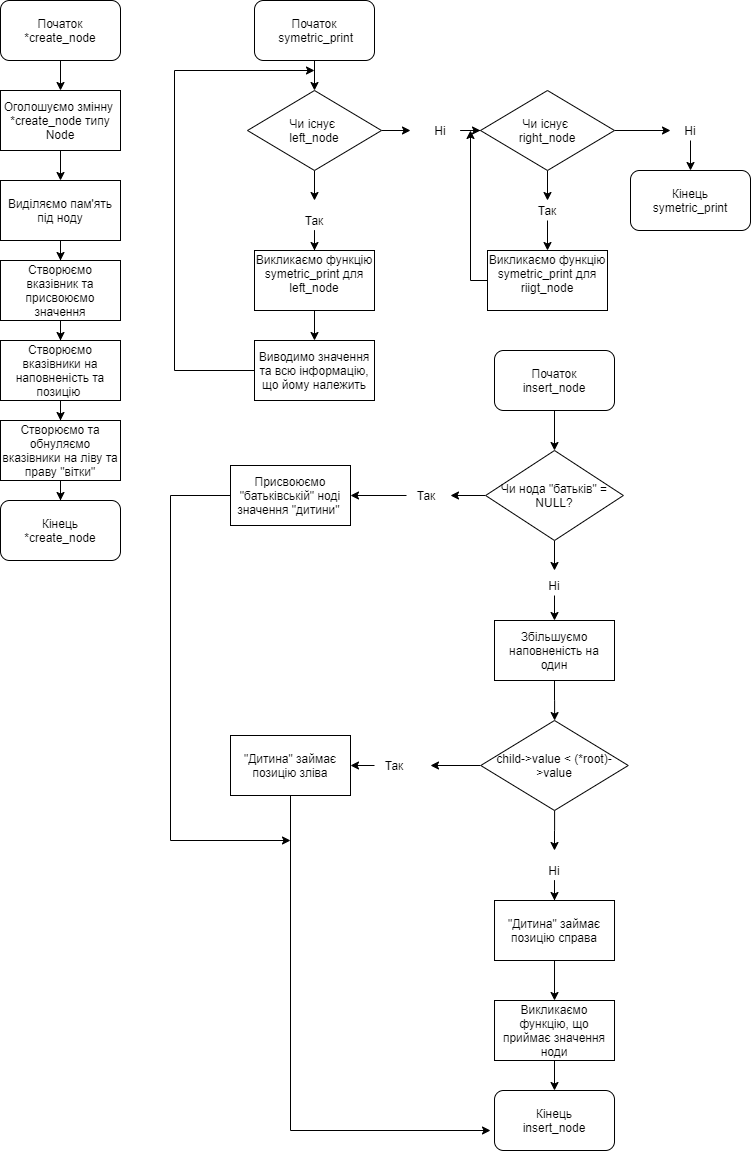
*Посилання на репозиторій:*

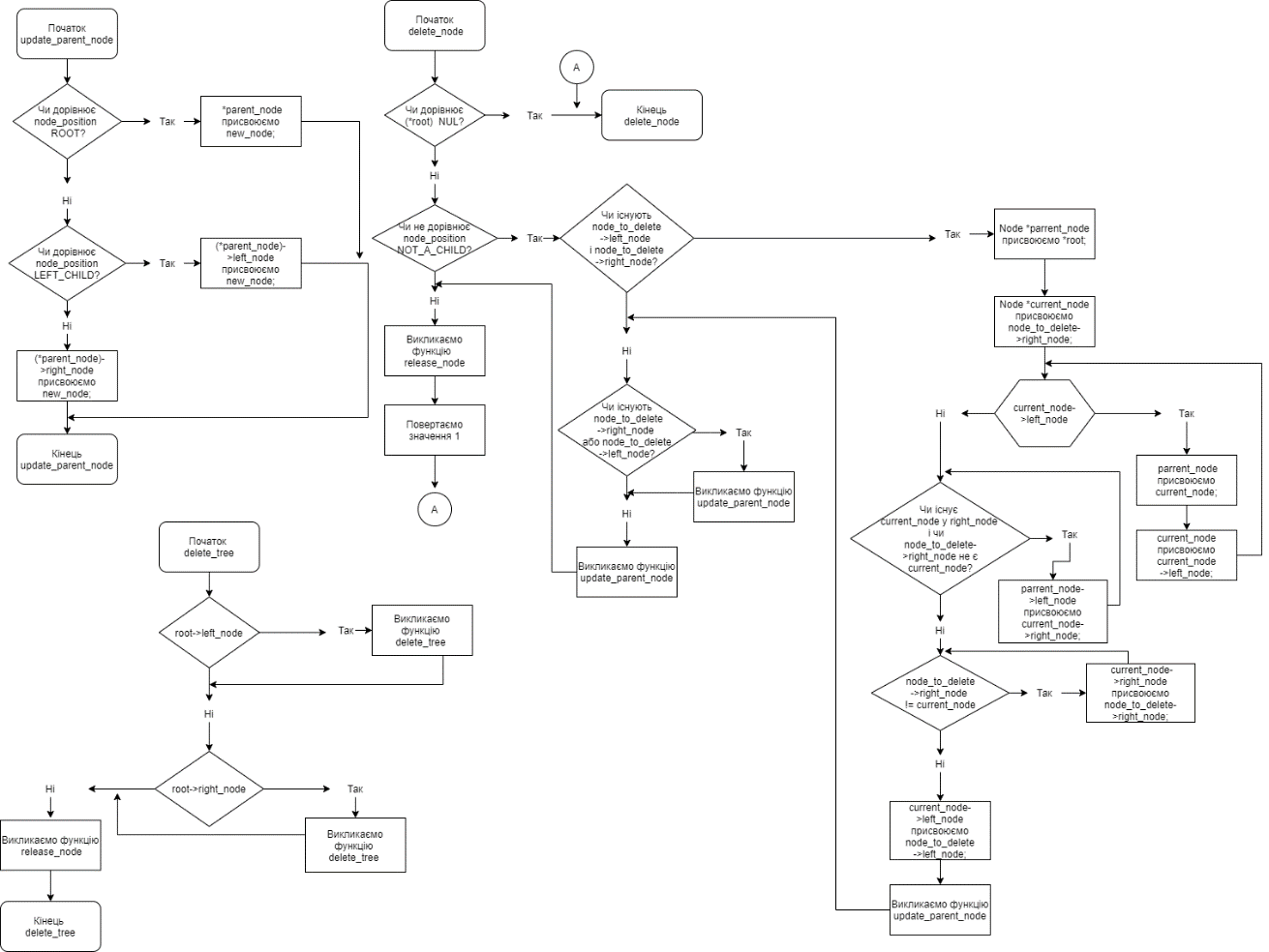
<https://github.com/sgubar/2020/tree/master/dk92/Baskakova%20Oleksandra/LAB4>

*Блок-схема:*

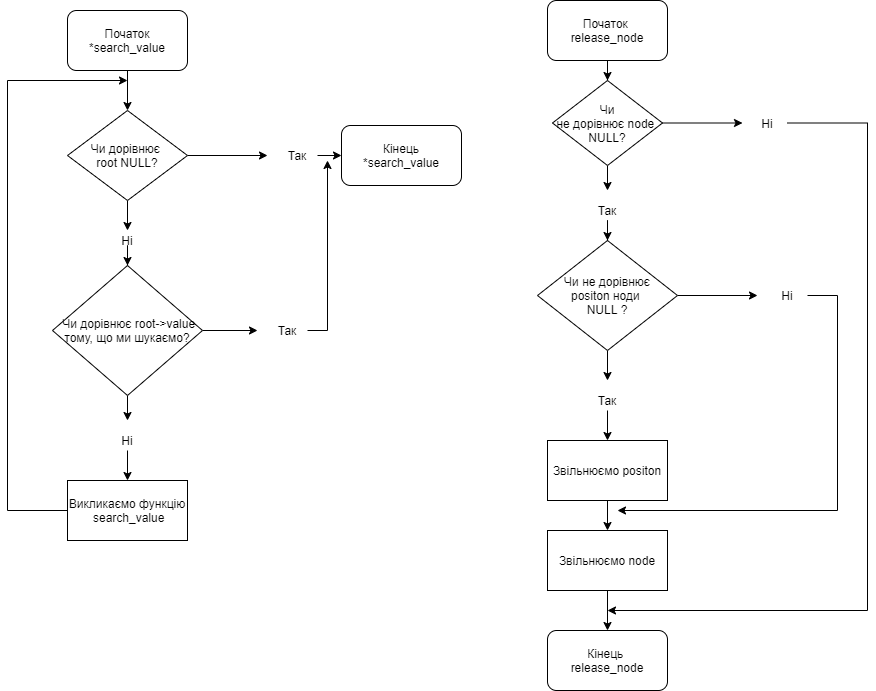
<https://github.com/sgubar/2020/blob/master/dk92/Baskakova%20Oleksandra/LAB5/main.png>

**

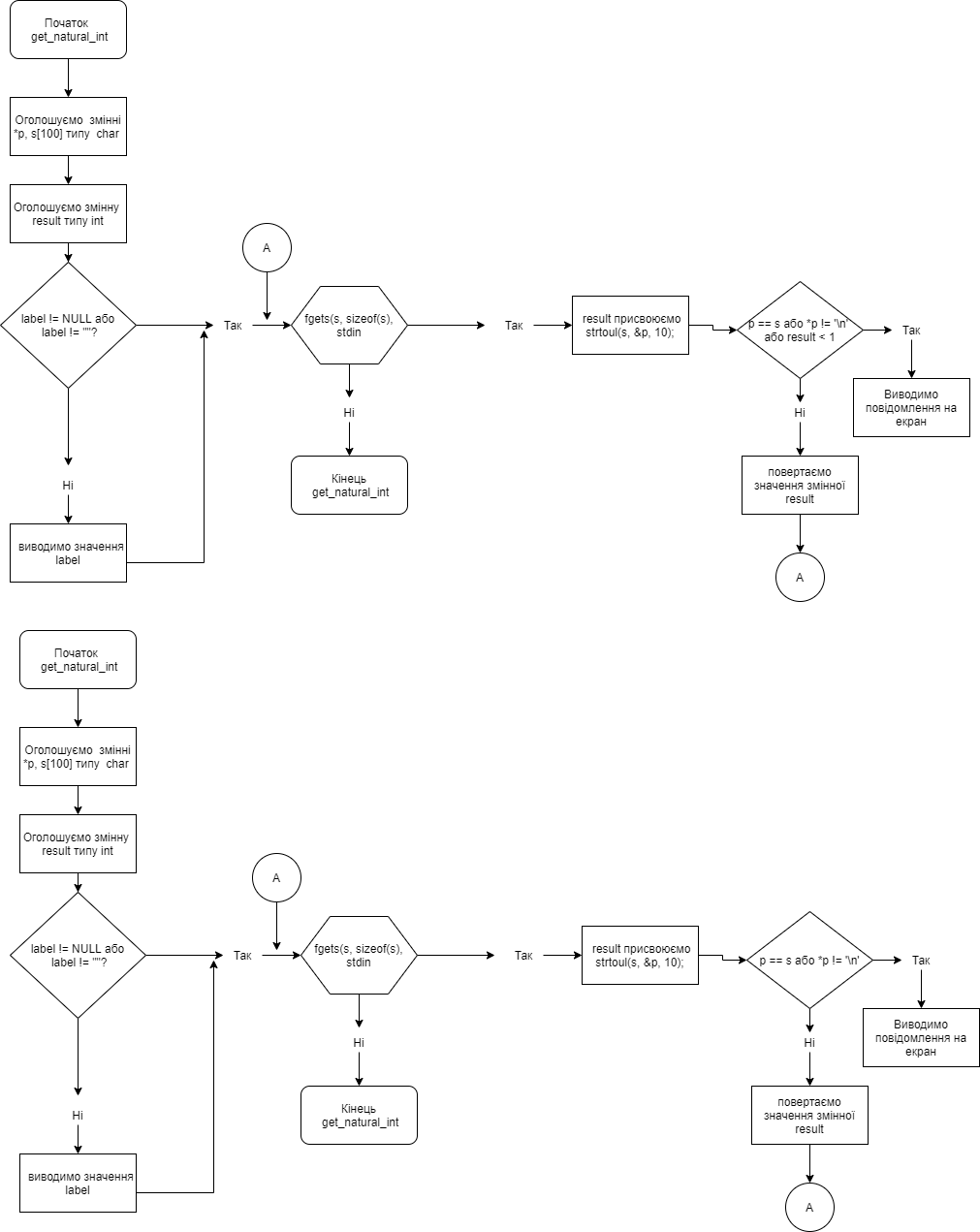
**<https://github.com/sgubar/2020/blob/master/dk92/Baskakova%20Oleksandra/LAB5/tree1.png>

**<https://github.com/sgubar/2020/blob/master/dk92/Baskakova%20Oleksandra/LAB5/tree2.png>

<https://github.com/sgubar/2020/blob/master/dk92/Baskakova%20Oleksandra/LAB5/tree3.png>

**

<https://github.com/sgubar/2020/blob/master/dk92/Baskakova%20Oleksandra/LAB5/tools.png>



*Висновок:*

Бінарне дерево - це кінцева безліч елементів, які або порожні, або містять елемент (корінь), пов'язаний з двома різними бінарними деревами – лівим і правим піддеревами. Кожен елемент бінарного дерева називається вузлом. Зв'язки між вузлами дерева називаються його гілками. На відміну від списка дані завжди впорядковані, тож тут легше знайти елемент, «відкидаючи» непотрібні значення.

Додавання елемента може в кінець або на початок ж може бути повільнішим, ніж в списку, проте легшим, ніж в масиві. Це пов’язано з тим, що бінарне дерево завжди має бути відсортованим.

Різні типи проходу існують для різних задач:

1. Pre-order варто використовувати саме тоді, коли ви знаєте що вам потрібно перевірити рути перед тим як перевіряти їх листя.

2. In-order обхід використовується коли нам треба перевіряти на початку дітей і тільки потім підніматися до батьківських вузлів.

3. Post-order – коли нам потрібно почати з листів і завершити головним вузлом - тобто розкласти дерево на те, як воно будувалося.

Дерева часто використовують для створення баз даних.