

# Список 05 – Лабораторная практика

## ООП структуры данных:

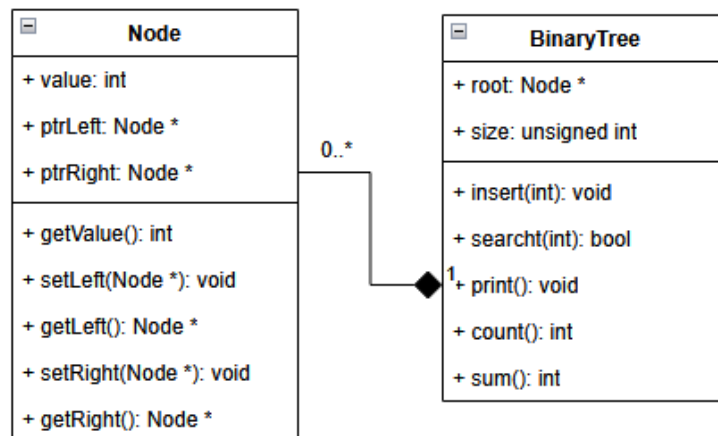
### Двоичные деревья, Связанные списки, Стеки

Предмет: Алгоритмизация и программирование

Преподаватель: Хольгер Эспинола Ривера

**1. Двоичные деревья.** Реализация двоичного дерева поиска с базовыми операциями в ООП. Связь, определяемая между двоичным поиском и его узлами, определяется следующим образом:

BinaryTree "1" \*-- "0..\*" Node: Отношения **Композиция**



[1]. Определите конструкторы по умолчанию, на основе параметров и конструкторы копирования объектов, setters и getters для класса **Node**. Атрибуты класса Node имеют привилегию частного доступа (**private**).

[2]. В случае класса **Binary Tree** (BST) он состоит из нуля или более узлов. Нет необходимости существовать в двух экземплярах. Существуют ли дружеские отношения между классами Node и Binary Tree. BST имеет следующий инвариант:  $left\ child \leq parent \leq right\ child$

[3]. В структуре данных двоичного дерева поиска (BST) реализуются следующие операции:

- **Вставка (insert):** добавление новых узлов с сохранением свойств BST
- **Поиск (search):** проверка наличия значения в дереве
- **Печать (print):** отображение элементов дерева по порядку
- **Подсчет (count):** возврат общего количества узлов
- **Суммирование (sum):** расчет суммы всех значений узлов