Отчет о проделанной работе

Лабораторная работа №2

Дисциплина: Алгоритмы и структуры данных

Студент: Хабутдинов Р.А.

Группа: ПРО-201в

# Задание на лабораторную работу

Сгенерировать 23, 2-х значных неповторяющихся числа.

Вывести их на экран.

Представить в виде поискового дерева.

Вывести элементы дерева в симметричном порядке.

# Входные и выходные данные

## Входные данные

**array** – сгенерированная последовательность.

## Выходные данные

**root** – узел корня дерева, от него можно добраться до любого элемента дерева.

**simmetrArray** – список состоящий из элементов дерева представленных в обратном порядке.

# Алгоритм построения дерева

//Создание поискового дерева

System**.**out**.**println**(**">> Build Search Tree"**);**

root **=** **new** Node**(**array**.**get**(**0**));** //Корнем дерева, назначается первый элемент из списка: array

array**.**remove**(**0**);**

**for** **(**Integer el **:** array**)** **{**

addR**(**el**,** root**);** //В цикле передаем все элементы списка: array, в ф-ю addR()

**}**

/\*\* Рекурсивное добавление элемента в Поисковое дерево \*/

private Node addR**(**int el**,** Node node**)** **{**

**if** **(**node **==** **null)** **{** //Если узел = null

node **=** **new** Node**(**el**);** //Создаем новый узел, знач-е узла берем из el

**}** **else** **if** **(**el **>** node**.**getData**())** **{** //Если el > текущего узла

node**.**setRight**(**addR**(**el**,** node**.**getRight**()));** //Передаем его в правое поддерево

**}** **else** **{** //Если el <= текущего узла

node**.**setLeft**(**addR**(**el**,** node**.**getLeft**()));** //Передаем его в левое поддерево

**}**

**return** node**;** //Ф-я возвращает текущий узел

**}**