Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

# по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Бинарное дерево поиска»

Выполнили:

студенты группы 24ВВВ3

Агапов И.А.

Любченко В.К.

Приняли:

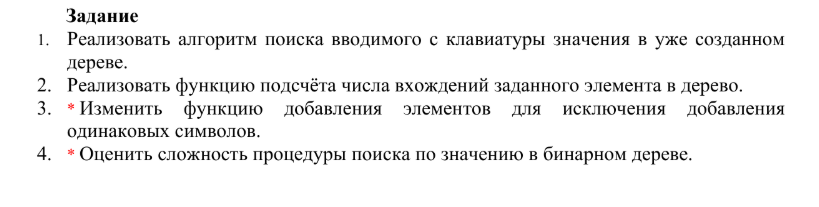
к.т.н., доцент Юрова О.В.

к.т.н., Деев М.В.

Пенза 2025

**Цель работы** – Изучить структуру данных «бинарное дерево поиска», реализовать базовые операции (добавление элементов, визуализация дерева, поиск количества вхождений заданного значения) и приобрести практические навыки работы с рекурсивными алгоритмами при обработке древовидных структур.

**Лабораторное задание:**

****

**4.\***

**Лучший случай: O(log n) - сбалансированное дерево**

**Средний случай: O(log n)**

**Худший случай: O(n) - вырожденное дерево**

**Память: O(h) - из-за рекурсии (h - высота дерева)**

**Код программы**

**C#**

using System.Numerics;

using System.Security.Cryptography.X509Certificates;

using System.Xml.Linq;

class TreeNode

{

public int Data { get; set; }

public TreeNode Left { get; set; }

public TreeNode Right { get; set; }

public TreeNode(int data)

{

Data = data;

Left = null;

Right = null;

}

}

class Tree

{

public TreeNode Root;

public int count = 0;

public Tree()

{

Root = null;

}

public void Add(int data)

{

Root = AddRecursive(Root, data);

}

private TreeNode AddRecursive(TreeNode node, int data)

{

if (node == null)

{

return new TreeNode(data);

}

if (data < node.Data)

{

node.Left = AddRecursive(node.Left, data);

}

else if (data >= node.Data)

{

node.Right = AddRecursive(node.Right, data);

}

//else

//{

// Console.WriteLine("Такое значение уже есть");

//}

return node;

}

public void PrintTree()

{

PrintTreeRecursive(Root, 0);

Console.WriteLine();

}

private void PrintTreeRecursive(TreeNode node, int level)

{

if (node == null) return;

// Сначала выводим правое поддерево

PrintTreeRecursive(node.Right, level + 1);

// Затем текущий узел

Console.WriteLine($"{new string(' ', level \* 3)}{node.Data}");

// Затем левое поддерево

PrintTreeRecursive(node.Left, level + 1);

}

public int Find(TreeNode current, int data)

{

if (current == null) return count;

if (current.Data == data) count++;

if (data >= current.Data && current.Right != null)

{

Find(current.Right, data);

}

if (data <= current.Data && current.Right != null)

{

Find(current.Left, data);

}

return count;

}

}

class Program

{

public static void Main()

{

Tree tree = new Tree();

int[] nums = { 8, 3, 10, 1, 6, 14, 8, 7, 13 };

for (int i = 0; i < nums.Length; i++)

{

Console.WriteLine("Добовляем число: " + nums[i]);

tree.Add(nums[i]);

}

Console.WriteLine("Дерерво:");

tree.PrintTree();

Console.WriteLine("\n ВВедите значение для поиска: ");

int data = Convert.ToInt32(Console.ReadLine());

tree.count = 0;

int result = tree.Find(tree.Root, data);

Console.WriteLine("Число вхождений: " + result);

}

}

**Результаты работы программы**

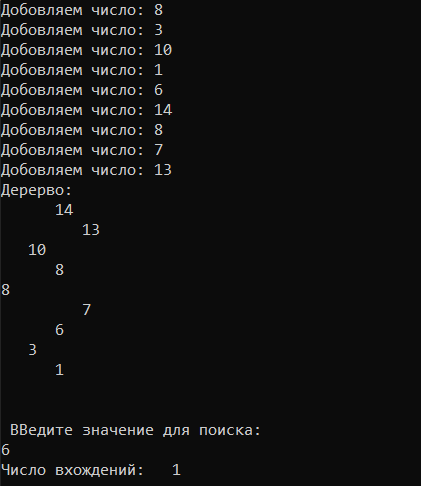


Рисунок 1 - Результат работы программы

**Вывод:** В ходе работы была успешно реализована структура бинарного дерева поиска. Освоены основные операции: добавление элементов с сохранением свойств дерева, визуализация структуры и рекурсивный поиск количества вхождений значения.