МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

профессионального образования

**«Дальневосточный федеральный университет»**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информационных систем управления**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3**

По дисциплине «Методы, средства и технологии информационных систем управления»

На тему «Иерархические структуры данных»

|  |
| --- |
| Выполнил студент гр. Б8219 |
| О.В. Константинов |
|  |
| Проверил ст. преп. |
| Е.А. Елсукова |
|  |
| (зачтено/не зачтено) |

г. Владивосток

2017

# Аннотация

Данный отчет подготовлен в рамках задания по предмету «Методы, средства и технологии информационных систем управления». Отчет призван систематизировать знания о методах работы с иерархическими структурами данных. Структура документа разделена, в соответствии с требованием преподавателя, на несколько разделов: «Постановка задачи», «Функциональное описание», «Примеры выполнения программы», «Вывод по работе».

# Постановка задачи

Требуется разработать демонстрационную программу, реализующую следующие методы работы с иерархическими структурами данных (AVL-деревьями).

* Добавление элемента.
* Удаление элемента.

В качестве входных данных желательно использовать:

* Случайные числа.
* Ввод с клавиатуры.

Результат работы программы должен представлять собой визуализацию полученной иерархической структуры. Необходимо вывести количество элементов в дереве.

# Функциональное описание

Программа представляет собой оконное приложение для ОС Windows NT (не ниже 5.1). Приложение было реализовано в IDE Visual Studio версии 2015, на языке C#.

Структурно, проект состоит из одной формы: MainForm.cs. И четырех модулей: AVLTree.cs, CircleNode.cs, IDrawable.cs, TreeNode.cs.

В программе, данные представляют собой последовательность целых чисел в диапазоне от 10 до 99. При удалении элемента, программа предлагает самое минимальное значение из дерева. При добавлении – случайное число от 10 до 99.

Общую структуру узлов дерева можно представить как:

public class AVLTreeNode<TNode> : IComparable<TNode> where TNode : IDrawable, IComparable

{

AVLTree<TNode> \_tree;

AVLTreeNode<TNode> \_left; // левый потомок

AVLTreeNode<TNode> \_right; // правый потомок

PointF Center; // позиция узла

#region Отрисовка дерева

#region Принадлежность точки узлу

#region Конструктор

#region Свойств

#region CompareTo

#region Балансоровка

#region Левый поворот

#region Правый поворот

#region ЛевыйПравый поворот

#region ПравыйЛевый поворот

#region Перемещение корня

}

Само же дерево представляет собой класс:

public class AVLTree<T> : IEnumerable<T> where T : IComparable, IDrawable

{

#region Вершина дерева

#region Количество узлов дерева

#region Добавление узла

#region Поиск узла

#region Удаление узла

#region Очистить дерево

#region Итераторы

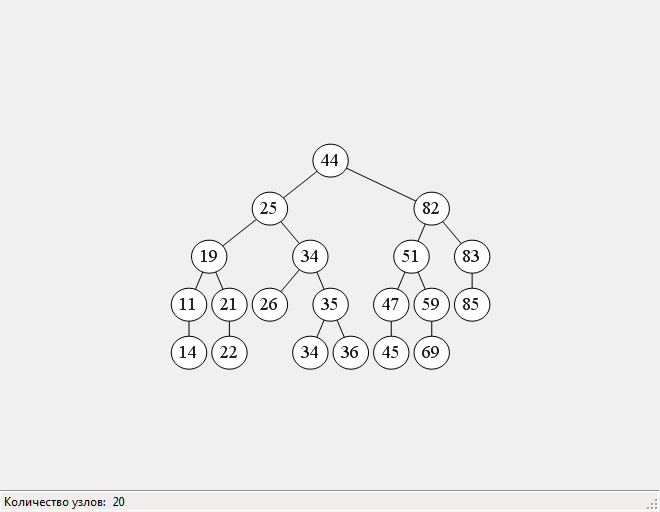
}

# Примеры выполнения программы

ВХОД (ПОСЛЕДОВАТЕЛЬНОСТЬ ЧИСЕЛ):

25 11 51 19 21 34 44 35 47 26 83 85 82 59 22 36 69 45 34 14

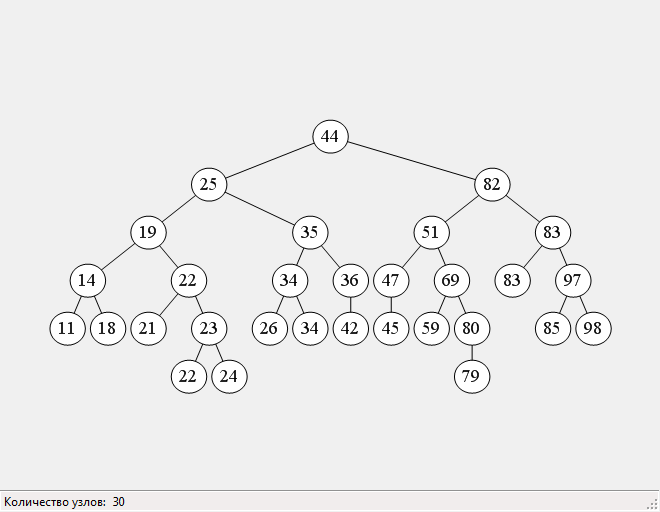
ВЫХОД (ВИЗУАЛИЗИРОВАННОЕ ДЕРЕВО):



ВХОД (ПОСЛЕДОВАТЕЛЬНОСТЬ ЧИСЕЛ):

25 11 51 19 21 34 44 35 47 26 83 85 82 59 22 36 69 45 34 14 22 80 18 83 23 42 24 98 79 97

ВЫХОД (ВИЗУАЛИЗИРОВАННОЕ ДЕРЕВО):



# Вывод по работе

В результате проведенной работы была достигнута поставленная цель: изучение основных методов работы с иерархическими структурами данных.