Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №20**

Дисциплина: «Основы алгоритмизации и программирования»

Тема: “ Бинарные деревья.”

Вариант 21

Выполнил:

Студент группы ИВТ-20-2Б Галинов О.Ю.

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Постановка задачи**

1. Сформировать идеально сбалансированное бинарное дерево.
2. Тип информационного поля char\*. Найти количество элементов дерева, начинающихся с заданного символа.
3. Выполнить обработку дерева в соответствии с заданием, вывести полученный результат.

**Анализ задачи**

**1.** Для решения задач необходимо:

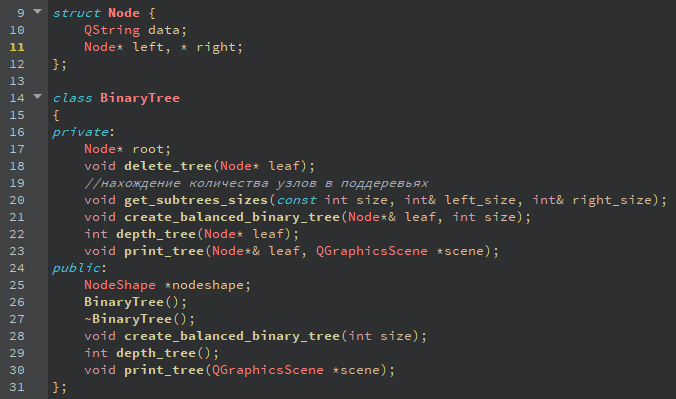
**1.1.** Организовать класс BinaryTree с полем root типа Node – как указатель на узел дерева.

**1.2.** Отрисовка дерева осуществляется через QTCreator, решение задачи через Visual Studio.

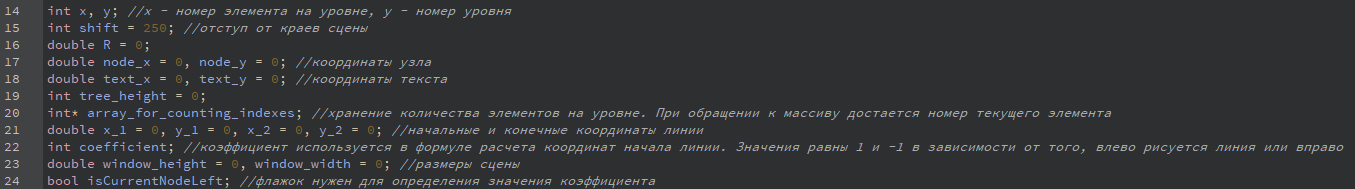
**1.3.** Организовать структуру Node для хранения левого и правого узла, а также значения узла.

**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Класс BinaryTree с полем root типа Node – как указатель на узел дерева.

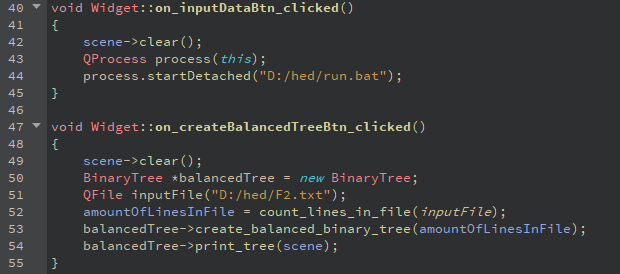


**2.2.** Несколько переменных типа int и double, отвечающие за координаты на форме. И переменную типа bool для создания сбалансированного дерева.



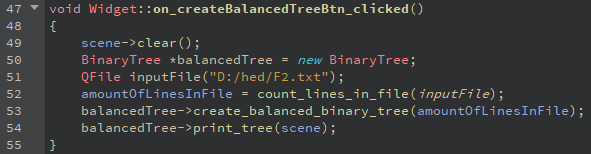
**3.** Для решения задачи данные были представлены в следующем виде:

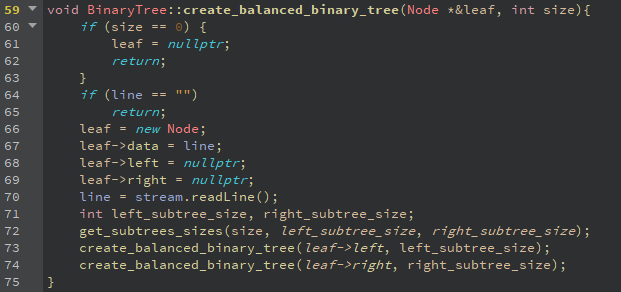
**3.1.** Бинарное дерево создаётся через ввод данных через написанное консольное приложение, которые записываются в файл и далее считываются из него.



**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Дерево выводится при помощи формы.ui. Для его отрисовки создан класс binarytree, в нем считывается файл, в который были записаны все необходимые данные. При нажатии на кнопку Создать дерево считается кол-во строк в файле и это число передается в метод создания дерева. В нем элементы дерева также считываются из файла, в который мы их ввели.



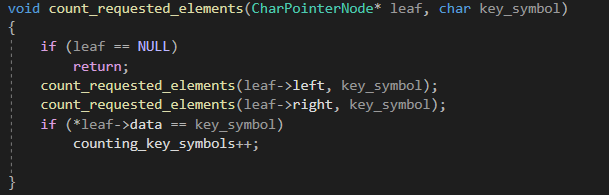


**4.2.** Метод print\_tree необходим для полной отрисовки дерева вместе с узлами и связями. Узлы рисуется по принципу треугольников, где вершина – родительский класс. Метод вызывается рекурсивно для отрисовки каждого узла по отдельности.





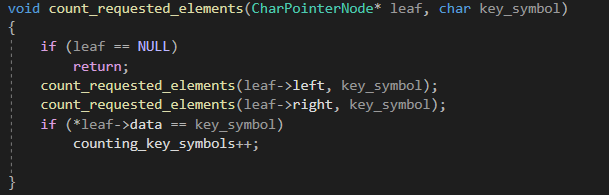
**4.3.** В функции count\_requested\_elements происходит подсчет элементов дерева, начинающихся с одного символа.



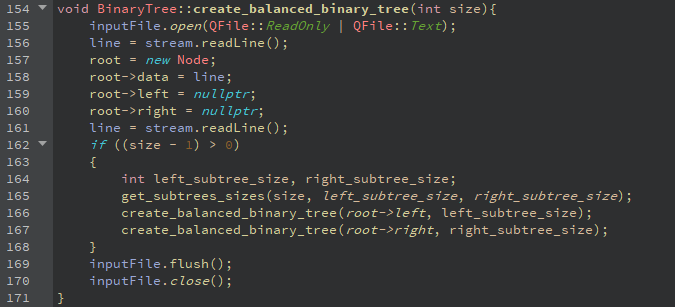


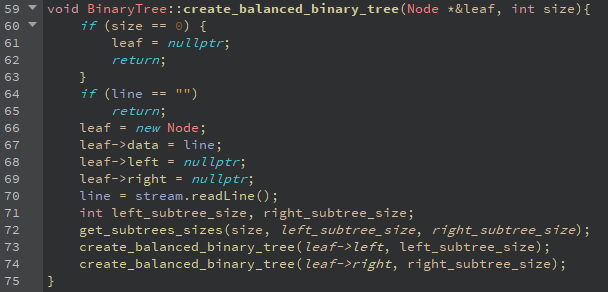
**5.** Поставленные задачи будут решены следующими действиями:

**5.1.** Нахождение кол-ва элементов, начинающихся с одного символа происходит в функции count\_requested\_elements.

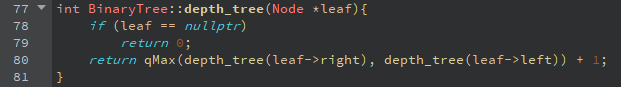


**5.2.** Добавление узлов в дерево обеспечивается методом create\_balanced\_binary\_tree.Метод считывает информацию из файла и передает ее в метод с таким же названием, который создает дерево исходя из его размера.





**5.3.** Вычисление высоты дерева происходит в методе depth\_tree (). Данный метод рекурсивно двигается по правой и по левой вычисляет максимально количество по узлов с обеих сторон и затем возвращает, большее из них увеличенной на 1, чтобы учесть ещё корень дерева.

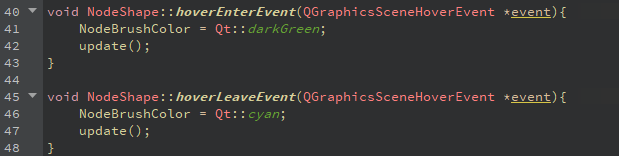


В классе nodeshape задаются данные для отрисовки текста, узлов и их размера. А также события при наведении курсора на узел.

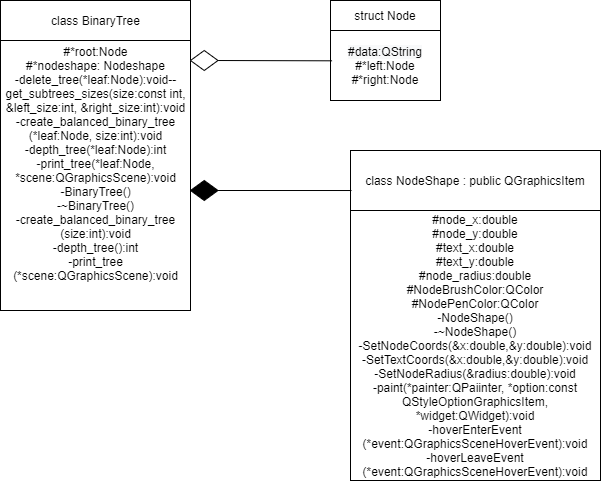
**5.5.** Задание размеров и отрисовка узлов с текстом.



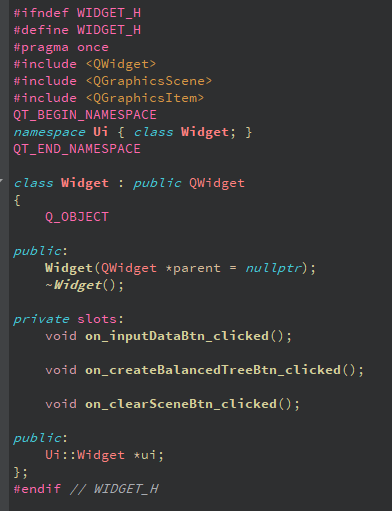
**5.6.** События при наведении курсора и без наведения.

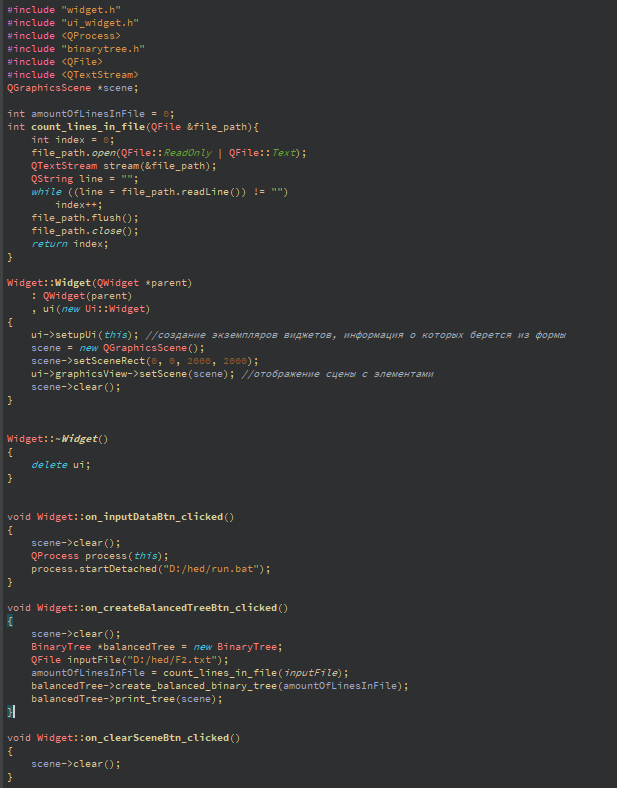


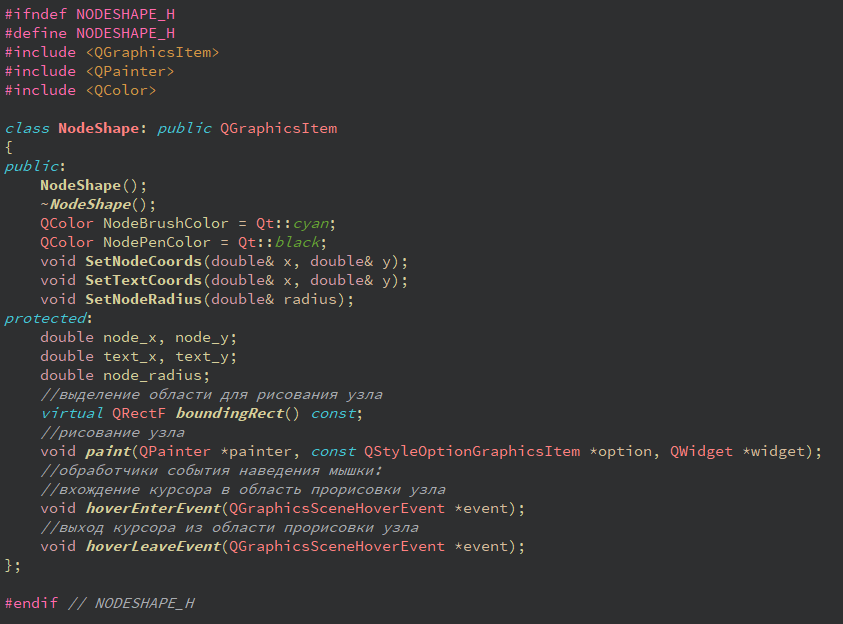
**UML - диаграмма**

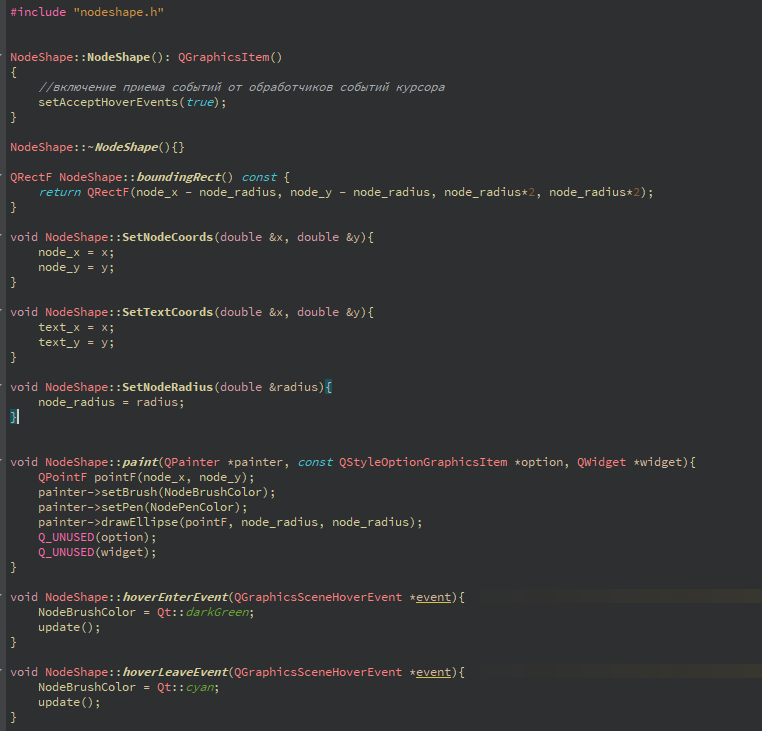


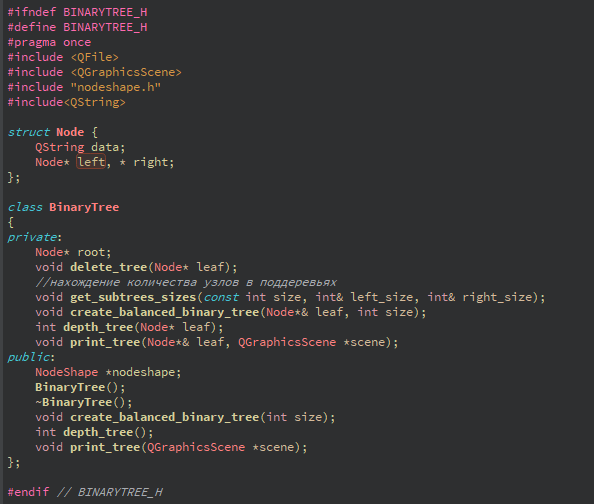
**Код**

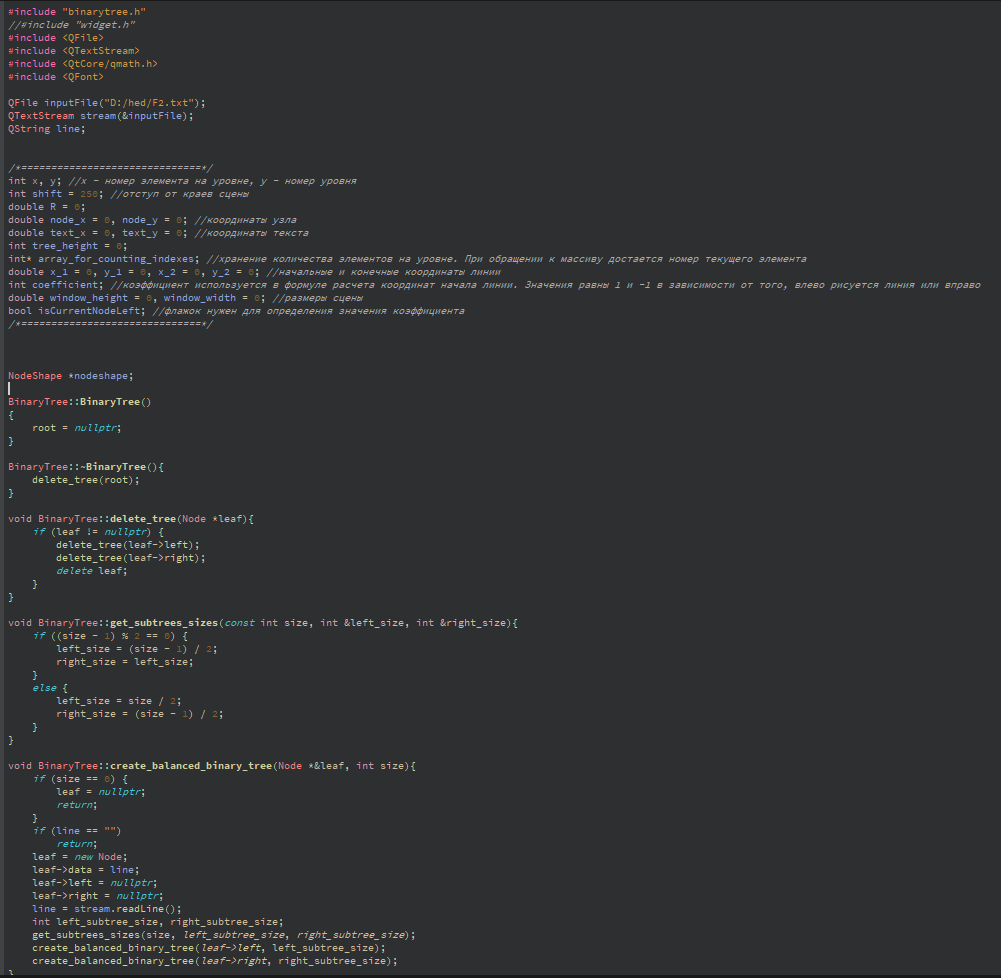








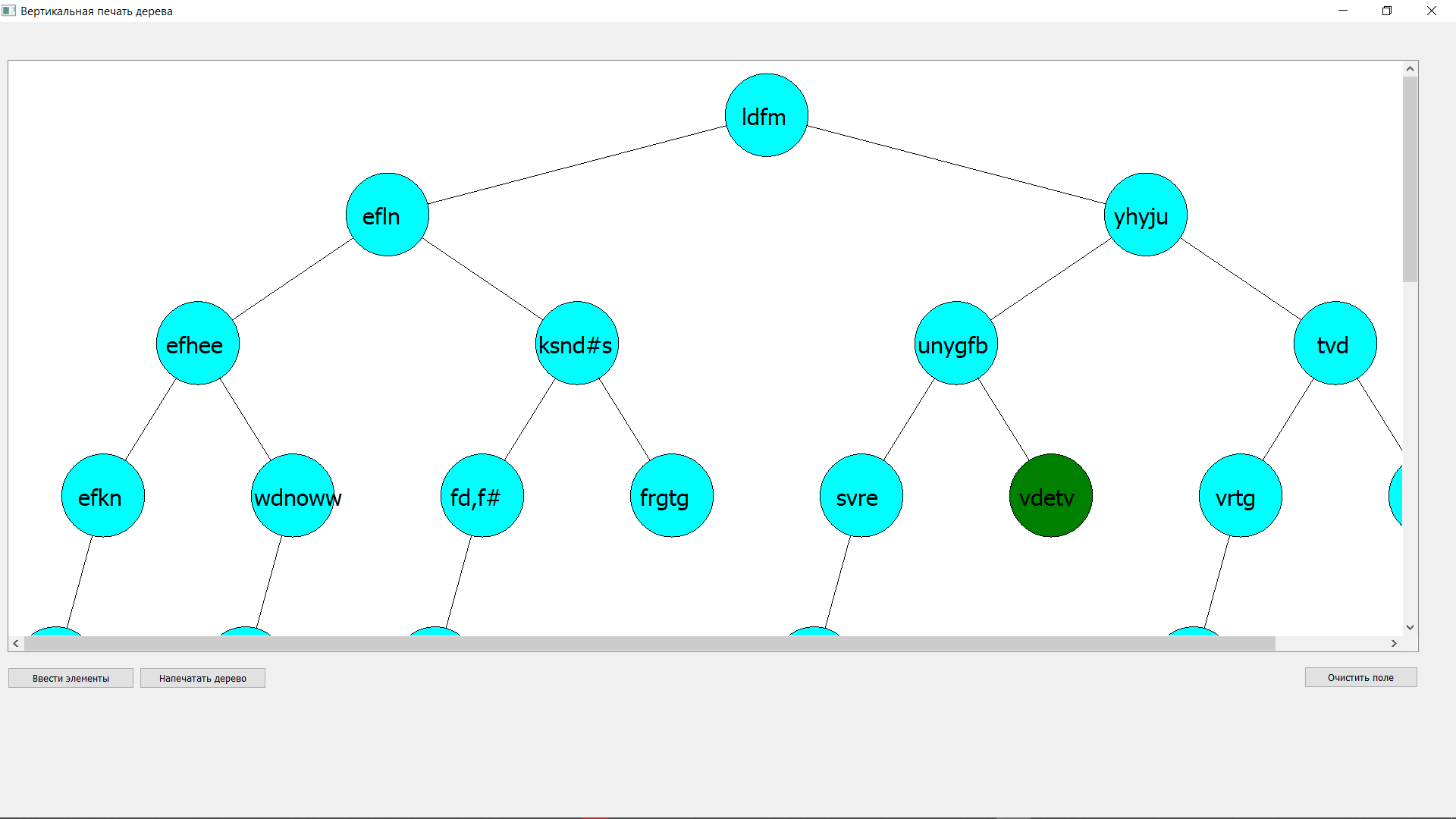


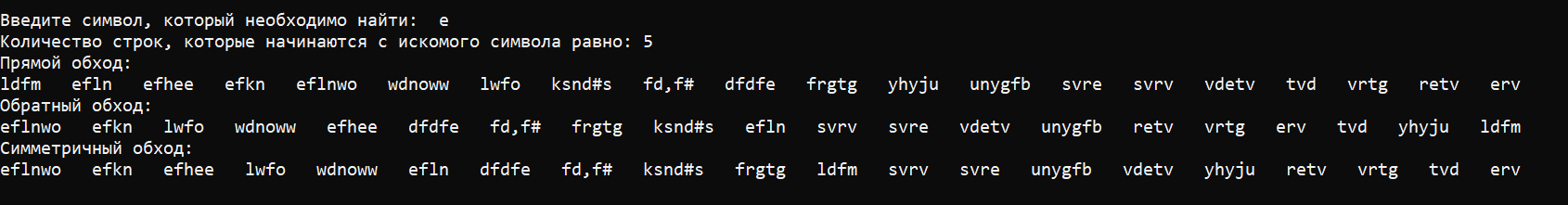






**Скриншоты**



****