

## Отчет по лабораторной работе № 23 по курсу Практикум на ЭВМ

Студент группы М8О-106Б-21 Мезенин Олег Александрович, № по списку 10

Контакты www, e-mail, icq, skype Jktu332@yandex.ru

Работа выполнена: « 26 » апреля 2022 г.

Преподаватель: ст. преп. каф. 806 Дубинин А.В.

Входной контроль знаний с оценкой \_\_\_\_\_

Отчет сдан «    » \_\_\_\_\_ 202 \_\_ г., итоговая оценка \_\_\_\_

Подпись преподавателя \_\_\_\_\_

1. **Тема:** Динамические структуры данных. Обработка деревьев

2. **Цель работы:** Освоить навыки в реализации структуры данных дерево и научиться составлять программы для их обработки

3. **Задание (вариант № 4):** Составить программу на языке Си для построения и обработки упорядоченного двоичного дерева, содержащего узлы типа enum. Основные функции реализовать в виде универсальных процедур или функций. После того, как дерево создано, его обработка должна производиться в режиме текстового меню со следующими действиями: добавление нового узла, текстовая визуализация дерева, удаление узла, определение значения листа двоичного дерева, имеющего минимальную глубину.

4. **Оборудование (лабораторное):**

ЭВМ \_\_\_\_\_, процессор \_\_\_\_\_, имя узла сети \_\_\_\_\_ с ОП \_\_\_\_\_ Мб,  
НМД \_\_\_\_\_ Мб. Терминал \_\_\_\_\_ адрес \_\_\_\_\_. Принтер \_\_\_\_\_  
Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор \_\_\_\_\_ с ОП \_\_\_\_\_ Мб, НМД \_\_\_\_\_ Мб. Монитор \_\_\_\_\_  
Другие устройства \_\_\_\_\_

5. **Программное обеспечение (лабораторное):**

Операционная система семейства \_\_\_\_\_, наименование \_\_\_\_\_ версия \_\_\_\_\_  
интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_  
Система программирования \_\_\_\_\_ версия \_\_\_\_\_  
Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_  
Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_  
Местонахождение и имена файлов программ и данных \_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства \_\_\_\_\_, наименование \_\_\_\_\_ версия \_\_\_\_\_  
интерпретатор команд \_\_\_\_\_ версия \_\_\_\_\_  
Система программирования \_\_\_\_\_ версия \_\_\_\_\_  
Редактор текстов \_\_\_\_\_ версия \_\_\_\_\_  
Утилиты операционной системы \_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_  
Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_

**6. Идея, метод, алгоритм** решение задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Идея основной функции - определение значения листа, имеющего минимальную глубину - в том, чтобы найти рекурсивно узел, являющийся терминированной вершиной и имеющий минимальную глубину. Для реализации этой функции нам понадобится структура `tree_level`, имеющая два поля: сам узел дерева и его глубина. Эту структуру функция должна принимать в качестве аргумента. Возвращать функция будет также значение типа `tree_level`.

Будем вызывать функцию рекурсивно, передавая сначала левое поддерево, затем правое, и сохраняя значения соответственно в переменных `left` и `right`. При каждой передаче структуры `tree_level` будем увеличивать глубину на 1.

Затем будем делать четыре проверки:

- 1) Если `left` и `right` пустые, то возвращаем текущее значение структуры (значит, это лист).
- 2) Если `left` пустой, а `right` не пустой, то возвращаем `right`.
- 3) Если `right` пустой, а `left` не пустой, то возвращаем `left`.
- 4) Если `left` и `right` не пустые, то возвращаем узел с минимальным значением глубины.

Таким образом мы получим лист с минимальной глубиной. Всё, что остаётся сделать - это вывести поле значения полученного листа.

**7. Сценарий выполнения работы** (план работы, первоначальный текст программы в черновике [можно на отдельном листе] и тесты либо соображения по тестированию)

- 1) Выбрать предметную область для составления значений `enum`'а;
- 2) Заполнить `enum`;
- 3) Определить структуру дерева и функций для его обработки;
- 4) Реализовать функции для обработки дерева;
- 5) Реализовать основную функцию интерфейса для пользователя, используя конечный автомат;
- 6) Реализовать функции вывода справки, вывода возможных значений `enum`'а, добавления нового узла, текстовой визуализации дерева, удаления узла, определения значения листа двоичного дерева, имеющего минимальную глубину;
- 7) Протестировать программу.

*Пункты 1-7 отчета составляются строго до начала лабораторной работы.*

*Допущен к выполнению работы. Подпись преподавателя* \_\_\_\_\_

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем)

9. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

№	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

**10. Замечания автора** по существу работы:

**11. Выводы:** Двоичное дерево, у которого для каждой вершины  $t_i$  справедливо утверждение, что все значения левого поддерева меньше значения вершины  $t_i$ , а все значения правого поддерева больше значения вершины  $t_i$ , называется двоичным деревом поиска. Такое дерево предоставляет эффективный поиск, вставку и удаление элементов за  $O(\log n)$ . AVL-деревом называется двоичное дерево, в котором высоты левого и правого поддеревьев отличаются не более, чем на 1. Двоичное дерево называется В-деревом, если в нем нет ни одного узла степени 1.

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента