**Отчет по лабораторной работе №23** по курсу алгоритмы и структуры данных

Студент группы М8О-101БВ-24 Рахматуллин Айдар Рамильевич, № по списку 27

Контакты e-mail

Работа выполнена: « » 2025 г.

Преподаватель: \_\_\_\_\_\_\_ каф. 806 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Отчет сдан « » \_\_\_\_\_\_\_\_\_202 \_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

1. **Цель работы:** Изучить новую структуру данных Деревья и научиться работать с ней, узнать виды деревьев и основные понятия связанные с ними, научиться писать не рекурсивные функции.
2. **Задание** (*вариант №* **27**)**:** Составить программу на языке Си для построения и обработки дерева общего вида или упорядоченного двоичного дерева, содержащего узлы типа float, int, char или enum (для групп 1,8; 3,4; 2,5; 6,7 соответственно). Основные функции работы с деревьями 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 (для двоичного дерева положение нового узла определяется в соответствии с требованием сохранения порядка; для дерева общего вида должен задаваться отец добавляемого узла. Добавляемый узел становится самым младшим сыном);
* текстовая визуализация дерева (значение каждого узла выводится в отдельной строке, с отступом, пропорциональным глубине узла, в порядке старшинства узлов);
* удаление узла (двоичное дерево перестраивается в соответствии с требованием сохранения целостности и порядка; для дерева общего вида удаляется все поддерево, исходящее из удаляемого узла. Должно быть предусмотрено корректное освобождение памяти);
* вычисление значения некоторой функции от дерева (целой или логической), в соответствии с номером варианта. (Функция №22: Определение числа вершин дерева).

1. **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 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ГБ, НМД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ГБ. Монитор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **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** решения задачи(в формах:словесной,псевдокода,графической[блок-схема,диаграмма,рисунок,таблица] или формальные спецификации с пред- и постусловиями)

**Идея**: Реализовать динамическую структуру данных — дерево общего вида, где каждый узел содержит значение типа float и указатели на сына и брата. Дерево обрабатывается через текстовое меню, поддерживающее добавление узлов, текстовую визуализацию, удаление поддерева и подсчёт числа вершин. Для обработки дерева используется стек.

**Метод**:

* **Структура дерева**: Каждый узел node содержит значение val типа float, указатель на первого сына и указатель на следующего брата. Дерево представлено указателем на корень tree.
* **Добавление узла**: Пользователь задаёт путь к родителю (последовательность символов s для сына и b для брата) и значение узла. Новый узел становится самым младшим сыном или братом в указанной позиции.
* **Визуализация**: Дерево обходит стек, выводя значения узлов с отступами, пропорциональными уровню вложенности.
* **Удаление узла**: Удаляется поддерево, начинающееся с узла, указанного путём. Освобождается память для всех узлов поддерева с использованием стека.
* **Подсчёт узлов**: С помощью стека подсчитывается общее количество узлов в дереве.

**Алгоритм**:

1. **Добавление узла**:
   * Прочитать путь (строку из s и b) и значение (float).
   * Пройти по пути от корня, переходя к сыну (son) или брату (bro).
   * Создать новый узел с заданным значением и добавить его как самого младшего сына или брата в конечной позиции.
2. **Визуализация дерева**:
   * Использовать стек для хранения узлов и их уровней.
   * Начать с корня, добавляя узлы в стек (сыновья с увеличением уровня, братья с тем же уровнем).
   * Извлекать узлы из стека, выводить значения с отступами, пропорциональными уровню.
3. **Удаление узла**:
   * Пройти по заданному пути до целевого узла.
   * Извлечь поддерево, начиная с этого узла.
   * Освободить память поддерева с помощью стека, обходя узлы.
4. **Подсчёт узлов**:
   * Использовать стек для обхода всех узлов дерева.
   * Подсчитывать каждый узел, добавляя в стек сыновей и братьев.

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

1. **Анализ задания и проектирование**:
   * Изучить требования: реализация дерева общего вида с узлами типа float, текстовое меню для добавления узлов, визуализации, удаления поддерева и подсчёта числа вершин (вариант 22).
   * Определить структуру дерева: каждый узел node содержит значение float, указатели на сына son и брата bro. Путь к узлу задаётся строкой из символов s сын и b брат.
   * Разработать модули: tree.h, tree.c, stack.h и stack.c, main.c.
   * Использовать стек для обхода дерева вместо рекурсии.
2. **Реализация стека (**stack.h**,** stack.c**)**:
   * Создать структуру stack\_tree для хранения элементов stack\_item (узел дерева и уровень вложенности).
   * Реализовать функции: stree\_create, stree\_destroy, stree\_is\_empty, stree\_push, stree\_pop, grow.
   * Обеспечить динамическое расширение стека при необходимости.
3. **Реализация дерева (**tree.h**,** tree.c**)**:
   * Реализовать функции:
     + add: добавление узла по заданному пути.
     + print\_tree: текстовая визуализация с отступами, пропорциональными уровню.
     + remove\_node: удаление поддерева по пути с освобождением памяти.
     + free\_tree: освобождение всего дерева.
     + count\_nodes: подсчёт числа узлов.
   * Использовать стек для итеративного обхода в print\_tree, free\_tree и count\_nodes, избегая рекурсии.
   * Обеспечить корректное управление памятью.
4. **Реализация основной программы (**main.c**)**:
   * Создать текстовое меню с пятью опциями: добавление узла, визуализация, удаление, подсчёт узлов, выход.
   * Реализовать обработку ввода: выбор действия, значение узла float, путь строка из s и b.
   * Добавить проверку корректности ввода и обработку ошибок.
   * Обеспечить завершение программы с освобождением памяти при выборе выхода.
5. **Отладка и тестирование**:
   * Проверить корректность операций с деревом: добавление узлов, визуализация, удаление, подсчёт.
   * Протестировать краевые случаи: пустое дерево, некорректный путь, большое дерево, EOF.
   * Подготовить тесты для проверки всех функций меню и устойчивости программы.

### Соображения по тестированию

* **Цели тестирования**:
  + Проверить корректность добавления узлов по заданным путям, включая сложные структуры (множество сыновей и братьев).
  + Убедиться, что визуализация правильно отображает дерево с отступами, соответствующими уровням вложенности.
  + Подтвердить, что удаление поддерева корректно обновляет структуру дерева и освобождает память.
  + Проверить подсчёт числа узлов для деревьев разного размера.
  + Гарантировать устойчивость программы к ошибкам ввода (неверный выбор, некорректный путь, неправильный формат числа).
* **Подход к тестированию**:
  + Использовать интерактивный ввод через терминал для проверки работы меню.
  + Проверить разные тестовые сценарии с различными структурами дерева: пустое дерево, дерево с одним узлом, сложное дерево с несколькими уровнями.
  + Проверить краевые случаи: некорректный ввод, попытка удаления несуществующего узла, обработка EOF.
* **Критерии успешности**:
  + Все операции меню выполняются корректно и соответствуют ожидаемому выводу.
  + Дерево визуализируется с правильными отступами и в порядке старшинства.
  + Удаление поддерева не нарушает структуру дерева, память освобождается.
  + Подсчёт узлов возвращает правильное значение.
  + Программа устойчиво обрабатывает ошибки ввода и завершает работу без сбоев.

*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,подписанныйпреподавателем)

**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 других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

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

1. **Замечания автора** по существу работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Выводы**

Реализовал динамическую структуру данных — дерево общего вида с узлами типа float, используя модульное программирование на языке C. Разработал текстовое меню для добавления узлов, визуализации дерева, удаления поддерева и подсчёта числа вершин. Применил стек для итеративного обхода дерева, избегая рекурсии и упрощая управление памятью. Обеспечил корректное освобождение памяти при удалении узлов и завершении программы. Научился обрабатывать ввод путей (последовательности s и b) и проверять корректность пользовательского ввода. Проверил функциональность программы на тестах, подтвердив правильность визуализации, удаления и подсчёта узлов. Освоил проектирование модульных программ с разделением на интерфейсы и реализацию.

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_