

Постановка задачи

1. Разработать статическую библиотеку, реализующую определенный вариантом задания абстрактный тип данных.
2. Разработать демонстрационную программу – консольное приложение, обеспечивающее ввод данных из файла (файлов), их обработку и вывод в файл (файлы); имена файлов передаются в качестве параметров командной строки.

Варианты заданий

1. Дерево (с узлами произвольной степени).
2. AVL-дерево.
3. Красно-черное дерево.
4. B-дерево.
5. Двоичная куча.
6. Ненаправленный граф, представление матрицей смежности.
7. Ненаправленный граф, представление списком смежных вершин.
8. Ориентированный граф, представление матрицей смежности.
9. Ориентированный граф, представление списком смежных вершин.
10. Хеш-таблица с ключами-строками.
11. Префиксное дерево.
12. Списки пропусков.
13. Файловая система в памяти.
14. Менеджер памяти для малых объектов (см., например, [2] параграф 12.2).
15. Фильтр Блума.
16. Диаграммы бинарного выбора (см., например, [1] параграф 12.7, [4] параграф 7.1.4).
17. Синтаксическое дерево регулярного выражения.
18. Длинная целочисленная арифметика (операции сложения, вычитания, умножения, деления, вычисления остатка, преобразования в базовые типы C, формирования и разбора строкового представления).

Требования к ПО

1. Язык разработки – C.
2. Реализация абстрактного типа данных должна использовать динамическое выделение памяти, при этом должна быть предусмотрена функция деинициализации, обеспечивающая освобождение всей выделенной памяти.
3. Библиотека и демонстрационная программа должны быть снабжены модульными тестами.
4. Разработанный исходный код должен компилироваться gcc без ошибок и предупреждений со следующими параметрами: `-std=c11 -pedantic -Wall -Wextra`.
5. Сборка библиотеки, демонстрационной программы и модульных тестов должна осуществляться утилитой make.

Требования к отчету

Отчет должен содержать:

- развернутую формулировку задачи;
- описание API библиотеки;
- описание форматов файлов, параметров командной строки;
- руководство программиста;
- список использованной литературы.

Руководство программиста должно содержать пояснения, необходимые для понимания организации библиотеки и демонстрационной программы.

Рекомендуемая литература

1. Ахо, Альфред В., Лам, Моника С, Сети, Рави, Ульман, Джеффри Д. Компиляторы: принципы, технологии и инструментарий, 2-е изд.
2. Ахо, Альфред В., Хопкрофт, Джон, Ульман, Джеффри, Д. Структуры данных и алгоритмы.
3. Вирт Никлаус. Алгоритмы + Структуры данных = Программы.
4. Кнут, Дональд Э. Искусство программирования. Том 4, А. Комбинаторные алгоритмы. Часть 1.
5. Кормен, Томас Х., Лейзерсон, Чарльз И., Ривест, Рональд Л., Штайн, Клиффорд. Алгоритмы: построение и анализ, 2-е издание.
6. Седжвик Роберт. Фундаментальные алгоритмы на С. Анализ/ Структуры данных/ Сортировка/ Поиск/ Алгоритмы на графах.
7. Hanson, David R. C interfaces and implementations: techniques for creating reusable software.