Лабораторная работа № 3а «Работа с просматриваемыми и упорядоченными таблицами»

Введение

Необходимо спроектировать и разработать на языке С:

- 1. Прикладную программу, позволяющую пользователю в диалоговом режиме работать с таблиней.
- 2. Библиотеку, предоставляющую функциональность по работе с таблицей, размещенной в основной памяти (в оперативной памяти).

Структура таблицы

Таблица задаётся структурой:

```
struct Table {
    // указатель на пространство ключей
    KeySpace *ks;

    // опциональное поле, ограничивающее размер пространства ключей,
    // его наличие определяется типом организации соответствующего пространства,
    // в соответствии с условиями индивидуального задания

    // размер области пространства ключей
    IndexType msize;

    // опциональное поле с текущим количеством элементов
    // в пространстве ключей,
    // его наличие определяется типом организации соответствующего пространства,
    // в соответствии с условиями индивидуального задания

    // количество элементов в области пространства ключей
    IndexType csize;
};
```

Операции, поддерживаемые таблицей

Должны быть предусмотрены следующие операции:

- 1. вставка нового элемента в таблицу с соблюдением ограничений на уникальность значений ключевой информации;
- 2. удаление из таблицы элемента по заданному значению ключа;
- 3. поиск в таблице элемента по заданному значению ключа, результатом поиска должны быть копии всех найденных элементов со значениями ключей;

- 4. вывод содержимого таблицы в стандартный поток;
- 5. импорт данных из текстового файла;
- 6. особые операции, в соответствии с индивидуальным заданием.

Дополнительное задание

Спроектировать и реализовать альтернативный программный интерфейс для работы с таблицей, использующий итераторы:

- В библиотеке, реализующей работу с таблицей, определить тип данных, являющийся итератором, и реализовать функции, необходимые для работы с ним:
 - создание итератора (при необходимости);
 - удаление итератора (при необходимости);
 - сравнение итераторов (функция должна вернуть 1, если итераторы указывают на один и тот же элемент одной таблицы, в противном случае — 0);
 - получение итератора, указывающего на первый элемент таблицы;
 - получение итератора, указывающего на следующий элемент таблицы;
 - получение итератора, указывающего на последний элемент таблицы.
- Разработать для библиотеки, реализующей работу с таблицей, второй (дополнительный) набор функций для вставки, удаления и поиска, при этом:
 - Возвращаемым значением для функции вставки должен являться итератор, указывающий на вставленный элемент.
 - Возвращаемым значением для функции удаления должен являться итератор, указывающий на следующий после удаляемого элемент таблицы. В качестве одного из параметров функция удаления должна принимать итератор, указывающий на удаляемый элемент.
 - Возвращаемым значением для функции поиска должен являться итератор, указывающий на найденный элемент.
- Добавить в диалоговую программу возможность применения итераторов. Выбор реализации (с использованием и без использования итераторов) должен осуществляться на этапе сборки.

Примечания

- 1. Логически законченные части алгоритма решения задачи должны быть оформлены в виде отдельных функций с параметрами. Использование глобальных переменных не допускается.
- 2. Функции для работы с таблицами не должны быть диалоговыми, т. е. они должны принимать все необходимые данные в качестве параметров и возвращать результат работы в виде соответствующих структур данных и кодов ошибок (исключение: функции вывода таблицы в стандартный поток вывода или записи файл).
- 3. Диалоговые функции должны использовать описанные выше функции.
- 4. Программа должна осуществлять проверку корректности вводимых данных.
- 5. В случае возникновения ошибочных ситуаций при выполнении операций с таблицами программа должна выводить соответствующие сообщения, после чего продолжать работу.

- 6. Для сборки программы и библиотек должна использоваться система сборки (например: Make или CMake).
- 7. Библиотеки и прикладная программа должны собираться независимо друг от друга.
- 8. Программа должна корректным образом работать с памятью, для проверки необходимо использовать соответствующие программные средства, например: valgrind, санитайзеры, встроенные в IDE средства и т.д.