Лабораторная работа №2

**Условие (вариант 7):**

Создать таблицу, содержащую список литературы. Для каждой книги необходимо указать фамилию автора, название книги, издательство, количество страниц, вид литературы (1: техническая – отрасль, отечественная, переводная, год издания; 2: художественная – роман, пьеса, стихи; 3: детская – сказки, стихи). Также необходимо вывести на экран список отечественной технической литературы по указанной отрасли.

**Назначение**

Программа для хранения и сортировки данных о литературе разных видов, нахождения литературных произведений технического вида по определенно заданной отрасли.

**Область применения**

Использоваться может в библиотеках, книжных магазинах, интернет-магазинах – везде, где необходимо сортировать и искать нужную литературу.

**Сроки исполнения**

3 недели, начиная с 24.09.2018

**Основание для разработки**

Заказчик – преподаватель Силантьева Александра Васильевна

Исполнитель – студент Ильясов Идрис Магомет-Салиевич, группа ИУ7-33Б

Основание – лабораторная работа №2 по предмету типы и структуры данных

**Назначение разработки**

Общая концепция системы: консольное приложение.

Описание функционала системы: взаимодействие пользователя с программой осуществляется через консоль с помощью меню. Начальные данные загружаются из готового файла, где уже хранится не менее 40 записей. При завершении программы при необходимости данные можно записать в тот же файл.

Пользователь взаимодействует с программой, выбирая один из предложенных пунктов меню, отображаемых в консоли. Вводятся в числовых полях целые числа, в вариантных полях – номер выбранного варианта. После введенных неправильных значений в консоли высветится предупреждающее сообщение об ошибке. После ошибки происходит возврат в начальное меню.

**Требования к программе**

Требования к информационным структурам и методам решения

Программа полностью написана на языке Си. Данные, хранящиеся в базе данных, представляют собой структуры со следующими полями:

**struct** technical

{

**char** \*industry\_type;

**char** \*country;

**int** published\_year;

};

**struct** artistic

{

**char** \*artistic\_genre;

};

**struct** children

{

**char** \*children\_genre;

};

**struct** table\_inf

{

**char** \*surname;

**char** \*book\_name;

**char** \*publishing\_house;

**int** pages;

**char** \*literature\_type;

**union**

{

**struct** technical;

**struct** artistic;

**struct** children;

}

};

};

};

* Фамилия автора
* Название книги
* Издательство
* Количество страниц
* Вид литературы (техническая, художественная, детская)
* Вариантное поле, зависящее от вида литературы
* Техническая
* Художественная
* Детская

**Виды литературы представляют собой следующие структуры с полями:**

Техническая:

* Отрасль
* Инженерия
* Металлургия
* Биология
* География
* Авиация
* Физика
* Страна
* Отечественная
* Переводная
* Год публикации

Художественная:

* Жанр
* Роман
* Пьеса
* Стихи

Детская:

* Жанр
* Сказки
* Стихи

**Требование к функциональным характеристикам**

Программа позволяет осуществлять управление базой данных, в которой хранится список литературных произведение разных видов, жанров. Взаимодействие осуществляется через консоль, интерфейс представлен в виде меню.

Пользователь через данное меню может:

* Просмотреть базу данных
* Добавить запись
* Удалить запись
* Произвести поиск технической отечественной литературы по заданной отрасли
* Отсортировать таблицу (сортировка производится методом пузырька, но также можно при помощи qsort)
* Отсортировать таблицу ключей методами пузырька и qsort и замерить время для сравнительного анализа
* Вывести отсортированную базу данных
* Просмотреть сравнение эффективности сортировок
* Выйти из программы

Сама база данных загружается из файла init\_data.txt при запуске программы (название файла необходимо указать в параметрах командной строки). Все изменения при желании пользователя записываются в этот же файл.

**Требования к надежности**

Любой некорректный ввод возвращает пользователя в начальное меню с прерыванием предыдущей операции и выводом сообщения об ошибке. При вводе некорректного значения в главном меню, программа завершает свою работу.

**Структуры данных**

См. раздел «Требования к информационным структурам и методам решения. Сортировка таблицы выполняется методом пузырька и qsort.

**Модули и тесты**

Программа не нуждается в описании модулей и модульных тестах.

Пользовательские функциональные тесты:

|  |  |  |
| --- | --- | --- |
| **Входные данные** | **Вывод** | **Класс эквивалентности** |
| Меню: 1 | Выбранный пункт меню | Корректный ввод в меню |
| Меню: 78 | Wrong input | Некорректный ввод целого числа в меню |
| Меню: а | Wrong input | Некорректный ввод в меню |
| Ввод строки: любая строка | Успешный ввод | Корректный ввод строки |
| Ввод числа (дата публикации): 2019 | Wrong input | Некорректный ввод даты публикации |
| Ввод числа (дата публикации): 2008 | Успешный ввод | Корректный ввод даты публикации |
| Ввод числа (количество страниц): 205 | Успешный ввод | Корректный ввод количества страниц |
| Ввод числа (количество страниц): -50 | Wrong input | Некорректный ввод количества страниц |
| Ввод вида литературы: econafa | Wrong input | Некорректный ввод вида литературы |
| Ввод вида литературы: technical | Успешный ввод | Корректный ввод вида литературы |
| Ввод жанра для какого-либо вида литературы: poems | Успешный ввод | Корректный ввод жанра |
| Ввод жанра для какого-либо вида литературы: as | Wrong input | Некорректный ввод вида литературы |

**Алгоритм**

Ввод: choice

Вывод: -

Загрузить данные

Пока check == 1

Считать choice

Если choice == 0

Вывести таблицу на экран

Если choice == 1

Добавить запись

Если choice == 2

Удалить запись

Если choice == 3

Произвести поиск отечественной технической литературы по заданной отрасли

Если choice == 4

Произвести сортировку и замерить время

Если choice == 5

Произвести сортировку по массиву ключей

Если choice == 6

Вывести отсортированную таблицу по массиву ключей

Если choice == 7

Сравнить производительность (эффективность)

Если choice == 8

Записать в файл

Если choice == 9

Выйти из программы

Все если

Все пока

Выход

**Результаты сравнения скорости и потребления памяти:**

Размер таблицы: 40

|  |  |  |
| --- | --- | --- |
| Сравнение времени | Сортировка таблицы | Сортировка массива ключей |
| Пузырек с флагом | 100% | 74.93% |
| qsort | 100% | 67.21% |

Размер таблицы: 240

|  |  |  |
| --- | --- | --- |
| Сравнение времени | Сортировка таблицы | Сортировка массива ключей |
| Пузырек с флагом | 100% | 68.23% |
| qsort | 100% | 86.13% |

Выигрыш по памяти составляет 7.142% при использовании таблицы ключей во время сортировки, вместо сортировки самой таблицы.

Мы можем увидеть, что для пузырька с флагом использование подхода с использованием таблицы (массива) ключей при увеличении размеров таблицы дает прирост скорости сортировки, хотя затраты памяти не увеличиваются. Для сортировки qsort данный метод, наоборот, дает обратный эффект: скорость сортировки увеличивается не так сильно, но затрачиваемая память не меняется.

**Вывод**

Во время выполнения данной лабораторной работы я научился работать со структурами и объединениями в СИ. Также я научился работать с базами данных через таблицу ключей и сравнил два алгоритма сортировки и их эффективность по времени и по памяти. Тип «запись» целесообразно использовать, так как данный тип позволяет хранить все данные объекта в одной переменной. Вариантная часть позволяет сэкономить память при использовании разных данных, потому что для них не нужно выделять отдельные поля.

**Контрольные вопросы**

1. Как выделяется память под вариантную часть записи?

Под вариантную часть записи выделяется участок, соответствующий размеру самого большого поля вариантной части.

2. Что будет, если в вариантную часть ввести данные, не соответствующие описанным?

Вариантная часть не проверяется при компиляции.

3. Кто должен следить за правильностью выполнения операций с вариантной частью записи?

Вариантная часть не проверяется при компиляции, поэтому следить за правильностью выполнения операций с ней должен программист.

4. Что представляет собой таблица ключей, зачем она нужна?

Таблица ключей – это таблица, содержащая ключи и позиции соответствующих им записей исходной таблицы. Она нужна для экономии времени при сортировке данных, поскольку переставлять в ней записи быстрее, чем в исходной таблице.

5. В каких случаях эффективнее обрабатывать данные в самой таблице, а когда – в таблице ключей?

Таблица ключей используется при больших объемах данных, когда выигрыш по времени значительно превышает дополнительные затраты памяти. Но в случае, если в таблице мало полей, то использовать таблицу ключей также не рационально.

6. Какие способы сортировки предпочтительнее для обработки таблиц и почему?

Для обработки таблиц предпочтительнее использовать те способы сортировки, в которых выполняется наименьшее количество перестановок.