

	<p>Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования</p> <p>«Московский государственный технический университет имени Н.Э. Баумана</p>
---	---

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«Работа с типом данных «запись» (структура)»

Студент Фролов Евгений

Группа ИУ7 – 35Б

2020 г.

Описания условия задачи

Создать таблицу, содержащую не менее 40-ка записей (тип - запись с вариантами). Упорядочить данные в ней по возрастанию ключей, где ключ – минимальная цена, используя: а) саму таблицу, б) массив ключей (возможность добавления и удаления записей в ручном режиме).

Ввести репертуар театров, содержащий: название театра, спектакль, режиссер, диапазон цены билета, тип спектакля: детский – для какого возраста, тип (сказка, пьеса, музыкальный); взрослый – тип (пьеса, драма, комедия); музыкальный – композитор, страна, минимальный возраст, продолжительность). Вывести список всех музыкальных спектаклей для детей указанного возраста с продолжительностью меньше указанной.

Цель работы: приобрести навыки работы с типом данных «запись» («структура»), содержащим вариантную часть, и с данными, хранящимися в таблицах. Оценить относительную эффективность программы (в процентах) по времени и по используемому объему памяти в зависимости от используемого алгоритма и от объема сортируемой информации.

Входные данные:

На входе программа получает число записей в таблице и саму таблицу.

Количество записей вводит пользователь с консоли.

название театра

спектакль

режиссер

нижняя цена

верхняя цена

тип спектакля:

детский – для какого возраста, тип (сказка, пьеса, музыкальный);

взрослый – тип (пьеса, драма, комедия);

музыкальный – композитор, страна, минимальный возраст, продолжительность.

Выходные данные:

Программа выводит пользовательское меню для дальнейшей работы после каждого успешно выполненного пункта. Также на выходе программа может вывести (отсортированную так и не отсортированную) таблицу, таблицу ключей, вывести время сортировки.

Задачи реализуемые программой:

1. Вывести таблицу
2. Добавить запись
3. Удалить запись
4. Вывести список всех музыкальных спектаклей для детей указанного возраста с продолжительностью меньше указанной
5. Вывести отсортированную таблицу ключей
6. Отсортировать таблицу
7. Вывод таблицы используя ключи
8. Эффективности сортировок
0. Выйти

Возможные аварийные ситуации и ошибки пользователя

- Запрос на выполнение несуществующего пункта меню
- При добавлении новой записи неверный ввод
- Попытка удалить несуществующие поля
- Не существует файл (First.txt, theathre_1.txt, theathre_2.txt)

Описание структур данных

Таблица репертуара театров — массив типа **struct theatre_t arr[NMAX]**;

-где NMAX описан в define = 50

//структура для хранения данных о театре

```
struct theatre_t{
    int number;
    char name[30]; //название театра
    char performance[30]; //спектакль
    char producer[30]; //режиссер
    int low_price;//нижняя цена
    int high_price;//верхняя цена
    int flag;
    union{
        struct{
            int year;//возраст
            char type_for_child [30]; //тип детского спектакля
        } children;
        struct{
            char type_for_adult [30]; //тип взрослого спектакля
        } adult;
        struct{
            char composer[30]; //композитор
            char country[30]; //страна
            int min_year; //минимальный возраст
            int duration; //продолжительность
        } music;
    }staff;
};
```

```
//структура для хранения ключей
struct keys_t{
    int ID;
    int pricelow;
};
```

Обращение к программе: через консоль командой ./main.exe

Описание алгоритма

1. Принять на ввод кол-во считываемых строк из существующего файла (First.txt) с данными, проверяем на правильность ввода. Если ошибок нет, при помощи функции read_file(file,n,arr) считать данные в массив структур.
2. Вывод меню при успешном считывании кол-ва строк.
3. Выполнить соответствующие команды из предложенных в меню, пока ввод не равен "0" – выход из программы.

Тесты

Ввод	Вывод
Пункт меню	Выполнение пункта меню
Пункт, которого нет в меню	Вывод меню снова
Добавить запись (некорректные данные)	WRONG input
Чтение из несуществующего файла	Can't open file.
Удалить записи по позиции(некорректный диапазон)	Нет такой строки
Статистика эффективности сортировок	Вывод таблицы со статистикой

Ответы на вопросы

1. Как выделяется память под вариантную часть записи?

Элементы вариантной части хранятся в одном и том же участке памяти. Выделяется область памяти, равная размеру максимального поля вариантных частей.

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

Так как при компиляции тип данных вариантной части не проверяется, то произойдет ошибка.

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

За правильностью выполнения операций с вариантной частью записи должен следить программист.

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

Таблица ключей содержит поле по которому осуществляется сортировка и соответствующий ему индекс в исходной таблице. Если необходимо отсортировать большое количество данных, сортируется только таблица ключей, а исходная таблица остается неизменной.

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

Если ключ — сложный тип на обработку которого требуется время не сильно отличающиеся от времени обработки таблицы, то эффективнее обрабатывать саму таблицу. Если в таблице много записей (или записи большого размера), удобно использовать таблицу ключей.

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

Предпочтительнее способы, в которых над таблицей выполняется наименьшее количество действий, так как обработка большого количества данных влечет большие затраты по времени.

Выводы

Посмотрим на вывод таблицы времени сортировки.

	Keys	Table
TIME of inser sort 2000 elements :	0.002000s	0.010000s
TIME of buble sort 2000 elements :	0.004000s	0.031000s
TIME of inser sort 900 elements :	0.001000s	0.002000s
TIME of buble sort 900 elements :	0.001000s	0.006000s

Структура theatre_t – 1746 Структура keys_t – 86

Таким образом сортируя таблицу ключей вместо самой таблицы — при потере памяти в 4,6%, можно отсортировать в 4-8 раз быстрее.

Из данного результатов видно, что сортировка пузырьком менее эффективна, чем сортировка вставками и при использовании ключей, и самой таблицы.

По итогу получается, что при минимальной потере памяти, мы можем ускорить время работы программы.