|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Типы и структуры данных.**

**Лабораторная работа №2.**

**«Записи с вариантами. Обработка таблиц»**

Студент **Леонов Владислав Вячеславович**

Группа **ИУ7-36Б**

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Леонов В.В.

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Силантьева А.В.

*подпись, дата фамилия, и.о.*

*2020 г.*

Оглавление

[Условие и техническое задание 3](#_Toc53531586)

[Цель работы 3](#_Toc53531587)

[Задание 3](#_Toc53531588)

[Входные данные 3](#_Toc53531589)

[Обращение к программе 3](#_Toc53531590)

[Функции программы 4](#_Toc53531591)

[Аварийные ситуации 4](#_Toc53531592)

[Реализация 5](#_Toc53531593)

[Структуры данных 5](#_Toc53531594)

[ Структура ***show\_t*** 5](#_Toc53531595)

[ Структура ***type\_t*** 6](#_Toc53531596)

[ Структура ***kid\_show\_t*** 6](#_Toc53531597)

[ Структура ***adult\_show\_t*** 7](#_Toc53531598)

[ Структура ***music\_show\_t*** 7](#_Toc53531599)

[ Структура ***key\_t*** 8](#_Toc53531600)

[ Таблицы репертуара и ключей 8](#_Toc53531601)

[Алгоритм 9](#_Toc53531602)

[Тестирование 10](#_Toc53531603)

[Аварийные ситуации 10](#_Toc53531604)

[Штатное поведение 10](#_Toc53531605)

[Оценка эффективности сортировок 12](#_Toc53531606)

[Контрольные вопросы 13](#_Toc53531607)

[1. Как выделяется память под вариантную часть записи? 13](#_Toc53531608)

[2. Что будет, если в вариантную часть ввести данные, не соответствующие описанным? 13](#_Toc53531609)

[3. Кто должен следить за правильностью выполнения операций с вариантной частью записи? 13](#_Toc53531610)

[4. Что представляет собой таблица ключей, зачем она нужна? 13](#_Toc53531611)

[5. В каких случаях эффективнее обрабатывать данные в самой таблице, а когда – использовать таблицу ключей? 13](#_Toc53531612)

[6. Какие способы сортировки предпочтительнее для обработки таблиц и почему? 14](#_Toc53531613)

[Вывод о проделанной работе 15](#_Toc53531614)

# Условие и техническое задание

## 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.

## Задание

Создать таблицу, содержащую не менее 40-ка записей (тип – запись с вариантами (объединениями)). Упорядочить данные в ней по возрастанию ключей, двумя алгоритмами сортировки, где ключ – любое невариантное поле (по выбору программиста), используя: а) саму таблицу, б) массив ключей. (Возможность добавления и удаления записей в ручном режиме обязательна).

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

При сортировке используется поле спектакль.

## Входные данные

1. **Файл**. Для ввода данных таблицы репертуара используется файл, содержащий соответствующие поля (предполагается, что каждое отдельное поле содержится в отдельной строке и что длина любой строки не превышает 64 символа).
2. **Опция меню** (целое число от 0 до 13).
3. **Числовые или строковые данные** (тип данных зависит от заполняемого поля полям таблицы репертуара).

## Выходные данные

1. Файл с таблицей репертуара, хранящейся в памяти на момент выхода, в случае успешного завершения программы.
2. Время сортировки таблицы ключей и таблицы репертуара разными методами.
3. Таблица репертуара, таблица ключей и результаты поиска.

## Обращение к программе

Обращение к программе **app.exe** осуществляется через запуск из терминала.

## Функции программы

1. Выполнить чтение таблицы репертуара театров из файла.
2. Очистить таблицу репертуара
3. Добавить запись в таблицу репертуара
4. Удалить запись из таблицы репертуара
5. Печать текущей таблицы репертуара, без использования таблицы ключей
6. Печать текущей таблицы репертуара, с использованием таблицы ключей
7. Печать текущей таблицы ключей (поле - название спектакля)
8. Сортировка таблицы репертуара по названию спектакля
9. Сортировка таблицы ключей (поле - название спектакля)
10. Сравнение эффективности сортировки таблицы репертуара и таблицы ключей, используя алгоритмы быстрой сортировки и сортировку пузырьком
11. Вывести список всех музыкальных спектаклей для детей указанного возраста, c продолжительностью меньше указанной
12. Справка
13. Об авторе
14. Завершить работу программы

## Аварийные ситуации

1. Некорректное содержимое файла (не соответствует заданному формату данных в файле или является пустым).
2. Несуществующий файл.
3. Некорректный пункт меню (буква или число, не соответствующее опции из меню).
4. Некорректный данные при ручном вводе (попытка записи символов в поля, представляемые числами, или превышение максимально допустимой длины строки).

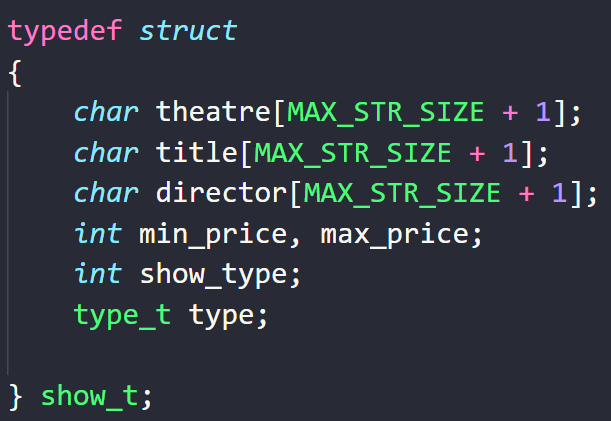
# Реализация

## Структуры данных

### Структура ***show\_t***

Для хранения строк таблицы репертуара используется агрегированный тип ***show\_t***

Константа ***MAX\_STR\_SIZE*** определяет максимально допустимое количество символов с статических строках.

****

Поле ***theatre*** представляет собой строку, размером не более **MAX\_STR\_SIZE**, ислужит для хранения названия театра.

Поле ***title*** представляет собой строку, размером не более **MAX\_STR\_SIZE**, ислужит для хранения названия спектакля.

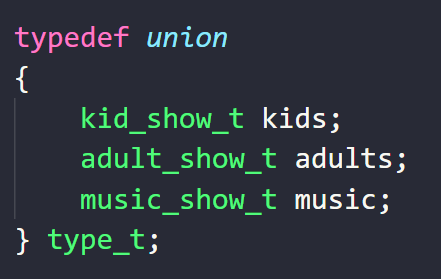
Поле ***director*** представляет собой строку, размером не более **MAX\_STR\_SIZE**, ислужит для хранения имени режиссера спектакля.

Поля ***min\_price*** и ***max\_price*** представляют собой целое число и необходимы для хранения минимальной/максимальной цены за билет на спектакль соответственно.

Поле ***show\_type*** представляет собой целое число и необходимо для хранения флага о том, какая информация хранится в вариативном поле ***type***.

Поле ***type*** является вариативным и представляется агрегированным типом ***type\_t*** для хранения вариативной части таблицы репертуара.

### Структура ***type\_t***

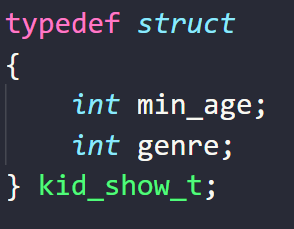


Поле ***kids*** служит для хранения информации о детских спектаклях и представляется агрегированным типом ***kid\_show\_t.***

Поле ***adults*** служит для хранения информации о спектаклях для взрослых и представляется агрегированным типом ***adult\_show\_t.***

Поле ***music*** служит для хранения информации о музыкальных спектаклях и представляется агрегированным типом ***music\_show\_t.***

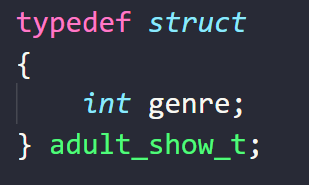
### Структура ***kid\_show\_t***



Поле ***min\_age*** служит для хранения информации о минимальном возрасте детей и представляется целым числом.

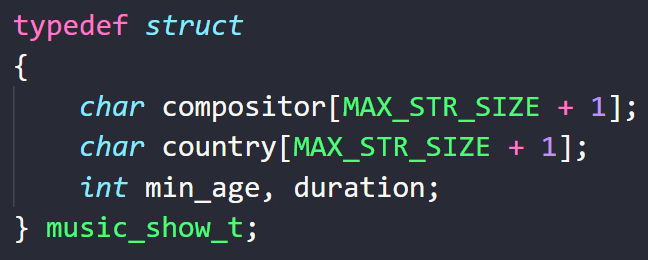
Поле ***genre*** служит для хранения информации о жанре детского спектакля и представляется целым числом.

### Структура ***adult\_show\_t***



Поле ***genre*** служит для хранения информации о жанре взрослого спектакля и представляется целым числом.

### Структура ***music\_show\_t***



Поле ***compositor*** представляет собой строку, размером не более **MAX\_STR\_SIZE**, ислужит для хранения имени композитора.

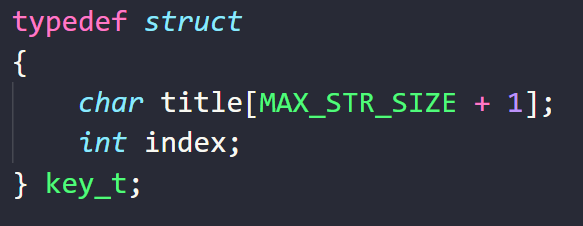
Поле ***country*** представляет собой строку, размером не более **MAX\_STR\_SIZE**, ислужит для хранения названия страны производства.

Поле ***min\_age*** служит для хранения информации о минимальном возрасте и представляется целым числом.

Поле ***duration*** служит для хранения информации о продолжительности спектакля и представляется целым числом.

### Структура ***key\_t***

Для хранения строк таблицы ключей используется агрегированный тип ***key\_t***



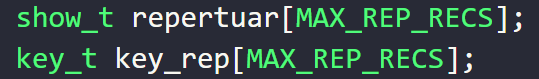
Поле ***title*** представляет собой строку, размером не более **MAX\_STR\_SIZE**, ислужит для хранения названия спектакля.

Поле ***index*** представляет собой целое число и служит для хранения информации о положении структуры ***show\_t*** в таблице репертуара.

### Таблицы репертуара и ключей



Константа MAX\_REP\_RECS определяет максимально допустимое количество записей в таблицах репертуара и ключей.



Для хранения таблиц репертуара и ключей используются статические массивы ***repertuar*** и ***key\_rep*** типов ***show\_t*** и ***key\_t,*** размерностью ***MAX\_REP\_RECS***, соответственно.

(пример таблицы репертуара находится в файле ***repertuar.txt***)

## Алгоритм

1. Пользователь вводит номер команды из меню.
2. Пока пользователь не введет 0 (выход из программы), ему будет предложено выполнять действия с таблицей.

Для реализации функции ручного добавления данных в таблицу (***repertuar\_add\_rec***) выполняется ввод данных с последующим добавлением записи в конец таблицы.

Для реализации функции ручного удаления данных из таблицы (***repertuar\_del\_rec***) выполняется выбор поля с вводом его значения для удаления. При проходе по записям ищется точное совпадение и в случае оного выполняется затирание записи сдвигом влево записей правее на 1 ячейку памяти, отводимой для хранения одной записи.

Для реализации функций сортировки данных таблицы репертуара (***sort\_rep***) и таблицы ключей(***sort\_key***) используется классический алгоритм сортировки пузырьком.

Для реализации функции сравнения эффективности сортировки таблицы репертуара и таблицы ключей используются функции сортировки пузырьком(см.выше) и библиотечная функция быстрой сортировки (***qsort***).

Для реализации функции поиска подходящих для детей музыкальных спектаклей (***shows\_for\_kids***) выполняется ввод интересующих характеристик с последующим проходом по записям. В случае соответствия полей записи требуемым производится печать записи на экран пользователя (***print\_show***).

# Тестирование

## Аварийные ситуации

|  |  |  |
| --- | --- | --- |
| № | Описание теста | Сообщение |
| 1 | Файл для чтения не существует или же у программы отсутствует доступ к нему | Некорректное название файла. |
| 2 | Файл для чтения пустой или не соответствует требуемому формату | Некорректное содержимое файла. |
| 3 | Добавление/удаление/поиск записи:  Превышение максимальной длины строки | Превышена максимальная длина строки. |
| 4 | Добавление/удаление/поиск записи:  Некорректный ввод числового поля | Некорректный ввод. |
| 5 | Некорректный пункт меню | Некорректный ввод/  Неизвестный пункт меню |

## Штатное поведение

|  |  |  |  |
| --- | --- | --- | --- |
| № | Описание теста | Действие | Сообщение |
| 1 | Чтение из файла | Ввод имени файла | Данные успешно прочитаны |
| 2 | Очистка таблицы репертуара | - | Данные успешно очищены |
| 3 | Добавление записи | Ввод полей записи | Запись успешно добавлена |
| 4 | Добавление записи  Таблица уже полна | - | Таблица репертуара полностью заполнена |
| 5 | Удаление записи | Ввод ключевого поля и его значения | Очистка по заданному полю завершена |
| 6 | Удаление записи  Таблица пуста | - | Таблица репертуара пустая |
| 7 | Печать таблицы репертуара/ключей | - | Вывод записей на экран |
| 8 | Печать таблицы репертуара/ключей  Таблица пустая | - | Таблица репертуара пустая |
| 9 | Сортировка таблицы репертуара/ключей | - | Сортировка успешно выполнена |
| 10 | Сортировка таблицы репертуара/ключей  Таблица пустая | - | Таблица репертуара пустая |
| 11 | Сравнение эффективности сортировок | - | Печать времени сортировок |
| 12 | Сравнение эффективности сортировок  Таблица пустая | - | Таблица репертуара пустая |
| 13 | Поиск музыкальных спектаклей для детей | Ввод продолжительности и возраста | Вывод подходящих записей |
| 14 | Поиск музыкальных спектаклей для детей  Таблица пустая | - | Таблица репертуара пустая |
| 15 | Поиск музыкальных спектаклей для детей  Результаты не найдены | Ввод продолжительности и возраста | Ничего не найдено |
| 16 | Справка | - | Печать справочной информации |
| 17 | Об авторе | - | Печать информации об авторе |
| 18 | Выход | - | Завершение программы |

# Оценка эффективности сортировок

Для усреднения оценки времени сортировки и снижения погрешности, связанной с исходным состоянием таблицы, замеры были проведены многократно. Таким образом, приведенное ниже время является суммарным.

Таблица 1. Сравнение времени сортировок (время в мкс)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Количество записей | Сортировка пузырьком | | Быстрая сортировка | |
| Исходная таблица | Таблица ключей | Исходная таблица | Таблица ключей |
| 40 | 69812 | 49387 | 14957 | 13963 |
| 60 | 177476 | 145610 | 89364 | 48871 |
| 80 | 303103 | 252283 | 138836 | 54836 |
| 100 | 503980 | 418919 | 156632 | 62836 |

Таблица 2. Сравнение объема занимаемой памяти (в байтах)

|  |  |  |
| --- | --- | --- |
| Количество записей | Исходная таблица | Таблица ключей |
| 40 | 13920 | 2880 |
| 60 | 20880 | 4320 |
| 80 | 27840 | 5760 |
| 100 | 34800 | 7200 |

Таким образом, итоги также удобно представить в виде таблицы.

В качестве 100% при сравнении времени использования принимает время сортировки полной таблицы.

|  |  |  |  |
| --- | --- | --- | --- |
| Количество записей | Количество дополнительной памяти необходимое для таблицы ключей | Преимущество по времени использования таблицы ключей (сортировка пузырьком) | Преимущество по времени использования таблицы ключей (быстрая сортировка) |
| 40 | ~21% | +41% | +8% |
| 60 | +21% | +83% |
| 80 | +20% | +153% |
| 100 | +20% | +149% |

# Контрольные вопросы

## 1. Как выделяется память под вариантную часть записи?

Эта память является общей для всех полей вариантной части записи. Размер памяти, выделяемый под вариантную часть, равен максимальному объему поля вариантной части.

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

Тип данных в вариантной части при компиляции не проверяется. Из-за того, что невозможно корректно прочитать данные, поведение будет неопределенным.

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

Ответственность за правильность выполнения операций с вариантной частью записи возлагается на программиста

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

Дополнительный массив (структура), содержащий индекс элемента в исходной таблице и выбранный ключ и нужна для оптимизации сортировки.

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

В случае, если записи таблицы имеют большой объем, эффективнее использовать таблицу ключей, чтобы не тратить время на копирование из одной области памяти в другую больших данных.

В случае, если записи таблицы имеют сравнительно не большой объем, то эффективнее использовать саму таблицу непосредственно, так как выигрыш по времени будет сравнительно не большой, но в тоже время под таблицу ключей будет необходимо выделить объем памяти, сопоставимый с размером основной таблицы, что в худшем случае даст проигрыш по памяти в 2 раза и 0 выигрыш по времени.

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

Если обработка данных производится в таблице, то необходимо использовать алгоритмы сортировки, требующие наименьшее количество операций перестановки. Если сортировка производится по таблице ключей, эффективнее использовать сортировки с наименьшей сложностью работы.

# Вывод о проделанной работе

В ходе лабораторной работы мы научились работать с таблицами данных, обрабатывать записи с вариативными полями, сортировать таблицы различными методами с использованием разных методов представления их.

Выделили преимущества и недостатки метода представления таблиц с помощью таблицы ключей (в данном случае потребовалось порядка 21% дополнительной памяти от размера таблицы репертуара для хранения таблицы ключей), а также оценили скорость работы различных методов сортировки для разных структур и разных объемов данных (при увеличении объема данных выигрыш использования сортировки таблицы ключей пузырьком падает с +41% до +20%, а выигрыш использования быстрой сортировки таблицы репертуара возрастает с +8% до +149%).

Таким образом, использование таблицы ключей дает существенный выигрыш при сортировке.