



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)

ФАКУЛЬТЕТ

Информатика и системы управления

КАФЕДРА

Программное обеспечение ЭВМ и информационные технологии

**Отчет по лабораторной работе №2**  
**«ЗАПИСИ С ВАРИАНТАМИ. ОБРАБОТКА ТАБЛИЦ»**

Студент

Козырнов Александр Дмитриевич

Группа

ИУ7 – 32Б

Преподаватель

Барышникова Марина Юрьевна

Вариант

6

2023 год.

## Оглавление

<a href="#">Описание условия задачи.....</a>	<a href="#">3</a>
<a href="#">Описание технического задания.....</a>	<a href="#">3</a>
<a href="#">Набор тестов.....</a>	<a href="#">5</a>
<a href="#">Описание структуры данных.....</a>	<a href="#">6</a>
<a href="#">Временные замеры (мс).....</a>	<a href="#">8</a>
<a href="#">Объём занимаемой памяти (байт).....</a>	<a href="#">9</a>
<a href="#">Оценка эффективности.....</a>	<a href="#">9</a>
<a href="#">Описание алгоритма.....</a>	<a href="#">10</a>
<a href="#">Ответы на контрольные вопросы.....</a>	<a href="#">10</a>
<a href="#">Вывод.....</a>	<a href="#">11</a>

## Описание условия задачи

Создать таблицу, содержащую не менее 40 записей с вариантной частью. Произвести поиск информации по вариантному полю. Упорядочить таблицу, по возрастанию ключей (где ключ – любое невариантное поле по выбору программиста), используя: а) исходную таблицу; б) массив ключей, используя 2 разных алгоритма сортировки (простой, ускоренный). Оценить эффективность этих алгоритмов (по времени и по используемому объему памяти) при различной реализации программы, то есть, в случаях а) и б). Обосновать выбор алгоритмов сортировки. Оценка эффективности должна быть относительной (в %).

Ввести список литературы, содержащий фамилию автора, название книги, издательство, количество страниц, вид литературы (1: техническая – отрасль, отечественная, переводная, год издания; 2: художественная – роман, пьеса, поэзия; 3: детская – минимальный возраст, сказки, стихи). Вывести список всех романов указанного автора.

## Описание технического задания

### Входные данные:

**Файл с данными:** Все строки файла через пробельный символ содержат поля структуры.

**Целое число:** задаёт пункт меню.

**Строковые и числовые данные:** в зависимости от пункта меню делают либо удаление/вставку записи, либо поиск.

### Выходные данные:

Текущая таблица, исходная таблица, отсортированная исходная таблица, отсортированная таблица ключей, сравнение эффективности сортировок, результат поиска по таблице.

### Функции меню программы:

1. Выгрузка данных из файла.
2. Просмотр текущей таблицы.

3. Добавление информации о новой книге в конец таблицы.
4. Удаление информации о книге с заданным названием.
5. Просмотр отсортированной таблицы ключей по имени автора, написавшего книгу.
6. Просмотр упорядоченной по имени автора книги таблицы.
7. Просмотр исходной таблицы в упорядоченном по упорядоченной таблице ключей.
8. Вывод списка книг с указанным автором, книги которого являются романом.

## **Обращение к программе:**

Запускается через терминал: ./app.exe

## **Аварийные ситуации:**

1. Указано неверное имя файла.  
“Файла не существует или не может быть открыт”
2. Если в файле не содержатся структуры или он заполнен некорректно  
“Файл некорректно заполнен или пуст”
3. Ввод структуры с руки оказался некорректным.  
“Вы ввели структуру неверно”
4. Удаление не существующей записи  
“Ничего не удалено по вашему запросу”
5. Ввод не существующего пункта меню  
“Выберите пункты меню!”
6. Ввод в численные поля структуры иных типов данных.

Ошибка будет локализовано и уточнено, что следует ввести.

### Набор тестов

№	Название теста	Пользовательский ввод	Вывод
1	Некорректный ввод пункта меню	abc	Выберите пункты меню!
2	Некорректный ввод пункта меню	112	Выберите пункты меню!
3	Просмотр текущей таблицы	2	Текущая таблица
4	Корректный файл	1	Данные загружены в программу
5	Корректное добавление книги	Author Kniga Izdatel 1000 1 Drevesnaya 1 2014	Запись добавлена в конец таблицы
6	Некорректное добавление книги	A 2 3 45 4	Неверный ввод типа книги! Вводить можно только число 1 или 2 или 3
7	Запрос на удаление несуществующей книги.	4 NetTakoy	Ничего не удалено по вашему запросу

8	Отсутствие книг, удовлетворяющим условиям поиск	8 NetTakoy	Ничего не найдено!
---	---	---------------	--------------------

## Описание структуры данных

### Перечисление перевода для технических книг

```
typedef enum translated
{
    native,
    translated
} translation_t;
```

**Native** — отечественная техническая книга

**Translated** — переводная техническая книга

### Структура технической книги

```
typedef struct tech
{
    char branch[BRANCH_LEN + 1];
    translation_t translation;
    int year;
} tech_t;
```

**Branch** – это отрасль, про которую написана книга.

**BRANCH\_LEN** – 25 символов

**Translation** – это тип перевода этой книги

**Year** – год издания этой книги

### Перечисление типа художества

```
typedef enum art
{
```

```
roman,  
play,  
poetry  
} art_t;
```

**Roman** — книги-романы

**Play** — пьесы

**Poetry** - поэзия

### Структура художественной книги

```
typedef struct artistic_t  
{  
    art_t type ;  
} artistic_t ;
```

**Type** – тип книги (роман, пьеса, поэзия)

### Перечисление типа детской литературы

```
typedef enum baby  
{  
    tale,  
    poem  
} baby_t;
```

**Tale** — сказка

**Poem** — стихотворение

### Структура книги для детей

```
typedef struct child  
{  
    int min_year;  
    baby_t type;  
} child_t;
```

**Min\_year** – минимальный возраст для прочтения (от 0 до 18)

**Type** – тип литературы (стихи, сказки)

## Перечисление типов книг

```
typedef enum form_t
{
    tech;
    artistic;
    childs;
} form_t;
```

**Tech** — Структура технической книги

**Artistic** — Структура художественной книги

**Childs** — Структура книги для детей/детская литература

## Структура всех книг

```
typedef struct book_t
{
    union type_t
    {
        tech_t tech;
        artistic_t artistic;
        child_t childs;
    } type;
    char author[AUTHOR_LEN + 1];
    char title[TITLE_LEN + 1];
    char producer[PROD_LEN + 1];
    form_t form;
    int number_pages;
} book_t;
```

**Type** – это смесь (объединение) всех ранее объявленных структур литературы.

**Author** – имя автора. **AUTHOR\_LEN** = 25

**Title** – название книги. **TITLE\_LEN** = 25

**Producer** – издатель. **PROD\_LEN** = 25

**Form** – тип литературы (техническая, художественная, детская)

**Number\_pages** – количество страниц



### Временные замеры (мс)

Замеры проводились с использованием ключа «Номер страницы». Замеры проводились 1000 раз на случайных ключах и усреднялись.

Кол-во записей	Быстрая сортировка		Сортировка выбором	
	Основная таблица	Таблица ключей	Основная таблица	Таблица ключей
40	0.97	0.79	2.35	1.89
500	14.44	13.11	317.67	291.97
1000	35.89	42.50	1538.09	1401.60
2500	100.22	122.94	8935.62	8159.43

### Объём занимаемой памяти (байт)

Кол-во записей	Основная таблица	С таблицей ключей
40	4960	5280
500	62000	66000
1000	124000	132000
2500	310000	330000

### Оценка эффективности

Кол-во записей	Отношение времени <b>быстрой</b> сортировки основной таблицы к таблице ключей	Отношение времени сортировки <b>выбором</b> основной таблицы к таблице ключей
40	1.23	1.24

500	1.1	1.08
1000	0.84	1.09
2500	0.82	1.1

Довольно странное поведение наблюдается у быстрой сортировки. Возможно, что я слишком сильно уменьшил размер структуры книги с помощью перечислений, из-за чего размер ключа в принципе сопоставим с быстрой сортировкой. При менее эффективных методах сортировки наблюдаем, что ключи сортируются быстрее. В среднем моя сортировка и быстрая сортировка работают одинаково и на таблице, и на ключах. Размер таблицы с ключами примерно на 6 % больше самой таблицы.

## **Описание алгоритма**

1. Программа запускается и выводит меню
2. Пользователь вводит пункт меню.
3. Если пункт меню связан с записью/вводом данных – проводится проверка данных и выполняется запрос.
4. Если пункт меню связан с выводом отсортированной таблицы – список книг/ключей сортируется методом быстрой сортировки и выводится на экран.
5. В случае возникновения аварийной ситуации программа оповещает пользователя о возникшей ошибке и продолжает работу.

## **Ответы на контрольные вопросы**

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

Выделяется на самое большое поле. Для моего случая размер смеси (объединения) будет равно размеру технической книги.

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

Ничего не произойдет, так как программа не проверяет корректность ввода данных в этой области.

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

Программист. Компилятор и программа не могут засечь ошибки работы с вариантной частью записи.

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

Таблица ключей — это только ключ и индекс исходной записи в исходной таблице. В моем случае ключом является имя автора.

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

Использование ключей эффективней во всех случаях, когда ключ достаточно мал по сравнению с самой структурой. Если структура достаточно маленькая, то использование ключей бесполезно.

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

Те сортировки, чья сложность выполнения измеряется логарифмически или линейно  $O(n \cdot \log(n))$  или  $O(\log(n))$  или  $O(n)$ .

### **Вывод**

Объединение в языке СИ помогает уменьшать размеры типов и структур данных, отчего я считаю их крайне полезными. Однако еще можно сделать вывод, что их использование требует большего понимания и внимания для работы с ними. Также было усвоено, что сортировка ключей эффективнее сортировки самой таблицы, если ключ достаточно мал и быстро сравним. То есть выделяя дополнительную память можно достичь серьезных прибавок к производительности. Также выбор сортировок может серьезно повлиять на скорость работы программы.