



Министерство образования и науки 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**

**По дисциплине «Типы и структуре данных»**

Название Записи с вариантами, обработка таблиц

Студент Мамврийский Иван Сергеевич  
*фамилия, имя, отчество*

Группа ИУ7-36Б

Тип лабораторной работы Учебная

Название  
предприятия \_\_\_\_\_

Студент	_____	<u>Мамврийский И. С</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

Преподаватель	_____	<u>Никульшина Т.А.</u>
	<i>подпись, дата</i>	<i>фамилия, и.о.</i>

**Цель работы** – приобрести навыки работы с типом данных «запись» (структура), содержащим вариантную часть (объединение, смесь), и с данными, хранящимися в таблицах, произвести сравнительный анализ реализации алгоритмов сортировки и поиска информации в таблицах, при использовании записей с большим числом полей, и тех же алгоритмов, при использовании таблицы ключей; оценить эффективность программы по времени и по используемому объему памяти при использовании различных структур и эффективность использования различных алгоритмов сортировок.

### Описание условия задачи

Ввести список машин, имеющихся в автомагазине, содержащий: марку автомобиля, страну-производитель, цену, цвет и состояние: новый – гарантия (в годах); нет - год выпуска, пробег, количество ремонтов, количество собственников. Вывести цены не новых машин указанной марки с одним предыдущим собственником, отсутствием ремонта в указанном диапазоне цен.

### Описание СД

*Для хранения данных о машине была создана запись cars\_t.*

```
typedef struct
{
    char brand[MAX_STR_LEN_BRAND];
    char country[MAX_STR_LEN_COUNTRY];
    int price;
    char color[MAX_STR_LEN_COLOR];
    car_type_t type;
    union
    {
        new_car_t new_car;
        old_car_t old_car;
    } car_type;
} cars_t;
```

**brand** - название марки машины(максимальная длина 25 символов).

**country** - название страны производителя машины(максимальная длина 25 символов).

**price** - цена машины.

**color** - цвет машины(максимальная длина 25 символов).

```
typedef enum
{
    NEW,
    OLD
} car_type_t;
```

**type** - перечисляемый тип (enum), задающий набор значений. NEW - новая машина. OLD - старая машина.

**Вариативное поле cars\_type, состоящее из двух структур.**

Первая - информация о новой машине.

```
typedef struct
{
    int garanty;
} new_car_t;
```

**garanty** - гарантия на новую машину.

**Вторая - информация о старой машины.**

```
typedef struct
{
    int year;
    int mileage;
    int count_repair;
    int count_owner;
} old_car_t;
```

**year** - год выпуска машины.

**mileage** - пробег машины.

**count\_repair** - количество ремонтов.

**count\_owner** - количество владельцев автомобиля.

*Для хранения ключей была создана структура cars\_key\_t.*

```
typedef struct
{
    size_t index;
    int price;
} cars_key_t;
```

**index** - номер машины.

**price** - цена машины.

### Описание алгоритма.

Для сортировки были выбраны алгоритмы bubble sort и quick sort. Для сортировки с использованием ключей производится создание таблицы ключей по исходной таблице, и сортировка ключей. После этого возможен вывод неупорядоченной исходной таблицы по упорядоченным ключам.

Оценка времени работы сортировок производится путем замера времени сортировок исходной таблицы и таблицы ключей с помощью библиотеки time.

## Набор тестов

### *Негативные тесты*

Ввод	Вывод	Проверяемый случай
no_file.txt	Файл не найден.	Проверка на несущ файл
51 Honda Ger ...	Неверное количество машин	Количество структура в файле больше 50
bmw Ger 1000 Red 0	Ошибка в чтении файла	Неверная запись структуры в файле
1 строка файла: abcd	Ошибка чтения количества машин	Неверно задано количество структур
1 строка файла: -1	Ошибка чтения количества машин	Неверно задано количество структур
Команда 10: 2000 1000	Введен неверный диапазон	Неверно задан диапазон цены при фильтрации
Команда 10: -1 10	Введен неверный диапазон	Неверно задан диапазон цены при фильтрации

Ввод	Вывод	Проверяемый случай
Команда: 5 Abcd	Цена введена неверно	Проверка на правильность ввода цена автомобиля
-1	Цена введена неверно	Проверка на правильность ввода цена автомобиля
Команда: 5 -1000	Пробег введен неверно	Проверка на правильность ввода пробега автомобиля
Команда: 5 -1	Количество владельцев введено неверно	Проверка на правильность ввода количества владельцев автомобиля
Команда: 5 -1	Количество ремонтов введено неверно	Проверка на правильность ввода количества ремонтов автомобиля
Команда: 6 41	Указана позиция большая, чем количество строк таблицы	Проверка на правильность ввода позиции машины для удаления
Команда 6: Ask	Данные введены неверно	Проверка на правильность ввода позиции машины для удаления

### *Позитивные тесты*

Ввод	Вывод	Проверяемый случай
Команда: 1	Время сортировки:	Успешность выполнения сортировки bubble_sort
Команда: 3	Время сортировки:	Успешность выполнения сортировки inserts_sort
Команда: 5 Bmw Ger 1000 Red 0 10	Новая машина успешно добавлена	Успешность добавления новой машины
Команда: 6	Машина успешно удалена	Проверка на успешность удаления машины
Команда: 7	«Вывод таблицы»	Проверка на правильность выполнения команды 7
Команда: 8	«Вывод таблицы ключей»	Проверка на правильность выполнения команды 8
Команда: 9	«Вывод таблицы по ключам»	Проверка на правильность выполнения команды 9

## **Сравнение эффективности (сортировка производилась по полю цена).**

### ***Таблица из 10 записей:***

**Время bubble сортировки таблицы:** 0.000010с

**Время быстрой сортировки таблицы:** 0.000004с

**Время bubble сортировки ключей:** 0.000004с

**Время быстрой сортировки ключей:** 0.000003с

Сортировка quicksort с ключами и без:

**Время сортировки таблицы:** больше на ~33%

Сортировка bubble sort с ключами и без:

**Время сортировки таблицы:** больше на 150%

**Объем исходной таблицы:** 1040

**Объем затрачиваемый для работы с ключами:** 1200

### ***Таблица из 50 записей:***

**Время bubble сортировки таблицы:** 0.000048с

**Время быстрой сортировки таблицы:** 0.000014с

**Время bubble сортировки ключей:** 0.000028с

**Время быстрой сортировки ключей:** 0.000010с

Сортировка quicksort с ключами и без:

**Время сортировки таблицы:** больше на ~40%

Сортировка bubble sort с ключами и без:

**Время сортировки таблицы:** больше на ~71%

**Объем исходной таблицы:** 5200

**Объем затрачиваемый для работы с ключами:** 6000

### ***Таблица из 90 записей:***

**Время bubble сортировки таблицы:** 0.000124с

**Время быстрой сортировки таблицы:** 0.000024с

**Время bubble сортировки ключей:** 0.000075с

**Время быстрой сортировки ключей:** 0.000017с

Сортировка quicksort с ключами и без:

**Время сортировки таблицы:** больше на ~41%

Сортировка bubble sort с ключами и без:

**Время сортировки таблицы:** больше на 65%

**Объем исходной таблицы:** 9360

**Объем затрачиваемый для работы с ключами:** 9800

### **Итог сравнительного анализа:**

Благодаря таблице ключей сортировка таблицы происходит существенно быстрее, при этом затрачиваемая дополнительная память незначительна. Из данного анализа следуют, что для сортировки таблицы, имеющей незначительные размеры, можно использовать исходную таблицу. При исходной таблице, имеющей большие размеры лучше использовать таблицу ключей для лучшей эффективности.

## **Ответы на контрольные вопросы:**

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

Объем памяти, необходимый для записи с вариантами складывается из длин полей фиксированной части и максимального по длине поля вариантной части.

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

Тип данных в вариативной части не проверяется при компиляции, поэтому могут быть успешно записаны в нее, что приведет к дальнейшей ошибки в программе.

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

Контроль за правильностью выполнения операций с вариативной частью должен следить программист.

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

Таблица ключей - это дополнительная таблица, содержащая индекс элемента в исходной таблице и значение выбранного ключевого поля (по которому, например, будет производиться сортировка всей таблицы). Таблица ключей помогает экономить время обработки основной таблицы.

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

Обрабатывать данные в самой таблице эффективнее, если таблица имеет небольшой размер. В противном случае, эффективнее использовать таблицу ключей.

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

Предпочтительнее те способы сортировки, которые не переставляют уже отсортированные элементы, так как перестановка элементов таблицы занимает много времени.