Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

—

Институт прикладной математики и механики

**Кафедра «Информационная безопасность компьютерных систем»**

**ЛАБОРАТОРНАЯ РАБОТА №2**

**База данных**

По дисциплине «Структуры данных»

Выполнил

студент гр. 13656/2 Д.В. Михайлов

Руководитель,

асс. преподавателя А.В. Мясников

Санкт-Петербург

2018

# цели работы

Написать программу **lab\_db.c**, позволяющую выполнять с БД операции вставки, удаления, обновления и поиска. Команды записаны в файле input.txt, по одной строке на команду. Результаты должны выводиться в output.txt. Программа не должна использовать никаких других "временных" файлов. Описание команд и формата их задания приведено далее. Схема базы данных указана в вашем варианте задания.

# 2 Описание задачи

Сущность описывается набором *полей*. Поле имеет *имя и тип данных*. Экземпляр со значениями полей, описывающих конкретную сущностью, называется *записью базы данных*. Диапазон возможных значений, которые может принимать поле, зависят от типа данных поля. Тип поля также называют *доменом*. Например, поле типа string может содержать строки любой длины (в том числе - пустую), поле типа date - корректную дату. От типа данных поля зависит также и набор возможных операций, участвующих в логических выражениях сравнения. Например, числа и даты можно сравнивать на больше/меньше/равно/не равно. Для строк существует такой же набор операций, но, например, сравнение на больше или меньше сравнивает порядок строк в алфавитном порядке.

В зависимости от типа данных определяется способ записи значений поля. Например, строки всегда записываются в двойных кавычках, даты - в одинарных, а числа - без кавычек.

# 3 Алгоритмы и методы

В качестве метода решения данной задачи я решил использовать метод, используемый в первой лабораторной работе, а именно: парсинг строки, определение ключевого слова и выполнений соответствующей для него функции.

# 4 Структуры данных и алгоритмы

Используемые структуры:

typedef enum { Select, Del, Update, Uniq, Sort } **OperationType**;

typedef struct

{

char\* str;

} **String**;

typedef struct

{

char\* phoneNumber;

} **Phone**;

typedef struct

{

char Number[9];

} **Carnum**;

typedef struct

{

char\*\* items;

int value;

} **Enum**;

typedef struct

{

String\* firstName;

String\* lastName;

Phone\* call;

Carnum\* car;

Enum\* status;

int id;

int used;

} **PassCard**;

typedef struct

{

PassCard\* card;

struct Element\* next;

struct Element\* prev;

} **Element**;

typedef struct

{

Element\* data;

Element\* tail;

int size;

} **List**;

typedef struct

{

char\* fields[7];

PassCard\* values;

PassCard\* updateValues;

char\* cond[7];

int condCount;

} **Conditions**;

Кроме того, в программе используется большое количество алгоритмов и функций, которые упрощают работу программы, например проверка корректности введенной строки, компаратор значений, отдельный алгоритм перехода к другой записи в базе данных, алгоритм сохранения и загрузки базы данных, алгоритм сортировки записей по условиям и т.д.

# 5 Тестирование

* Создаются входной файл: input.txt
* Запускается программа lab\_db.exe с помощью системного вызова system(), ожидается ее завершение.
* Считываются созданные программой lab\_db выходные файлы.
* Считанные данные анализируются: проверяются на соответствие ожидаемым.
* В случае соответствия в стандартный поток выводится 'test ZZ:ok', иначе 'test ZZ:fail', где ZZ - номер теста. По одной строке на каждый тест.

# 6 выводы

В ходе выполнения работы я научился исследовать строку, используя указатели на пробелы и знаки «=», преобразовывать строку «string» в целое значение типа «int», используя функции «atoi» и «atol». Во время выполнения работы я столкнулся с трудностью работы со структурами и списками.

Помимо этого, проблемой стала реализация алгоритма сортировки для элементов структуры и переход с новой записи. Я так же научился использовать структуры, переходить по указателю к новой записи, определять имя и тип поля исходя из множества, работать с регулярными выражениями и писать шаблоны для них.